



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE POSGRADO EN ECONOMÍA

INSTITUTO DE INVESTIGACIONES ECONÓMICAS
CAMPO DEL CONOCIMIENTO: ECONOMÍA POLÍTICA

LA ORGANIZACIÓN DEL PROCESO DE TRABAJO COMO MECANISMO DE
DEBILITAMIENTO DEL PODER DE NEGOCIACIÓN DE LOS TRABAJADORES:
LA PROGRAMACIÓN INFORMÁTICA COMO CASO PARADIGMÁTICO DEL
CAPITALISMO COGNITIVO

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN ECONOMÍA

PRESENTA:
DIEGO JOSÉ ZÁRATE MONTERO

TUTOR
MTRO. JOSÉ GUADALUPE SANDOVAL MANZANO
FACULTAD DE ECONOMÍA

CIUDAD UNIVERSITARIA, CD DE MÉXICO, DICIEMBRE, 2021



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Dedicatoria

Al pueblo de México, que derramó su sangre por su Independencia

y la de Centroamérica.

Agradecimientos

En primer lugar agradezco al pueblo de México por su hospitalidad, calidez y solidaridad, agenciadas a través de tantos mexicanos y mexicanas que conocí y compartí durante mis casi tres años de vida en ciudad de México, la mayoría desde el anonimato histórico que caracteriza al proletariado.

En segundo lugar agradezco a mis compañeros y compañeras durante los cursos de la maestría, porque fueron un apoyo fundamental tanto en asuntos de la inmediatez cotidiana como en temas de la vida académica y estudiantil en la UNAM.

En tercer lugar, agradezco a mis maestros y maestras a lo largo de los cursos del plan de estudios, así como a los sinodales que me han acompañado en esta etapa final. Su comprensión y paciencia, así como su rigurosidad intelectual y académica, han sido fuente de inspiración para no darme por vencido ante las adversidades que acompañan la maravillosa experiencia de ser estudiante internacional. Y en particular, a la Mtra. Georgina Naufal, tanto por sus clases y momentos dedicados a platicar conmigo, como por el meticuloso escrutinio al que sometió al borrador que le envié de este documento.

En cuarto lugar, le agradezco al Mtro. Sandoval por su apoyo desde el primer instante en que le consulté si querría ser mi tutor de tesis. Sus juicios, valoraciones, comentarios y recomendaciones siempre han sido incisivos, informados y pertinentes. En sus palabras encuentro una fuente de saberes en economía política que me motivan a leer, investigar y aprender más, mucho más.

Finalmente, agradezco a las y los trabajadores de la UNAM que desde sus oficinas han posibilitado mi participación en este programa de maestría.

Presentación

La tesis que se presenta a continuación busca disputar a los teóricos del capitalismo cognitivo los principales conceptos en los que está planteado el debate contemporáneo respecto del impacto de las tecnologías de la información en la sociedad.

La discusión está planteada en términos de los puntos de encuentro de los teóricos del capitalismo cognitivo, sucesores de las concepciones del post-fordismo bajo la influencia del operaísmo italiano y la escuela francesa de la regulación, según Thompson y Smith (2017: 7), con el pensamiento marxista en sus debates contemporáneos sobre el análisis de clase .

El capitalismo cognitivo es un planteamiento teórico que ha surgido como una interpretación convincente sobre tres cambios que se han venido acentuando desde el último cuarto del siglo XX: i) cambios en la estructura del empleo en términos sectoriales, ii) cambio en las relaciones industriales y el poder de negociación de las partes, iii) aumento de la desigualdad en la distribución del ingreso.

En sentido opuesto, la vigencia del pensamiento marxista en general ha sido ampliamente cuestionada en casi todos los flancos de la lucha de clases, y más aún desde los salones de la enseñanza en economía, devenidos en púlpitos laicos de la axiomática neoclásica y sus variaciones, donde la economía política es constantemente situada en el anaquel del pensamiento económico cuál saber obsoleto y de poca utilidad práctica para su empleabilidad por los estudiantes, como si no fuera igualmente válido predicar esos mismos adjetivos de la teoría neoclásica (Krueger, 1999: 156).

Por ello se considera relevante recuperar la radicalidad del marxismo ortodoxo, como análisis del modo de producción capitalista desde la perspectiva de la clase

proletaria, para abordar el novedoso sector de la informática y argumentar que las horas de trabajo tienen un interés científico para la economía en general y no solo el interés político de combatir la explotación capitalista.

Esta tesis tiene como objetivo analizar el efecto de debilitamiento del poder de negociación de los programadores informáticos causado por la división del trabajo impuesta a través de los capataces e ideólogos pagados por los capitalistas, enfatizando en la sistemática división en tareas cada vez más simples que persigue limitar todas las posibilidades de insubordinación de la clase trabajadora y asegurar con ello la existencia del plusvalor, y por tanto, de la ganancia a través de la cual se reproduce la clase capitalista como propietaria de todos los factores objetivos involucrados en dicho proceso de producción.

Para ello, en esta investigación se discuten las argumentaciones de los teóricos del capitalismo cognitivo, así como las de sus críticos, para buscar evidencia a favor y en contra de su interpretación sobre los cambios en la economía mundial, a partir del estudio de caso de los sistemas informáticos para la gestión del aprendizaje (LMS por sus siglas en inglés), dado el interés inapelable para las instituciones académicas y educativas como la UNAM.

El documento está organizado en cinco capítulos. En el primero se exponen los datos relacionados con los cambios cuantitativos y cualitativos en la economía mundial durante los últimos 50 años, enfatizando los acontecidos en el mercado de trabajo.

En el segundo capítulo se expone los lineamientos seguidos durante la investigación, detallando sus particularidades en relación con el problema que se pretende asir, las hipótesis sobre las que se busca evidencia a favor y en contra, así

como las características metodológicas asumidas para obtener resultados verosímiles, coherentes y debatibles.

En el tercer capítulo se exponen los hallazgos obtenidos al respecto de la división del trabajo en la programación informática y sus efectos observados y potenciales sobre el poder de negociación de los trabajadores de la rama, en el marco del debate de las tesis que fundamentan la interpretación de los teóricos del capitalismo cognitivo.

En este capítulo se muestran ejemplos y se detallan figuras de diversos componentes de un sistema informático con el fin didáctico de facilitar la lectura y comprensión de los conceptos propios de la informática para quienes no estén familiarizados con ellos.

En el cuarto capítulo se propone, desde el debate contemporáneo en el pensamiento marxista de la expresión monetaria del tiempo de trabajo (MELT por sus siglas en inglés), un procedimiento para aproximar el valor en horas de trabajo a partir del precio de venta de un sistema de gestión de aprendizaje (LMS) producido por trabajadores aparentemente autónomos en julio de 2019 para tres territorios articulables en el eje “Norte-Sur” del continente americano dentro la rama informática: Berkeley, California, Ciudad de México, México, y Heredia, Costa Rica.

Finalmente, en el quinto capítulo se resumen los principales hallazgos y resultados obtenidos para informar sobre el cumplimiento de los objetivos planteados en esta investigación.

Índice general

Índice de figuras	viii
Índice de tablas.....	viii
Capítulo 1. Contexto, antecedentes y estado del arte	1
1. Introducción	1
2. La organización del proceso de trabajo como determinante del poder de negociación de los trabajadores	2
3. Las tecnologías de la información en la economía mundial a finales del siglo XX.....	7
4. Las tesis del capitalismo cognitivo y sus críticas	10
5. El marxismo ortodoxo como análisis de la clase obrera en el capitalismo cognitivo	13
6. La autonomización del trabajo en relación con el poder de mando capitalista	19
Capítulo 2. Planteamiento de la investigación	22
1. Introducción	22
2. Planteamiento del problema	22
3. Pregunta de investigación.....	25
4. Objetivos	26
5. Hipótesis.....	26
6. Metodología	27
7. Variables de investigación y técnicas de recopilación y manipulación de la información y los datos.....	30
8. Limitaciones de la investigación	32
Capítulo 3. Análisis del proceso de trabajo de programación informática.....	36
1. Introducción	36
2. El valor de uso informático	37
3. Los factores objetivos en la producción de valores de uso informáticos	39
3.1 La materia prima	39
3.2 El medio de trabajo	46
3.3 Los materiales auxiliares.....	48
3.4 El edificio	49
4. El factor subjetivo del proceso de producción de valores de uso informáticos.....	50
4.1 Las tareas de concepción.....	52
4.2 Las tareas de ejecución.....	58
5. Las comunidades de programación libre y abierta como obrero colectivo ($s, -s$)	66

6.	Las metodologías para el desarrollo de los sistemas informáticos: autonomía y control....	71
6.1	La autonomía en términos de contenido, remuneraciones y horas de trabajo.....	80
7.	Fundamentos económicos de la productividad de los obreros colectivos $(s, -s)_i$	85
7.1	La modularidad	86
7.2	El reciclaje de código	88
7.3	“La ley de muchos ojos”	91
7.4	Las prácticas comunitarias	93
8.	División del trabajo y lucha de clases	98
9.	Flexibilidad modular, control numérico y analfabetismo informático.....	101
9.1	La flexibilidad modular como estrategia de debilitamiento del poder de negociación de los trabajadores informáticos	103
9.2	Tipología de usuarios de valores de uso informáticos según su nivel de alfabetización informática	107
10.	Conclusiones	111
Capítulo 4. Análisis de la formación del valor y la valorización de un sistema informático de gestión de aprendizaje mediante un ejemplo práctico.....		114
1.	Introducción	114
2.	Precio y valor en el debate contemporáneo de la expresión monetaria del tiempo de trabajo en la rama informática.....	116
3.	Precio de venta de los valores de uso informáticos.....	122
4.	El precio de venta de los sistemas de gestión del aprendizaje (LMS).....	128
5.	Cantidad de horas de trabajo en programación informática requeridos para producir un sistema de gestión de aprendizaje para 100 usuarios y cinco módulos.....	134
6.	El bloque continental americano Norte-Sur como eje territorial del análisis de los costos de producción.....	142
6.1	Condiciones generales del empleo y la ocupación en California, Ciudad de México y Costa Rica	145
6.2	La seguridad social.....	146
6.3	El valor de la fuerza de trabajo en la programación informática	149
7.	El precio de los factores objetivos para la programación informática	156
7.2	El medio de trabajo	159
7.3	Materiales auxiliares y del edificio en obreros colectivos $(s, -s)_j$	163
7.4	Precio de costo de una hora de programación informática para obreros colectivos $(s, -s)_j$	165
8.	Precio de costos para los obreros colectivos $(s, -s)_i$: industria domiciliar y trabajo autónomo.....	168

8.3 El precio de costo de una hora de programación informática en el obrero colectivo ($s, -s$) i como industria domiciliar.....	175
9.1 Los obreros colectivos ($s, -s$) i como empresa periférica en Berkeley, California.....	177
9.2 Tasa de plustrabajo en el obrero colectivo ($s, -s$) i en Ciudad de México y en Heredia, Costa Rica	179
10. Apropiación capitalista de los recursos comunitarios: subordinación del obrero colectivo ($s, -s$) i al poder de mando capitalista sobre su trabajo	185
Capítulo 5. Conclusiones.....	192
Bibliografía	201
Anexo 1: Evidencia documental de una práctica colaborativa en una comunidad de programación libre y abierta. Secuencia de tres capturas, 18 marzo -13 diciembre 2011	234
Anexo 2: Tecnología en la producción de alfileres en Francia 1760 y en Inglaterra en 1832 según Charles Babbage	235

Índice de figuras

Figura 1. Arquitectura informática de un soldado robótico dotado con inteligencia artificial... 53	
Figura 2. Diseño de una clase para un sistema multiagente de propósitos múltiples.....	55
Figura 3. Diseño de un componente electrónico multiagente para programación robótica	57
Figura 4. Algoritmo en el lenguaje interpretado Python para calcular equilibrios Nash en matrices 2x2	59
Figura 5. Salida obtenida mediante un algoritmo en Python que imita una calculadora de equilibrios Nash	60
Figura 6. Prueba del algoritmo desarrollado para calcular equilibrios Nash en matrices 2x2	61
Figura 7. Correcciones de integración de un sistema multiagente en un entorno con usuarios .	63
Figura 8. Modularidad del código de Mozilla antes y después de migrar a programación abierta	87
Figura 9. Matriz de asignación de tareas s para programas informáticos deterritorializados de gran escala.....	90
Figura 10. Consumo de Kwh por 8 horas de uso de la computadora “HP EliteBook 1050 G1 - 15.6” - Core i7 8750H - 16 GB RAM - 256 GB SSD”	169

Índice de tablas

Tabla 1. Precios según la cantidad de funciones disponibles para los sistemas de gestión de aprendizaje (LMS), al tipo de cambio peso/dólar del 15 de julio del 2019.....	130
Tabla 2. Estimación de horas requeridas para desarrollar un módulo de LMS.....	136
Tabla 3 Precios de las oficinas en Regus por jornada diaria de ocho horas por persona en pesos del julio del 2019.....	164

Tabla 4: Precio de costo de una hora de trabajo en programación informática de un obrero colectivo [(s,-s)] _j. Berkeley, California, EE.UU; Ciudad de México, Mx; Heredia, CR. En pesos mexicanos de julio, 2019.....	165
Tabla 5: Costo de internet por una velocidad de 100mbs en pesos de julio del 2019.....	174
Tabla 6. Precio de costo de una hora de trabajo en programación informática. De un obrero colectivo [(s,-s)] _i. Berkeley, California, EE.UU; Ciudad de México, Mx; Heredia, CR. En pesos mexicanos de julio, 2019.....	175
Tabla 7. Tasa de explotación en las comunidades de programación libre y abierta contratadas para desarrollar un sistema de gestión de aprendizaje (LMS) en ciudad de México y en Heredia, Costa Rica.	183
Tabla 8: División del trabajo en la industria de alfileres, Francia 1760 e Inglaterra 1830.....	235

Capítulo 1. Contexto, antecedentes y estado del arte

1. Introducción

En este primer capítulo se presenta el contexto general en el que surge la investigación, es decir, los cambios cuantitativos y cualitativos experimentados por la economía mundial a partir de la crisis en los países industrializados en la década de los 70, las reformas estructurales en América Latina en los 80 y la crisis en el pensamiento marxista a partir de los 90 en medio del auge de las tecnologías de la información y las comunicaciones en Estados Unidos y su posterior propagación al resto de las economías.

La profundización de estos cambios en la estructura del empleo a partir de la década de los 90 formó parte de los acontecimientos relacionados con la globalización, principalmente por el auge de la informática y el internet¹. Por ejemplo, para el año 2000 la ocupación en servicios (incluidos los informáticos) en Estados Unidos alcanzó el 74.5%, mientras que en América Latina rondaba el 60% del total de ocupados (Lee, McCann y Messenger, 2007: 88)

Con base en esto, los defensores del capitalismo cognitivo recuperan los conceptos de la teoría de la regulación, tanto para enfatizar los paradigmas de la

¹ La historia del internet, según Trigo Arana (2004: 23), se remonta a finales de la Segunda Guerra Mundial pero adquiere la forma de red a partir de 1969. Su desarrollo está ligado a la informática pero su difusión entre el público general se logró a través del lenguaje HTML por el físico británico Tim Berners-Lee en 1990. Durante los 90 hubo una intensa competencia entre las empresas por productos como navegadores, buscadores, correo electrónico en Estados Unidos. La “guerra de navegadores” tuvo consecuencias como la liberación del código de Netscape 1998, lo que daría lugar a un proyecto colaborativo con el navegador Mozilla Firefox, el cual fue ganando cuota de mercado paulatinamente (Tabarés Gutiérrez, 2012: 77)

organización del trabajo como la financiarización², en tanto fenómeno que se ha venido pronunciado desde la crisis de los 70, y se ha visto ampliamente expandido con el avance de las tecnologías informáticas, el internet y la globalización.

2. La organización del proceso de trabajo como determinante del poder de negociación de los trabajadores

La disminución porcentual de la industria en la ocupación total y el aumento de la participación relativa de los servicios, primero observado en países industrializados como Estados Unidos, y posteriormente en las periferias latinoamericanas y asiáticas³, ha sido destacado como uno de los cambios más significativos experimentados por la economía mundial durante la segunda mitad del siglo XX.

Más aún, el desempeño del capitalismo global entre la crisis de 1973 y la de 2008, a pesar de las ventajas que supuso la disolución del bloque soviético en los 90 y la incorporación de China a la Organización Mundial del Comercio (OMC) a inicios del

² Alicia Girón (2008: 52) ha señalado tres etapas a partir de Bretton Woods y el desempeño del FMI, tal que a partir de los 70 hay una era pos-Bretton Woods en la que dicha institución dejó de ser el centro del Sistema Financiero Internacional y regulador de los préstamos interbancarios. Su crisis de identidad dio lugar a reformas de la institución que le permitieron vincularse directamente con el Consenso de Washington en relación con el objetivo principal de la desregulación de los mercados financieros, productivos y laborales. Esto dio lugar a una reestructuración de las bases productivas del orden económico establecido desde finales de la Segunda Guerra Mundial, cuando las instituciones públicas y los sindicatos cumplieron un papel prioritario al favorecer un Estado benefactor y donde la inversión privada y pública convivían coordinadamente. Considera que los años 90 fueron la “década de la crisis financiera” a nivel global, y la primera década del siglo XXI fue de estabilidad sin crecimiento económico y una marcada profundización financiera. En este sentido, dicha profundización ha adoptado la forma de del endeudamiento externo en América Latina, cuya deuda regional representaba “casi la tercera parte del endeudamiento externo de los países en desarrollo y un poco más de la tercera parte del servicio de la deuda externa de los países en desarrollo” entre 2000 y 2008. Además, señala que el proceso de reformas a nivel global ha provocado inequidad, a pesar del crecimiento económico. Así, la heterogeneidad de las relaciones internacionales ha profundizado la dramática inequidad en la distribución del ingreso en muchos países.

³ Godbout (1993: 10) encuentra que en el periodo 1970-1990 en 10 países industrializados la ocupación en la industria y la manufactura disminuyó porcentualmente, mientras que las actividades de servicios pasaron del 40% en total en 1970 a representar el 63% de todo el empleo, con Estados Unidos, Canada, Australia y Reino Unido por encima del 70% del total en 1990. En América Latina, por ejemplo, Montusci (1999: 14) encuentra que esta dinámica de desindustrialización y tercerarización de la economía argentina inicia en 1974 con una caída del empleo que pasó de 41,3% al 20% en 1997, mientras que el sector de servicios y comercio pasó de representar el 40% del empleo al 60% en el mismo periodo. Según Mao, Xu y Zou (2018: 2) en 2014 el sector servicios ocupó el 41% del empleo en China.

2000, es constantemente comparado por diversos investigadores desde distintos enfoques con el del periodo previo, caracterizado tempranamente por Gramsci como fordismo (2000: 62), y cuyo pináculo fue el periodo comprendido entre el fin de la Segunda Guerra Mundial en 1945 y la crisis de los 70 en los países industrializados.

En este sentido, los paradigmas de la organización del trabajo han sido ampliamente tratados a partir de los aportes de la escuela francesa de la regulación, la cual se constituiría por los años de la crisis de los 70 en los países industrializados a partir de una interpretación althusseriana de *El Capital* que, según Gast (2010: 4), critica la desatención de la dimensión histórica del estructuralismo, rechaza la necesidad de la sucesión de una etapa competitiva del capitalismo a otra monopólica y no recurre a leyes generales de largo plazo⁴.

Boyer (2007: 69), exponente destacado de dicha escuela, identificó el cambio sobrevenido en los 70 como un agotamiento del régimen de "acumulación intensiva con consumo de masas", o mejor conocido como fordismo, y el advenimiento de un nuevo régimen al que denomina de "acumulación extensiva con profundización de la desigualdad", con algunos rasgos distintivos como el agotamiento de fuentes anteriores de crecimiento de la productividad (tecnológicos y sociales) y una pérdida de poder de negociación de los sindicatos expresada como una individualización de la relación salarial fordista que ha favorecido el crecimiento de las desigualdades de ingreso, incluso entre los mismos asalariados⁵.

⁴ Pereyra (1984: 14) critica la interpretación althusseriana de que la tradición marxista se ha opuesto a que sea el ser humano quien hace la historia. Para este autor es al revés, es una idea que está presente a lo largo de la obra de Marx y sus continuadores. Recupera la cita de *El 18 Brumario de Luis Bonaparte* donde Marx afirma que los seres humanos hacen su historia pero no a su libre arbitrio sino en condiciones que no eligen, para afirmar que no es trivial la comprensión de que los seres humanos se plantean en cada época los problemas que pueden resolver y las metas que pueden alcanzar.

⁵ En recientes investigaciones empíricas y teóricas se ha destacado la relación entre sindicalización y distribución del ingreso. Por ejemplo, Bacaro (2008: 28) muestra mediante un análisis econométrico

Las reformas laborales implementadas para la salida a la crisis de los 70, y que supusieron un viraje en las políticas públicas y la ideología dominante, al pasar del predominio del keynesianismo al monetarismo en la política macroeconómica y del fordismo al “posfordismo” en la organización del proceso de trabajo, estuvieron determinadas por la firme convicción de políticos, como Margareth Thatcher (1993), quienes consideraban que el alto nivel de desempleo estaba relacionado con el alcance del poder sindical.

En este sentido, en la teoría económica neoclásica la acción colectiva de los trabajadores, generalmente asociada a los sindicatos, fue destacada en el periodo “fordista-keynesiano” del capitalismo por la negociación colectiva de la relación salarial, tanto con respecto a la remuneración como a las condiciones de trabajo. Por eso en ese enfoque, la figura del sindicato o la negociación colectiva se considera una rigidez del mercado de trabajo (Mankiw, 2014: 285).

Esto determinó que en Reino Unido, entre otras medidas, la reforma declarara ilegal el requisito de pertenecer a un sindicato para obtener un contrato laboral. Los demás países industrializados siguieron lineamientos semejantes aunque no con tanta vehemencia.

comparativo entre países industrializados y en vías de desarrollo, que en los países avanzados los sindicatos siguen teniendo un papel importante en la redistribución de la riqueza. También desde la teoría de juegos se ha mantenido la idea del estudio de los sindicatos como acción colectiva relevante en la fijación salarial y la distribución del ingreso. Por ejemplo, García y Miller (2011) documentan experimentos a partir de la teoría de juegos no cooperativos en el contexto del mercado laboral empleador/empleado, en los cuales se establecen negociaciones salariales por tareas, donde encuentran que cuando "aumenta" el desempleo introduciendo más empleados que empleadores en el experimento, aumenta la tendencia a que los trabajadores acepten salarios más bajos. También encuentran que si se permite que los empleados se comuniquen entre sí y se coordinen a través de sindicatos, da lugar a mayores salarios y refuerza el poder de negociación de los trabajadores (2011: 150). Zapata Lilo (2013: 50) presenta un particular juego no cooperativo de negociación donde un sindicato se enfrenta a una empresa, pero no sabe si cuenta con el apoyo de la población general, lo cual es decisivo en caso de tener que resistir una lucha larga y difícil. Este caso no viene acompañado de un experimento como tal, sino de una identificación de escenarios y posibles soluciones en casos de decisión simultánea e información no perfecta. Bucella y Fanti (2018) publicaron recientemente un trabajo bajo este enfoque puramente teórico.

En América Latina este cambio tuvo como referencia el denominado Consenso de Washington, cuyas políticas fueron parcial o completamente aplicadas a partir de la década de los 80, y que en la década de los 90 fueron entendidas como reformas estructurales, asociadas al término de neoliberalismo y basadas en un decálogo en el que no se incluyó directamente al mercado laboral, pero que en la práctica ha sido la antesala de la flexibilización laboral en la región⁶.

Si se consideran las estadísticas de la Organización para la Cooperación y el Desarrollo Económicos, (OECD, 2018) , por sus siglas en inglés, el descenso de la afiliación sindical desde la segunda mitad de siglo XX ha sido una constante en países como Estados Unidos y Reino Unido, donde la tasa de densidad sindical pasó de un 25% y 43% respectivamente en 1975 a un 10, 26% y 23,5% en 2016.

En México la densidad de trabajadores sindicalizados pasó de 40% en 1975 a 27% en 1989. Para el 2017 solo el 12,46% de los trabajadores estaba sindicalizado (OECD, 2018). Este hallazgo parece confirmar la conclusión de De la Garza Toledo sobre la relación entre la reforma estructural y la organización sindical de los trabajadores, de que para la clase obrera mexicana, desde el punto de vista técnico, político y social, la crisis y la reconversión de los 80 significó su derrota y parcial recomposición (1993:132)

Para los 70, ya Braverman (1998:104) había diagnosticado que la aparente aclimatación de los trabajadores a los nuevos modos de producción, era debido a: 1) la destrucción de toda otra forma de vida; 2) las negociaciones salariales que permiten un

⁶ García Covarrubias (2015: 14) sostiene que la reforma laboral en Argentina inició en 1991 resultado de la implementación de cambios estructurales debidos a la crisis de la deuda, mientras que en México tuvo lugar hasta el 2012. Morandé (2016: 56) en su balance sobre el legado de las políticas del Consenso de Washington señala muchos sectores sociales que han sido excluidos de los frutos del crecimiento debido a las políticas de los 90, ya que la desregulación del mercado laboral ha permitido que los empleadores saquen ventaja de los trabajadores.

aumento de los límites de consumo de subsistencia para la clase trabajadora; 3) el tejido de la red del moderno estilo de vida capitalista que hace finalmente cualquier otro modo de vida imposible⁷.

Montusci (1998: 9) destacó tres cambios en los mercados de trabajo que fueron determinantes para la vida sindical: i) requerimiento de una mayor flexibilidad laboral; ii) la sostenida emergencia de una economía de servicios; iii) la creciente globalización de las economías. Concluye que la tendencia es que las relaciones laborales sean más individualizadas, ya que los objetivos y finalidad de la negociación colectiva están cuestionados, y la utilidad y fines del sindicalismo ya no atraen a la nueva fuerza laboral.

Pese a los acontecimientos recientes y los planteamientos teóricos al respecto, tras esta historia de los paradigmas de la organización del trabajo subyace la historicidad de la resistencia de los trabajadores a aceptar las condiciones que suponen dichos paradigmas en el ámbito concreto de la jornada laboral.

Los ludistas, por ejemplo, forman parte de esa historicidad, así como los concejistas en los talleres y fábricas y también las reiteradas huelgas en todos los países del mundo. En el debate heterodoxo, las posiciones marxistas suelen ser enfáticas en la visión de la clase trabajadora sobre esos episodios de insubordinación, a pesar que, como sostiene Zorzoli (2018: 3), no existe una teoría marxista del sindicato, incluso luego que autores como Gramsci dedicaran sesudas reflexiones a la organización obrera en las fábricas.

⁷ Asimismo, Gordon, Edwards y Reich (1986: 302) analizaron los cambios en el mercado de trabajo estadounidense, desde la perspectiva de las ondas largas, en el periodo 1820-1980, para contestar a la pregunta sobre la incapacidad de la clase obrera estadounidense para articular un programa político coherente con un discurso alternativo durante la crisis de los 70.

Esta concomitancia entre el estudio de la organización del trabajo y la historia de la acción conflictiva por parte de los trabajadores se encuentra de manera pronunciada e inequívoca en la sección IV de *El Capital*, donde Marx refiere los primeros años de vida de la organización sindical, o coaliciones obreras, durante la segunda mitad del siglo XVIII, las cuales fueron durante más de medio siglo asociaciones secretas y perseguidas por la policía. Las leyes represivas de 1799-1800 contra las coaliciones obreras fueron derogadas en 1824, sostiene Marx (1984: 552), pero de hecho no pocas de sus disposiciones siguieron en vigor hasta mucho más tarde.

Sin embargo, los teóricos del capitalismo cognitivo, nutridos de fuentes marxistas y de los planteamiento de la escuela de la regulación, insisten en que en el capitalismo contemporáneo ha habido una transformación más profunda más allá del cambio del régimen de acumulación, debido a que la segmentación del trabajo cognitivo que individualiza la relación de trabajo de acuerdo con conocimientos jerarquizados, ocasiona un predominio de la negociación individual del salario y las condiciones de trabajo, en vez de las convenciones colectivas (Míguez, 2014: 41)

3. Las tecnologías de la información en la economía mundial a finales del siglo XX

El despliegue de los servicios relacionados con la llamada revolución de las tecnologías digitales (los microprocesadores, la informática y el internet principalmente), tuvo como epicentro la economía estadounidense, y despertó un gran entusiasmo entre diversos autores para quienes una nueva economía había surgido y

propusieron conceptualizar los cambios en la economía como una etapa posterior al capitalismo industrial⁸.

Pero la euforia por el aspecto puramente tecnológico de la nueva economía duraría poco, ya que la crisis de 2001 en Estados Unidos representó un revés que dio lugar a reconsideraciones entre sus defensores, entre las que se encuentra que la nueva economía se basaba en una combinación entre activos tangibles e intangibles, como una fuerza de trabajo con habilidades amplias, aprendizaje durante toda la vida y marcada por el riesgo y la oportunidad. De hecho, para Benbya y Belbaly (2002: 7) una de las razones de la crisis de las dotcom y la quiebra de muchas empresas se debió a que en gran parte de ellas había conflictos en la administración y en el control del personal y sus responsabilidades, es decir, en la organización del proceso de trabajo.

Fue precisamente en este contexto, cuando a comienzos del año 2000 en Francia, surge la teorización sobre el capitalismo cognitivo alrededor de la revista *Multitudes* y donde volvieron a alzar la voz autores que habían participado en el debate de la década previa en torno a la revista *Futur Antérieur* y las transformaciones del mundo del trabajo en la última década del siglo XX, defendidas por Negri y Lazarato (2001: 10) a partir del ciclo de la producción “inmaterial”⁹.

⁸ Stiroh (1999: 87) identifica, dentro del marco de la nueva economía, tres amplios enfoques entre sus defensores durante los 90: el enfoque sobre el crecimiento de largo plazo (el impacto positivo de la globalización y las tecnologías de la información no puede ser medido simplemente...), el enfoque sobre el ciclo de los negocios (disminución simultánea del desempleo y la inflación), y el enfoque sobre las fuentes del crecimiento (rendimientos crecientes por efectos de la red). Todas ellas comparten el énfasis en la globalización creciente y la expansión de las tecnologías de la información como causas del sólido desempeño económico de Estados Unidos y la transformación de su economía en relación con el capitalismo industrial. En 1998, el entonces presidente de la Reserva Federal, Alan Greenspan (1998), también se refirió a los planteamientos sobre la nueva economía para rechazar que la economía hubiera dejado de regirse por la psicología humana que dirige las economías competitivas de mercado, pero reconoció que los cambios tecnológicos de los 90 alteraron la manera en la que se organizaba la producción y el comercio mundial, así como un aumento en el nivel de habilidades de la fuerza de trabajo era debido a la presión de la gerencia de negocios para aumentar la eficiencia y la innovación.

⁹ Dentro del pensamiento neoclásico se ha propuesto el término capital humano y crecimiento endógeno para referir las características de las economías basadas en conocimiento, es decir, las que conciben el conocimiento como un factor clave del crecimiento de la productividad y el desarrollo económico. En

Así, los cambios en las economías capitalistas avanzadas de occidente al final del siglo XX, son reinterpretados como un resultado novedoso de la subsunción de la ciencia por el capital, a pesar que dicha idea se encuentra arraigada en la sección IV de *El Capital* (1984: 470), y es retomada por Harry Braverman (1998:107) ante la evidencia de que el aporte de horas inmediatas de producción en cada jornada laboral se reduce constantemente, y que por ello cada trabajador se ubica en un segmento cada vez más pequeño y simple del complejo del proceso de trabajo.

Sin embargo, la reacción de los operaístas italianos que dan sustento a las tesis del capitalismo cognitivo, ya buscaba romper con la interpretación marxista desde los 70 en el sentido de que para ellos estos cambios debidos a la aplicación sistemática de la ciencia en los procesos de trabajo se explican porque la ley de valor ha perdido validez histórica, en el sentido de que el tiempo de trabajo como elemento cuantitativa y cualitativamente determinado y diferente de cada actividad laboral deja de ser la base de la producción (Negri, 2004: 24).

Para la década de los 90 Negri profundiza esta interpretación a partir del emergente sector informático y propone que la teoría del valor que debe usarse para el estudio de la sociedad debe construirse con base en el modelo comunicacional, porque según él, el lenguaje ha sido subsumido por el capital. El argumento esgrimido es que el

este enfoque la inversión en investigación y desarrollo se asocia a la formación de este capital humano. Autores como Romero, López y Martínez (2018: 17) señalan que México pasó de la posición 45 en el Índice de Economía del conocimiento en 1995 hasta la posición 72 en 2012, lo que evidencia que las políticas públicas no han aumentado la formación de capital humano ni el desarrollo de capacidades cognitivas. Jola Sánchez (2013: 65) propone un modelo de panel (efectos fijos) de al menos 124 países con periodos entre 1970 y 2007 a partir de un marco teórico neoclásico para comprobar que una mayor cantidad de servicios intensivos en capital humano y tecnología aumenta la competitividad y el crecimiento mediante aumentos en la productividad. Sostiene que en los países de ingresos altos los principales aumentos en la productividad provienen de la disponibilidad de servicios, mientras que en los países de renta media y baja la productividad aumenta principalmente con la manufactura y la agricultura respectivamente.

lenguaje no puede ser privatizado porque se soporta en una masa parlante y códigos puramente internos que no pueden ser enajenados del uso público¹⁰.

De esta forma, lo que en la última década del siglo XX se expresaba como un rechazo a la constitución de un sujeto político hegemónico a partir de la relación de clase y la organización sindical debido a reivindicaciones no económicas en la acción colectiva de los trabajadores, se presenta en los teóricos del capitalismo cognitivo como una defensa de la constitución de un sujeto político con autonomía del capital a partir de la acción colectiva de las comunidades de programación libre y abierta, tanto por su forma de organizar el trabajo como por las formas de propiedad colectiva a través de las cuales se crean a sí mismos, y en las cuales la acción sindical es marginal o inexistente (Blondeau, 2004: 53)

4. Las tesis del capitalismo cognitivo y sus críticas

Según Míguez (2014: 39) los principales exponentes de las tesis del capitalismo cognitivo son los economistas Carlo Vercellone, Yann Moulier Boutang y Andrea Fumagalli, quienes sostienen que la economía del saber y la privatización de lo común ha conducido a una economía rentista, y que por ello la propiedad intelectual ha sido reforzada en tanto es el único mecanismo que permite la apropiación privada para la valorización del capital, mediante la producción de mercancías ficticias y el mecanismo de escasez artificial para limitar su difusión y reglamentar el acceso.

En este sentido, las luchas emprendidas por los activistas de la programación libre, prácticamente desde los inicios de la informática en los años 60, y que durante los

¹⁰ En este contexto de diversas formas de producción, Marazzi (2008: 45) planteó que lo que caracterizaba esta nueva economía era una teoría de valor comunicacional, ya que el trabajo manual tiende a ser desarrollado por máquinas dirigidas automáticamente mientras que el trabajo innovador, el que realmente produce valor, es el trabajo mental. Más aún, si por una parte se reduce el tiempo comandado por la máquina separada del cuerpo del trabajador, por otra parte aumenta en modo explosivo, el tiempo de trabajo vivo lingüístico-comunicativo-relacional, aquel tiempo que en la New Economy concierne a la comunicación intersubjetiva o a la cooperación creadora de valor (2008: 58)

años 90 enfrentó a grandes corporaciones como Microsoft hasta consolidarse como una comunidad colaborativa estable y en crecimiento con iniciativas como Wikipedia, tuvo una gran influencia en estos teóricos, ya que los presentan como un caso en el que se verifican las características del “General Intellect” de Marx en la medida que ponen en peligro la posibilidad de privatizar el saber colectivo a través de los derechos de propiedad intelectual (Fumagalli, 2010: 114)

Ha sido precisamente la tesis de la no vigencia de la ley del valor la que más se ha criticado de la teoría del capitalismo cognitivo. Husson (2003: 12), por ejemplo, señala la contradicción entre la propuesta de una renta garantizada, como defienden ellos, si al mismo tiempo en su teoría describen la gratuidad de los bienes por su independencia con el tiempo de trabajo (p.e. copias a bajísimo costo) y una fuerza productiva que no puede ser objeto de propiedad privada, como el intelecto general. Desconfía, además, de la orientación estratégica de esa teoría porque niega el núcleo de las relaciones de explotación.

Starosta (2012: 367), por su parte, se concentra en los productos de esta pretendida nueva forma hegemónica de los productos del trabajo, es decir, las llamadas mercancías cognitivas. Estas provienen de una peculiar estructura de costos: la producción del primer artículo implica altos costos fijos en mucha investigación y desarrollo, pero el costo de reproducción de cada unidad adicional, es decir, el costo marginal, es muy bajo o casi insignificante, y por tanto hay una ruptura con la teoría del valor porque el costo marginal es cero. Para su análisis se concentra en los sistemas informáticos ya que los mismos son utilizados como caso paradigmático entre los defensores del capitalismo cognitivo.

Charchedi (2014: 82) ha criticado a los teóricos del capitalismo cognitivo precisamente por su posición en lo que llama la lucha de clases cognitiva, la cual plantea en términos gramscianos de intelectualidad de masas e intelectual orgánico.

La propiedad sobre los medios de producción en los trabajos cognitivos, como la programación de videojuegos o el desarrollo de sistemas informáticos en general, no solo es relevante en relación con la generación de contenidos y mensajes sino también en las técnicas de producción, lo cual determina que las empresas capitalistas del sector busquen maximizar su beneficio, ya que si la ganancia no es máxima o no hay del todo, el proceso de trabajo se ve interrumpido y la producción detenida: el capital se impone a través de la técnica.

Para dicho autor, esto constituye la racionalidad del capital que se impone sobre la racionalidad del trabajo. El internet no cancela la división entre capital y trabajo sino que establece una nueva arena para el conocimiento y da nueva forma a la multiplicidad de conocimientos que expresan la contradicción capital/trabajo. Pero para analizar este fenómeno, les contesta a los defensores de dicha teoría, hay que aplicar la teoría del valor trabajo de Marx.

Precisamente en relación con la generación de contenidos y mensajes mediante sistemas informáticos, Facebook y Twitter saldrían al mercado antes de que la primera crisis del siglo XXI conmocionara la economía mundial, y desde entonces vienen consolidándose como las principales redes sociales en Occidente en la actualidad y han sido utilizadas por críticos al capitalismo cognitivo para analizar postulados suyos tan drásticos como la pérdida de validez histórica de la ley del valor debido a la aplicación del conocimiento científico a la producción capitalista (Fuchs, 2014: 27)

Recientemente, Cafentzis (2016: 41) ha cuestionado la comparación de Vercellone entre las características autonómicas de los sistemas de trabajo externalizado entre los siglos XVI y XVIII, y de los trabajadores cognitivos contemporáneos, ya que el desenlace de los primeros fue su desaparición como clase. Por eso sostiene que el desenlace de los artesanos y los autónomos contemporáneos será el mismo que sus predecesores, si estos creen que "no pueden ser reemplazados y que el valor de su trabajo es inmensurable." Más aun, señala que en la actualidad hay competencia entre los trabajadores cognitivos, ya que ese es el destino de todos los trabajadores calificados en la historia del capitalismo.

5. El marxismo ortodoxo como análisis de la clase obrera en el capitalismo cognitivo

La pertinencia y actualidad de los aportes de Braverman para analizar las tesis del capitalismo cognitivo se basa, no solo en que él estudió los orígenes de la informática sino además, en que su enfoque de la organización del proceso de trabajo bajo el mando capitalista responde a la acumulación de capital y sus leyes, lo cual implica la acción conflictiva de los trabajadores.

En el caso de la informática, ya Braverman (1998: 138) había descrito dos ciclos del desarrollo histórico de la programación informática, pero desde la perspectiva del proceso de producción de la máquina mecánica, es decir, de la máquina computadora. Más aún, observó y analizó que para 1970 la programación informática fue transformada por la introducción del control numérico, el cual fue usado para dividir el proceso en operaciones separadas, cada una representó, más o menos en términos de entrenamiento, habilidades y costos laborales por hora, el del maquinista competente,

es decir, el *principio de Babbage* – dividir los oficios para abaratar sus partes individuales- aplicado en la configuración de la revolución técnica.

Para Braverman, ese proceso se vuelve cada vez más complejo, pero cada uno de estos trabajadores requiere saber y entender mucho menos que un trabajador simple. El maquinista capacitado, por esta innovación, es deliberadamente desplazado como obsoleto, de la misma forma que el soplador de vidrio, o el telegrafista en código morse, y como regla es remplazado por tres tipos de operativos (1998: 138).

Por: 1) Un programador, 2) Un codificador, 3) Un operario más la máquina, y aquí es donde según Braverman, se aplica el *principio de Babbage*. Sin embargo, para la época en que se publica su libro la programación informática se concebía como algo inherente (incrustado) a la producción de los componentes electrónicos y su adecuado funcionamiento.

Sin embargo, es claro que la noción de “máquina abstracta” o “computadora digital” de Alan Turing está presente en el análisis de Braverman aunque no lo cite, porque ambos son contundentes en que la programación informática es la forma más desarrollada del “management científico” que consiste en dividir todo proceso de trabajo en tareas más simples que a su vez son divididas en operaciones aún más simples hasta dar con la automatización formal de un proceso de trabajo.

Por consiguiente, así como Braverman cuestiona a Taylor por no citar a Babbage, es necesario cuestionar a Braverman por no citar a Turing, ya que analizando sus discursos, y ante la evidencia de que Turing leyó a Babbage, Braverman tenía que saber que la máquina de Turing (1950: 456) busca imitar la “máquina humana” que empuja los carros en las minas de carbón, al igual que el “Motor Analítico” de Babbage (2009: 22-24).

Más aún, la máquina de Turing (1950: 436) sustituye el papel con registros escritos en renglones por registros en máquina computadora en un lenguaje, fundamentalmente binario, en varios niveles, tal y como sucede con el desdoblamiento de la producción capitalista y su contabilidad, según la narración de Braverman (1998: 87).

Adicionalmente, al incorporar a Turing en la discusión sobre la división del trabajo, se abre la posibilidad de explorar la noción de máquina en Marx y el desarrollo de una historia crítica de la tecnología computacional, tema que está fuera del alcance de esta investigación y en el que se avanzó muy poco en la antigua Unión de Repúblicas Socialistas Soviéticas (URSS) (Dickson, 1988: 1034), y que tampoco fue abordado apropiadamente por los teóricos soviéticos de la teoría de juegos, a pesar de sus múltiples aplicaciones, dentro de la lucha de clases, en la estrategia militar y las relaciones internacionales (Robinson, 1970: 299)¹¹

Según la interpretación de Braverman, la división de tareas y la organización capitalista en general de los procesos de trabajo busca aprovechar al máximo su capacidad productiva, de lo que se sigue que los teóricos del capitalismo cognitivo, al rechazar la teoría del valor trabajo, se vean conminados a rechazar al mismo tiempo el poder de mando del capital sobre el trabajo.

Más aún, las principales tendencias contradictorias en el capitalismo cognitivo, como la precarización de las condiciones del empleo y el aumento de la desigualdad en el ingreso, se pueden explicar mediante la ley general de la acumulación capitalista y la

¹¹ La noción de lucha de clases está atravesada por la teoría de juegos, ya que la misma es una técnica fundamental de guerra. En el pensamiento marxista, Gramsci y la escuela italiana operaísta se destacan por considerar estrategias de lucha y diferenciarse entre sí por ellas, lo que en palabras Modonessi (2010: 151) es el análisis de “la guerra de movimiento, es decir, la hipótesis insurreccional de los consejos de fábrica a finales de la década del diez y la guerra de posiciones, es decir el proceso de progresivo arraigo político, territorial y cultural del partido nuevo togliattiano en la segunda posguerra.”

lucha de clases, ya que como sostiene Braverman (1998: 57-58), estas tienen como consecuencia una estructura que polariza extremadamente a aquellos empleados cuyo tiempo es infinitamente valioso (los trabajadores cuyo conocimiento es conservado) y aquellos cuyo tiempo vale casi nada (reducidos a trabajo simple).

Sin embargo, el crecimiento de las actividades relacionadas con la noción de servicios -entre los que se cuenta la programación informática y el internet- en la contabilidad nacional de todos los países a partir de la segunda mitad del siglo XX, y la necesaria desindustrialización del empleo que lo acompaña, ha estado marcado por la desindustrialización de la fuerza de trabajo y por sucesivas derrotas de la clase obrera (como la disolución de la URSS) que llevaron al marxismo a su mayor crisis de credibilidad en los 90 (De la Garza, 1993: 115)

Esto cuestiona la pertinencia del análisis de clase en dicha industria desde el punto de vista de la lucha del proletariado, es decir, desde la tradición marxista propuesta por Lukács como ortodoxa (1978: 32), y al mismo tiempo reivindica la noción de acción colectiva introducida por los teóricos de la acción racional como Olson (1971: 6) y desarrollada por sociólogos del posfordismo, como Melucci (1999: 34), para conceptualizar las comunidades de programación libre y abierta.

De hecho, la acción colectiva asociada al gremio de los informáticos ha sido la emprendida por las comunidades de programación libre a partir de los 70 y constituidas en 1984 a partir del impulso de Richard Stallman y la Fundación GNU (2004: 20)

El auge de estas comunidades de programación libre y abierta, de manera extendida y decisiva durante los 90, atrajo la atención de autores neoclásicos, primero con la conclusión apresurada de que era una Nueva Economía, y luego con la vertiente inaugurada por Lerner y Tirole (2002: 223) en su aplicación de la teoría de juegos para

explicar porqué existen dichas comunidades de programación libre y abierta, pero ha sido poco explorada por autores marxistas.

Esta carencia para el pensamiento marxista, no resuelta con la analítica de clase y el marxismo analítico en autores como Roemer (1982: 518) o Carpenter y Matthews (2003: 3), se puede deber en parte a su tratamiento ahistórico denunciado por Harvey (1986: 689) y en parte a las válidas objeciones De la Garza (1994: 365-366), pero además a que no le ha prestado atención adecuada a cómo la informática ha introducido el control numérico en los diversos oficios y profesiones, y cómo con ello los ha transformado, en la medida que, como ya observara Braverman en los 70, es una técnica que busca aplicarse a cualquier proceso automatizable (1998: 138)

Por esto, esta tesis trata sobre cómo incide la organización específicamente capitalista del proceso de trabajo en el poder de negociación de los trabajadores, y se inserta en el debate con la teoría del capitalismo cognitivo. Para hacerlo, analiza el proceso de trabajo, introduciendo la categoría clase, en el desarrollo de sistemas informáticos como caso paradigmático de la tesis de la crisis del teoría del valor trabajo de autores como Marazzi (2014: 48), Blondeau (2004: 47), Fumagalli (2010: 262), Moullier Boutang (2012: 171), en la que se soporta el resto de postulados relacionados con la noción de capitalismo cognitivo.

Sin embargo, es menester destacar las dos principales críticas a la comprensión de Braverman sobre los procesos de trabajo.

En primer lugar, a este autor marxista se le cuestiona no prestar suficiente atención al aspecto subjetivo del trabajo, es decir, a las diferencias internas entre los trabajadores y el aumento de las técnicas más humanizadas de gestión de mano de obra

y por tanto no enfatizar la cualificación como fuente de identidad de los trabajadores (García Calavia, 1999: 199)

En segundo lugar, se le cuestiona que el taylorismo analizado en su obra es una etapa superada. En este sentido, la rigidez taylorista de organización del trabajo en relación con la separación entre la concepción del proceso de producción y la ejecución de tareas estandarizadas y formalmente prescritas, hizo que otras tecnologías de gestión, como el toyotismo¹², tuvieran una mayor afinidad con la flexibilización laboral que predominó en algunas de las actividad que crecían junto al sector servicios en los 80 y 90, como la informática o la publicidad.

Pese a eso, es necesario recuperar la crítica humanista al economicismo taylorista por parte de Braverman, tanto del capitalismo occidental como del comunismo soviético¹³, y extenderla a la comprensión de los teóricos del capitalismo

¹² Alvarez (2012: 195) define toyotismo como una tecnología de gestión del trabajo cuya racionalización acontece mediante dispositivos espaciales y de organización técnica sobre los trabajadores y sus prácticas, se implementa mediante la regulación de las organizaciones sindicales, ya que el conflicto es concebido como limitante de la productividad.

¹³ La organización del proceso de trabajo en la URSS fue duramente criticada por Braverman debido a las similitudes que encontró entre las practicas capitalistas tradicionales y las soviéticas, lo que reafirmaba, según él, la conclusión apresurada de que no hay otra manera en la que puede ser organizada una industria moderna (1998: 10-11). En este sentido, Wren (1980: 9) afirma que durante los 20 en la URSS Polakov aplicó ideas radicales de Taylor como el "capataz funcional", una especie extrema de control técnico en el cual se delega el management a técnicos especialistas, cada uno de los cuales es responsable de una fase del trabajo de la fábrica, lo que fue muy ventajoso, pero que en los 30 hubo una rebelión sobre la autoridad de los managers, de tal forma que la purga estalinista acabó con el uso científico del "management" en esa misma década. Davis (1989: 486) sostiene que Stalin apoyó la organización del proceso de trabajo en las fábricas, granjas y minas de acuerdo con el "*Stakjanovismo*, en tanto que estrategia soviética alternativa a los enfoques autonomistas y "manageriales"- como el Taylorismo y la Tabla de Grantt- por sus implicaciones para las empresas colectivas. Y aunque considera que las estadísticas eran exageradas, reconoce evidencia de que aumentó la productividad del trabajo entre 1935 y 1940 en la industrialización soviética, especialmente en la ultra-stakjovinizada industria extractiva de carbón. Precisamente es este énfasis productivista, tanto del enfoque de Polakov como del Stanjanovismo, el obheto de la crítica de Braverman. Más recientemente, Kelly (2016: 310-311) recupera esta discusión sobre la organización del proceso de trabajo para reivindicar el aporte de Polakov a la historia del management en tanto convencido taylorista de que el "management científico" era un medio para el socialismo al proveer de la información y las capacidades de gestión en todos los niveles, y crear un ambiente de trabajo y condiciones de trabajo basadas en mediciones científicas con objetivos definidos en conjunto con los trabajadores. Así, considera que la Revolución Bolchevique le permitió a este autor demostrar en la URSS la importancia de organizar la producción para la sobrevivencia y no por la ganancia.

cognitivo respecto del efecto de la aplicación de la ciencia a la producción, y específicamente al proceso de trabajo.

6. La autonomización del trabajo en relación con el poder de mando capitalista

La interpretación de los teóricos del capitalismo cognitivo, y sus pioneros como Negri y Bologna, sobre los cambios en la organización del proceso de trabajo ante la propagación de las comunidades de programación libre y abierta, su modelo tecnológico y su forma de propiedad colectiva sobre el código fuente, es que en esta etapa histórica el trabajo se autonomiza del poder de mando capitalista sobre el trabajo (Fumagalli, 2010: 202).

En este sentido, las comunidades de programación libre representarían la manifestación más evidente de esta autonomización del trabajo en relación con el capital, ya que han desplegado una serie de características que ponen en tela de duda el derecho de propiedad intelectual de la empresa, muchas de ellas en la organización del proceso de trabajo que han implicado que sus productos sean elaborados colectivamente y deben ser libres, no necesariamente gratuitos pero sí sin derechos de autor (Moullier Boutang, 2012: 179)

De esta forma, esta autonomía del trabajo en relación con el capital, que altera las formas convencionales de organización del proceso de trabajo del capitalismo industrial (fordismo), es comprendida a partir de la conjunción de dos factores específicos de esta emergente etapa histórica.

Por un lado, en el capitalismo cognitivo la no vigencia de la ley del valor implica la imposibilidad de contabilizar las horas de trabajo porque no se pueden separar las horas de trabajo de las de no-trabajo, fenómeno presente en el carácter lúdico de la programación libre, pero también en lo que Morini (2007: 42) denomina la

“feminización del trabajo cognitivo”, es decir, a la apropiación por parte de la clase capitalista de la polivalencia, multiactividad y cualidad del trabajo femenino adquirido en su experiencia en labores de producción y domésticas¹⁴.

Por otro lado, en el capitalismo cognitivo el conocimiento y el control sobre el proceso de trabajo concreto aparecen como facultades inseparables del trabajador y no como una característica del capital constante impuesta al proceso de trabajo de los programadores informáticos, de tal manera que desempeñan su tareas de manera competente y eficiente sin la supervisión ni dirección capitalista, a diferencia del capitalismo industrial donde el management científico determinaba quien poseía el conocimiento, y quien ejercía el control, durante el proceso de trabajo.

Así, en el capitalismo cognitivo es el cuerpo humano el depositario del capital fijo (Vercellone, 2004: 72), y que en el caso de la programación informática se manifiesta a través de las cualidades para leer y escribir código informático como parte constitutiva, e inseparable, de la subjetividad del programador.

Esto es posible, según Jacob y Míguez (2016: 68), porque las relaciones de producción entraron en contradicción con las relaciones de propiedad debido a la aplicación de la ciencia. Por eso las licencias de código libre han aparecido como expresión de una fuerza productiva denominada intelecto general¹⁵ característica de esta época histórica, marcada por la programación informática como caso paradigmático

¹⁴ Marx encuentra precisamente esta contradicción que supone el modo de producción capitalista en relación con el tiempo de trabajo y el tiempo de vida cuando señala que “la paradoja económica de que el medio más poderoso para reducir el tiempo de trabajo se trastrueque en el medio más infalible de transformar todo el tiempo vital del obrero y de su familia en tiempo de trabajo disponible para la valorización del capital.”(1984: 497)

¹⁵ Miguez (2014: 31) sostiene que Negri rescató el concepto de General Intellect del Marx de los Grundrisse. Allí se señala que a medida que se desenvuelve la gran industria, la riqueza va a depender menos del tiempo de trabajo y más de la potencia productiva del saber social, de lo que depende en última instancia el estado general de la ciencia y la tecnología. El General Intellect deviene cada vez más en atributo del trabajo vivo en la medida que consiste cada vez más en prestaciones lingüísticas, o sea, a

En consecuencia, la no validez histórica de la ley del valor implica que el capitalismo cognitivo es rentista, y los derechos de propiedad intelectual operan como la forma social que adopta la ganancia extraordinaria relacionada con dicha tecnología al transformarse en renta, y de ahí que Moulier Boutang (2004: 127) defienda que la propuesta de renta universal podría tener el mismo efecto de desplazamiento que tuvo el salario sobre el trabajo esclavo.

De esta forma, lo novedoso de la etapa histórica contemporánea del capitalismo sería que aparece una forma de propiedad, aparentemente incompatible con el modo de producción capitalista y su forma específica de organizar el proceso de trabajo, que dota a la fuerza de trabajo de una autonomía en la esfera productiva: las licencias libres de ciertos sistemas informáticos (Cardoso y Vercellone, 2016: 45).

Esta forma de propiedad colectiva no privatizable les permite a los desarrolladores utilizar los códigos fuente, (librerías y otros tipos de códigos disponibles en la red de las comunidades de desarrollo), como medios de producción bajo el control de la fuerza de trabajo, y no bajo los criterios de la organización del proceso de trabajo del management científico, es decir, automatizados del poder de mando del capital.

De esta manera el proceso de trabajo en la programación informática se autonomiza del “management” capitalista porque el capital constante no es una barrera de entrada para la competencia ni es determinante en la acumulación de capital ya que esta se realiza como acumulación de conocimiento, y de ahí que denominen al periodo histórico actual “capitalismo cognitivo” (Míguez , 2014: 33)

medida que el proceso de trabajo es más locuaz y menos “mudo”, taciturno o tímido y donde se destacan cada vez más los aspectos lingüístico-relacionales y comunicativos. Por ello, para este autor, la crisis de la ley de valor que ocasiona la aplicación de la ciencia a la producción se manifiesta como "una independencia con relación al tiempo de trabajo impuesto por el capital".

Capítulo 2. Planteamiento de la investigación

1. Introducción

Con base en la exposición del contexto en el que se inscribe la investigación abordado en el capítulo anterior, en este se plantean los fundamentos teóricos y metodológicos del problema que se busca resolver, a partir de lo dicho antes sobre las tesis de los teóricos del capitalismo cognitivo y de acuerdo con la perspectiva planteada aquí como marxismo ortodoxo.

2. Planteamiento del problema

Los teóricos del capitalismo cognitivo sostienen que el capitalismo se desarrolla históricamente en etapas sucesivas, a las que denominan el capitalismo comercial, el capitalismo industrial y el capitalismo cognitivo, acontecidas entre el siglo XVII hasta la actualidad, como en una especie de onda larga definida a partir de la forma específica que asume el capital en su proceso de acumulación.

Así el capitalismo comercial se habría desarrollado en los orígenes de dicho modo de producción como proceso histórico en el que la riqueza se acumulaba bajo la forma primordial de capital mercantil. El capitalismo industrial habría partido de las condiciones creadas en la etapa anterior, pero con la diferencia de que la riqueza se acumulaba a través del capital constante.

Finalmente, el periodo histórico contemporáneo habría surgido del agotamiento de las fuentes de acumulación de la etapa anterior, por lo que la riqueza tendería a acumularse a partir del capital humano, es decir, a partir del conocimiento, de ahí la denominación de capitalismo cognitivo.

Por tanto, en el capitalismo cognitivo el capital no se produciría de acuerdo con la cantidad de horas de trabajo, sino de acuerdo con la intensidad en conocimiento acumulado en el cuerpo del trabajador y manifestado a través de sus capacidades lingüísticas-relacionales. Este cambio habría implicado que la teoría del valor trabajo resulte incapaz de explicar los precios de las mercancías cognitivas, debido a que el contenido del valor ya no serían las horas de trabajo humano en tanto que dicha magnitud sería tendencialmente cero debido a la aplicación de la ciencia y el conocimiento a la producción.

Como consecuencia de ello, la regulación de los precios de producción no provendría de la cantidad de horas de trabajo socialmente necesarias sino de los derechos de propiedad intelectual sobre el conocimiento para producirla, como una barrera impuesta por la clase capitalista que impide la gratuidad generalizada de las mercancías cognitivas.

Esto supone que los precios expresarían relaciones rentísticas y creativas (autovalorización), es decir, autonómicas, pero ya no de clase como sostiene reiteradamente Marx, tanto cuando argumenta que la mercancía fuerza de trabajo produce más valor del que consume (1984: 235), como cuando analiza la reproducción y descubre que en su curso todo el valor de capital adelantado se transforma en plusvalor capitalizado, es decir, se invierten las relaciones de propiedad a espaldas de la clase trabajadora (1984: 720).

Sin embargo, los intentos por ofrecer una teoría del valor diferente a la basada en las horas de trabajo, ya sea lingüística, semiótica o comunicativa, como propusieron Negri y Lazaratto (2001:38), han resultado tan infructuosos como la práctica del cálculo de los útiles como unidad de medición del valor, porque conduce a

una confusión entre la autorreferencialidad propia de las palabras y la de los instrumentos financieros (Marazzi, 2014:34).

Para estos teóricos del capitalismo cognitivo, si el valor de cambio es una manifestación del valor, entonces el valor lingüístico también: así como el significado de las palabras se constituye a través de convenciones, el precio de los activos financieros está determinado por un comportamiento gregario de la opinión según el cual los valores de los títulos cotizados en bolsa hacen referencia a sí mismos y no a algún valor económico subyacente.

De esta manera, esta subsunción del lenguaje al capital parece conectar la convención lingüística con el concepto keynesiano de profecía auto-realizadora: lo que los agentes piensan y el modo en que representan el mundo parecer tener un efecto sobre los precios, es decir, las cualidades lingüísticas y relacionales del cuerpo humano dotan a las creencias de los agentes de un rol creador de valor bursátil.

Sin embargo, las cualidades lingüístico-relacionales del cuerpo humano siempre han estado bajo el comando de los intereses capitalistas durante la jornada laboral, tanto en el contexto interior de las fábricas y granjas como en las actividades creativas y tomas de decisión, por lo que no representa una novedad que, ante el aparente giro lingüístico, justifique una nueva teoría del valor basada en la significación lingüística, semiótica o comunicativa¹⁶ y desechar la teoría del valor trabajo por obsoleta.

¹⁶ Investigaciones antropológicas sobre el uso del lenguaje durante la jornada laboral, como en Urciol y Ladousa (2013: 184), muestran que en los centros de llamadas los cambios tecnológicos han ampliado las posibilidades de trabajo, pero involucrando a los trabajadores en nuevos regímenes de vigilancia, como el caso de los centros ubicados en India, donde no solo se limitan las características de ejecución lingüística sino, también, la autopercepción como fuerza de trabajo para evitar ciertos comportamientos identificados con dicha población.

De esta forma, el problema de esta investigación surge alrededor de las paradojas, contradicciones y confusiones de la tesis según la cual en el capitalismo cognitivo hay una crisis de la ley del valor y de la teoría marxista sobre la que se fundamenta.

Puntualmente, en esta investigación se pone en cuestión el corolario de que dicha crisis de la ley de valor implica que en el capitalismo contemporáneo haya una tendencia hacia la hegemonización de una nueva fuerza productiva con su respectiva forma de propiedad que autonomiza a la fuerza de trabajo del poder de mando del capital. Lo que en palabras de Rengifo (2017: 90) se expresa como que: “la etapa donde el capital controlaba todos los aspectos de la producción es una época superada. Hoy estamos asistiendo a la transición en donde el trabajo está pasando a controlar y a definir, de manera creciente, al capitalismo”.

3. Pregunta de investigación

La pregunta general que busca responder esta investigación es ¿de qué manera afecta la organización capitalista del proceso de trabajo en la programación informática el poder de negociación de dichos trabajadores, en términos del control y autonomía de la fuerza de trabajo, durante la jornada laboral?

En la medida que el poder de negociación de los trabajadores informáticos está atravesada por acciones colectivas, esta pregunta a su vez implica considerar ¿cómo la organización de dicho proceso de trabajo incide en las experiencias de subordinación, antagonismo y autonomía de dichos trabajadores en relación con su explotación como fuerza de trabajo?, es decir, ¿cómo incide el “management” en las capacidades de los programadores para formar coaliciones de clase con poder de negociación ante la dominación de la clase capitalista?

4. Objetivos

Con base en los planteamientos expuestos, esta tesis tiene como objetivo central analizar el desarrollo de los sistemas informáticos como caso paradigmático de trabajo cognitivo para determinar los efectos potenciales de la organización del proceso de trabajo en el poder de negociación de los trabajadores de dicha rama.

Para conseguirlo se considera indispensable, en primer lugar, analizar el proceso de trabajo en la producción de los sistemas informáticos en general, destacando cada una de las tareas en las que se subdivide, para caracterizar la división del trabajo de las comunidades de programación libre y abierta (FOSS, por sus siglas en inglés) en las cuales los teóricos del capitalismo cognitivo han señalado evidencia de autonomización del poder de mando del capital.

En segundo lugar, se considera pertinente precisar los acontecimientos que marcan puntos de inflexión de la trayectoria histórica de las comunidades de programación libre y abierta (FOSS) para determinar si estas se están volviendo hegemónicas en los sectores tecnológicos, como es la conjetura de Cardoso y Vercellone (2016: 46) o más bien juegan un papel de empresa periférica, como sostiene Ross (2013: 204).

Finalmente, se considera menester analizar el proceso de formación de valor y de valorización en el caso de los sistemas informáticos de gestión de aprendizaje (LMS) para determinar si la formación del precio de venta tiene o no a las horas de trabajo como su principal variable explicativa, es decir, si en dicho caso específico se verifica o no la conjetura de los teóricos del capitalismo cognitivo sobre la crisis de la ley de valor.

5. Hipótesis

Como la producción de los sistemas informáticos es realizada principalmente por corporaciones capitalistas globales, como Apple, IBM, Microsoft, o Google, se parte de la hipótesis de que sus metodologías de desarrollo responden directamente al principio de Babbage, como mostrara Braverman en su estudio temprano al respecto, y por consiguiente que su organización del proceso de trabajo se encuentra explícitamente bajo el mando capitalista y su interés como clase de obtener una ganancia.

Por ello, se tratará de verificar la tesis de los teóricos del capitalismo cognitivo de que las comunidades de programación libre y abierta utilizan metodologías de desarrollo que no responden al interés capitalista de obtener una ganancia, y por tanto, su autonomía se manifiesta en que las horas de trabajo no constituyen el criterio a partir de cual se forma el precio mediante el cual circulan y se distribuyen entre sus usuarios.

6. Metodología

La metodología seguida es de carácter cualitativa y descriptiva en términos estadísticos, y puede ser definida como el análisis de un caso paradigmático a partir de un ejemplo práctico. Por paradigma se entiende la definición de Agamben (2010: 3-5), quien propone que son un objeto singular equivalente para todos los de una misma clase que, cuya exhibición, permite volver inteligible el conjunto del que forma parte.

La noción de análisis proviene de los aportes de Hjemslev quien lo define como la división de un objeto en sus partes para identificar sus dependencias e interdependencias. Según este autor, la totalidad no consta de cosas sino de relaciones, y de que no es la sustancia sino sus relaciones internas y externas las que tienen existencia científica (Hjemslev, 1971: 39-41)

Esta definición de análisis proviene de la teoría lingüística donde se toma al texto como dato, la cual es pertinente a este abordaje en tanto la programación informática se fundamenta en la manipulación de lenguajes.

Así, para el análisis del proceso de trabajo son los textos de “management” en programación informática, y sus hallazgos, los que se toman como datos de referencia sobre la división del trabajo en el desarrollo de sistemas informáticas, tal que pueda seguirse el principio de división de esos datos para ofrecer una descripción exhaustiva de la información disponible como evidencia sobre las tesis de los teóricos del capitalismo cognitivo bajo estudio.

Además, para el análisis del proceso de trabajo se considera necesario utilizar la teoría sociológica marxista de Modonessi (2010), la cual no solo ofrece un marco analítico, a partir del concepto de experiencia en Thompson, para la categoría de clase sino que además incorpora la noción de autonomía como uno de sus conceptos centrales.

Esto permite comprender la relación control-autonomía dentro del campo de la programación informática como una experiencia de clase, y favorece explorar configuraciones de subalternidad-antagonismo-autonomía dentro de las comunidades de programación libre y abierta para describir sus dinámicas organizativas y sus estrategias colectivas.

En este sentido, se sigue la propuesta epistemológica de Braverman (1998: 216) de que el estudio del proceso del trabajo, a partir de sus tiempos y actividades, se presenta como la contrapartida del estudio de los precios, y los valores, para comprender los movimientos de la acumulación capitalista.

Con base en esto, se identifican los mecanismos por medio de los cuales la tecnología informática incide en las experiencias de subjetivación política de los programadores en su constitución como clase, y a partir de ello identificar la estructura a través de la cual esta forma de organización capitalista del proceso de trabajo incide en el poder de negociación de sus trabajadores.

Para dicho estudio, se aplicarán las categorías utilizadas por Marx en su análisis del proceso de trabajo y del proceso de formación de valor y de valorización en los sistemas informáticos de programación libre y abierta, para debatir si la ley del valor moldea dicho proceso de trabajo o no.

Para el análisis del proceso de valorización se consideran específicamente los sistemas informáticos para la gestión del aprendizaje (LMS) a partir de datos observados, para lo que se consultan fuentes académicas que especifican la relación entre funcionalidades y precios, así como cotizaciones de estos sistemas en páginas disponibles en internet. En este sentido, se procede con herramientas estadísticas descriptivas pero no inferenciales ya que no se pretende desarrollar una solución general para todas las mercancías capitalistas ni del proceso global de producción, sino un ejemplo práctico que permita aproximar las relaciones fundamentales para el pensamiento marxista utilizando las horas de trabajo como unidad de medida.

Así, para estos aspectos cuantitativos de la investigación se han consultado diversas fuentes de proveedores de dichos sistemas en las que se puede cotizar en línea mediante la técnica de inspección.

De esta manera, no se ha optado por un enfoque cuantitativo para la identificación de un precio de mercado único, ya que ello podría constituir un sesgo en relación con la axiomática neoclásica, pero tampoco se han realizado estimaciones de

precios de producción por tratarse del análisis de una única rama, lo que impide precisar las transferencias de valor entre las ramas de menor composición orgánica hacia las de mayor composición. Por ello, este abordaje es descriptivo pero atravesado por la discusión teórica sobre la transformación de los valores en precios, dentro de los debates contemporáneos del pensamiento marxista.

Finalmente, para situar la investigación en el contexto Latinoamericano y su situación geopolítica en relación con Estados Unidos, como epicentro de las tecnologías informáticas y su integración en los procesos de desarrollo económico, se parte de que los precios y salarios se corresponden con la articulación territorial de la división internacional del trabajo en la programación informática en el eje americano constituido por Berkeley, California (EE.UU), Ciudad de México, México, y Heredia, Costa Rica. Esto permitirá abrir el debate respecto de las tendencias futuras sobre la automatización y la robotización en dicho campo debido a la aplicación de la inteligencia artificial en la región.

7. Variables de investigación y técnicas de recopilación y manipulación de la información y los datos

La noción de control es central en la literatura de “management” en ingeniería de sistemas informáticos, así como lo hicieron Babbage, Marx y Braverman en su momento, por lo que se utiliza como variable cualitativa para precisar el poder de mando del capital sobre el trabajo.

La noción de autonomía del poder de mando del capital debe ser entendida conceptualmente en relación con la teoría del valor trabajo, es decir, como un fenómeno contenido dentro de los efectos de la organización capitalista del proceso de

producción manifiesto en el poder de negociación de los trabajadores, y por tanto como variable en este análisis.

De esta manera, control y autonomía se utilizan como variables para diferenciar la relación de los procesos de trabajo con el poder de mando del capital, bajo el supuesto que el control capitalista busca regular los contenidos, los modos de ejecución, los tiempos y la intensidad del proceso de trabajo para que cada obrero carezca de poder de negociación .

Las definiciones y métricas que permiten definir el concepto de control en el campo de la informática provienen de publicaciones especializadas en dicho campo, por lo que es directamente de esta disciplina que se obtienen los datos y evidencia sobre la hipótesis aquí expuesta mediante la técnica de análisis de contenido.

Para ello, se realiza una selección de literatura mediante un proceso iterativo de búsqueda, revisión y resumen basada en los criterios de pertinencia, originalidad y rigurosidad en publicaciones en revistas y libros especializados en desarrollo de sistemas informáticos y su proceso de trabajo.

De esta forma, para el análisis del proceso de trabajo se sigue la técnica de análisis de contenido de los resultados en investigaciones empíricas y académicas consultadas sobre “management” en ingeniería en sistemas, el cual consiste en “decodificar el discurso de los mensajes contenidos en los documentos analizados, mediante la identificación de posibles contradicciones, ausencias o presencias, intenciones explícitas o implícitas, énfasis, objetivos perseguidos, destinatarios, etc (Fernández, 2002: 40).

El análisis de resultados, consecuentemente, hace énfasis en la estrategia intertextual que consiste en “determinar el sentido virtual de los textos mediante su

relación con otros textos, bajo el método discriminativo de agruparlos en distintos dominios analíticos y poder realizar comparaciones entre ellos”(Fernández, 2002: 43)

Para el análisis del proceso de formación de valor y del proceso de valorización se utiliza la inspección como técnica de observación de los precios de los factores productivos involucrados en programación informática, así como del precio de venta de los sistemas de gestión de aprendizaje (LMS).

En este sentido se utilizan herramientas de estadística descriptiva, como medias, modas y medianas, así como razones matemáticas, como proporciones y semejanzas en tanto que estructuras relacionales de fundamento lógico que permiten mostrar las interdependencias entre las partes en las que se divide el objeto de estudio.

Los datos territoriales se obtienen de los institutos de estadísticas de cada país, así como de informes y estudios aplicados al campo de la programación informática, y se manipulan mediante matrices en Excel y se calculan las relaciones fundamentales para el pensamiento marxista, como la tasa de explotación y la composición valor en la composición orgánica de capital, pertinentes para alcanzar satisfactoriamente los objetivos planteados en esta investigación.

Los precios nominales de los factores objetivos y subjetivos del proceso productivos de los tres territorios considerados son convertidos a pesos mexicanos de julio del 2019, `pero se ofrece el detalle al respecto en el cuarto capítulo sobre el proceso de formación de valor y valorización con el fin de favorecer la comprensión y la lectura de este informe.

8. Limitaciones de la investigación

La principal limitación en esta investigación proviene de que se formula desde la disciplina económica pero exige utilización de conceptos y métricas propias de la programación informática. En este sentido se ha implementado la técnica de observación participante mediante dos cursos de programación en R: uno virtual en la plataforma Coursera impartido por la UNAM durante mayo del 2019 y otro presencial en la FES Aragón en junio del 2019. Además, se ha consultado el criterio de especialistas con programadores informáticos de la empresa Solvo, radicada en Heredia, Costa Rica.

Otra limitación proviene de la metodología en tanto que la misma plantea que no es posible agotar todas las relaciones de determinación que pueden establecerse entre las partes de un análisis paradigmático, ni tampoco toda la generalidad del caso singular, ya que como sostiene Agamben (2010:6) es como en un campo magnético: no se trata de magnitudes extensivas y graduales, sino de intensidades vectoriales, es decir, tendencias.

Por tanto, la investigación se limita a las determinaciones en el caso de la programación informática entre la organización del proceso de trabajo y el poder de negociación de los trabajadores informáticos en un ejemplo práctico, pero no exime de la existencia de determinaciones entre otras partes del objeto de estudio y sus relaciones potenciales con otros objetos de estudio, lo cual queda fuera del alcance de esta tesis.

Así, esta investigación se acota a establecer un marco analógico en el que la organización del proceso de trabajo en informática afecta el poder de negociación de los trabajadores informáticos en los términos cualitativos de control y autonomía, pero no se propone medir esos efectos en términos cuantitativos, como en la propuesta de Dinardo, Fortin y Lemiux (1996: 1005) ya que la relación salarial y la organización

sindical no necesariamente caracterizan la dinámica de remuneraciones y acción colectiva en las comunidades de programación libre y abierta.

Se considera que el no asumir el modelo de inferencia lógica, que va de lo particular a lo general, en esta investigación es una limitación en términos de extrapolación de resultados, pero al mismo tiempo es una ventaja para la misma porque es un movimiento que va de la singularidad a la singularidad, y las comunidades de programación libre y abierta parecen representar una singularidad histórica dentro de las teorías de la división del trabajo. Por ello esta investigación se limita a la analógica del ejemplo que “transforma cada caso singular en ejemplar de una regla general que nunca puede formularse a priori ”(Agamben, 2010: 7)

En este sentido, la ausencia casi total de sindicalización entre los programadores informáticos - con excepciones como Argentina - impide utilizar los enfoques que miden el poder de negociación de los trabajadores mediante la relación entre sindicalización y nivel salarial, y más bien dicha ausencia podría ser efecto de la organización capitalista del proceso de desarrollo de sistemas informáticos, es decir, que esta tesis podría ofrecer una explicación de la ausencia relativa de experiencia de lucha de clases a través de la acción colectiva sindical en dicha rama.

Más aún, el análisis de la relación entre la organización del proceso de trabajo y el poder de negociación a partir del sindicalismo, representa una limitación conceptual respecto de la historicidad de las comunidades de programación libre desde el enfoque marxista. Por tanto el actual esfuerzo debe permitir “hacer inteligible una serie de fenómenos cuyo parentesco se le había escapado o podía escapar a la mirada del historiador”(Agamben 2010: 12)

Por ello, el marco analógico planteado en esta investigación busca favorecer que en futuros abordajes del tema pueda precisarse el efecto de la aplicación de las tecnologías informáticas en otros procesos de trabajo, donde simplifica sus tareas y sitúa a los trabajadores como usuarios de dichos sistemas en diferentes niveles de competencia. Futuras investigaciones podrán tomar los resultados obtenidos de la aplicación de la informática en la informática misma, para abordar la amplia variedad de campos en los que se utiliza e identificar sus efectos para los trabajadores en términos de sus remuneraciones y su poder de negociación. Esas dificultades exigen investigaciones más amplias y con más recursos.

La utilización de disposiciones metodológicas relacionadas con el análisis semiótico-lingüístico, como el análisis de contenido y el caso paradigmático, no se considera una limitación de la investigación, ya que el objeto mismo bajo estudio, la programación informática, constituye una práctica humana fundamentada precisamente en el dominio y manipulación de lenguajes como fenómeno cultural.

Sin embargo, se debe aclarar que no se busca extrapolar los hallazgos propios de la semiótica y la lingüística respecto de las palabras y la significación, ni tampoco de la sociología del trabajo, a los fenómenos económicos sino aprovechar los puntos de encuentro entre esas diversas disciplinas para ofrecer una comprensión menos economicista y más abierta a futuras investigaciones desde otros campos del saber humano.

Finalmente, en relación con el ejemplo práctico propuesto, sus mediciones representan un esfuerzo por utilizar datos observados para mostrar un aspecto de las relaciones estructurales dentro del pensamiento marxista, por lo que no resuelve las dificultades que supone la existencia de una tasa media de ganancia para toda la

economía configurada alrededor de los precios de producción, y su determinación a partir de las diferentes composiciones orgánicas de capital entre ramas.

Capítulo 3. Análisis del proceso de trabajo de programación informática

1. Introducción

En este capítulo se analiza el proceso de trabajo en los sistemas informáticos en general, e introduciendo el caso de los sistemas de gestión de aprendizaje LMS (por sus siglas en inglés), analizado en el siguiente capítulo. Se procede aplicando la división teórica propuesta por Harry Braverman entre concepción y ejecución de dicho proceso. Se clasifican las tareas identificadas en diversos textos sobre el desarrollo de los sistemas informáticos en estas dos categorías y se describen los dos grandes paradigmas de la programación informática: el privativo, y el libre y abierto, dentro del que destacan los denominados cinco casos de éxito (GNU, Linux, Apache, Mozilla y MySQL).

El análisis parte desde los medios de producción involucrados, es decir, los factores objetivos y subjetivos del proceso productivo. Dentro de los factores objetivos o productivos, se consideran las materias primas, los medios de trabajo, los materiales auxiliares y el edificio.

En relación con los factores subjetivos, estos se dividen en tareas de concepción y ejecución, como propone Braverman, ya que es mirando con detenimiento cada una de estas tareas, que puede dilucidarse si existe la autonomización del proceso de trabajo del poder de mando del capital, es decir, si existe una ruptura con la subordinación capitalista, o puede ser comprendido por algunas de las formas

descritas por Marx (1984: 448), a saber; i) industria domiciliar, ii) industria artesanal, iii) manufactura, iv) sistema fabril, iv) industria maquinizada.

2. El valor de uso informático

El resultado obtenido luego de un proceso de trabajo en programación informática es un valor de uso, es decir, una combinación de cuerpos físicos y sus sustancias químicas en medidas sociales para indicar cantidades útiles para el ser humano, y sus diversos fines. Estos productos del trabajo son tan variados que conviene denominarlos valores de uso informáticos e incluir todos los tipos de sistemas informáticos conocidos, desde los lenguajes de programación hasta las aplicaciones para teléfonos móviles.

El valor de uso informático es el resultado de una conjunción de tareas denominadas proceso de desarrollo y adopta tres formas generales: i) los lenguajes de programación informática (C++, Python), ii) los sistemas operativos de servidores y computadoras (Windows, Mac, Linux), iii) los sistemas de aplicaciones, como los paquetes de oficina (Microsoft Office, LibreOffice), los gestores de recursos empresariales (ERP, por sus siglas en inglés) o los sistemas de gestión de aprendizaje (LMS, por sus siglas en inglés) como el LMS365 para Office 365 de Microsoft, Blackboard Learn, Moodle o Claroline.

De esta forma, el valor de uso bajo estudio en esta tesis son los “sistemas informáticos” los cuales consisten en un conjunto de líneas de código en lenguajes de programación informática que permiten que los componentes electrónicos de las computadoras mecánicas ejecuten diferentes tipos de tareas automáticas a partir de entradas (input) para generar un resultado (output) esperado.

El caso de los sistemas de gestión de aprendizaje, LMS, se clasifican entre los sistemas de aplicaciones informáticas y es necesario especificar que la complejidad de producir un lenguaje de programación y la de producir sistemas operativos es mucho mayor que la de los sistemas de aplicaciones, aunque las tareas son las mismas. La mayor parte de las empresas constituidas después de Computer Usage Company (CUC), la primera en California en 1955, venden sistemas de aplicaciones informáticas a la medida y empaquetadas, entre ellas, los LMS.

En el siguiente capítulo, sobre el proceso de valorización, se volverá con más detalle sobre los módulos de este tipo de valor de uso informático. Por ahora, basta con aclarar que cuando la programación informática se utiliza para producir sistemas de aplicaciones para otros procesos de trabajo, y en este caso en los procesos educativos de la labor docente, dichos valores de uso informáticos pueden desempeñar funciones auxiliares dentro del proceso de trabajo del comprador, como los módulos de aula virtual con lecciones presenciales, o servir de medio de trabajo para los docentes que deben producir el contenido y escenificar las explicaciones ante una cámara digital. En la medida que los procesos educativos involucran al menos dos agentes, docentes y estudiantes, el fundamento lógico de estos sistemas informáticos se basa en la interacción de muchos agentes y las jerarquías que guardan entre sí.

Finalmente, debe aclararse que existen otro tipo de sistemas de aplicaciones, diferente a los lenguajes de programación, pero complementarios a ellos, y que se conocen como Entornos Integrados de Desarrollo, (IDE por sus siglas en inglés), que son desarrollados por programadores informáticos para otros programadores informáticos o para sí mismos. Entre los más destacados por su versatilidad técnica- aunque caído casi en desuso en la actualidad- se encuentra Emacks, de la Fundación GNU, mientras que entre los más populares se encuentran Visual Studio Core, para

diferentes lenguajes en el entorno de Windows y Atom desarrollado por Github en un entorno libre para cualquier sistema operativo.

3. Los factores objetivos en la producción de valores de uso informáticos

3.1 La materia prima

Para Olivier Blondeau las materias primas en la producción de sistemas informáticos son los lenguajes de programación. Más aún, considera que los "creadores de software" son trabajadores inmateriales y que esta actividad de servicios es un trabajo productivo (2004: 32-33).

Esta idea de que los lenguajes de programación, denominados artificiales según Barthes (1993 : 70) y formales por Downey, Elker y Meyers (2002: 6), se han convertido en un medio de producción en el capitalismo contemporáneo, siguiendo una tendencia hacia su hegemonización como principal factor de producción de riqueza en las sociedades capitalistas bajo la forma histórica de “intelecto general”, proviene de la confusión que introduce el pseudo-concepto de “inmaterial” (y al falso problema que conduce¹⁷) y de la influencia de la semiótica económica por parte de las escuelas europeas impulsadas por Negri y Lazaratto (2001: 42), que en última instancia, lo que proponen es sustituir la teoría del valor trabajo por la teoría del valor lingüístico (generalmente de inspiración saussuriana), declarando la primera obsoleta y la segunda universal¹⁸.

¹⁷ El término de “inmaterial” aparece en la obra de Lionel Robbins (1932:24) para rechazar la comprensión materialista de economía y justificar su comprensión instrumental de la misma, y con ello abandonar todo estudio del “bienestar material”. Es decir, conduce al falso problema de la “calculabilidad” de los precios de las diferentes mercancías a partir del análisis formal de las relaciones de intercambio (Hinkelammert y Mora, 2013: 155)

¹⁸ Miguez (2014: 31) sostiene que General Intellect del Marx de los Grundrisse. Allí se señala que a medida que se desenvuelve la gran industria, la riqueza va a depender menos del tiempo de trabajo y más de la potencia productiva del saber social, de lo que depende en última instancia el estado general de la ciencia y la tecnología. El General Intellect deviene cada vez más en un atributo del trabajo vivo en la medida que consiste cada vez más en prestaciones lingüísticas, o sea, a

La materia prima en el proceso de trabajo en el desarrollo de sistemas informáticos son las unidades lógicas con funcionamiento mecánico e impulso eléctrico, cuya unidad física es el bit o paquete de datos, y cuyos múltiplos se conocen como unidad de memoria, tanto de procesamiento de datos como de almacenamiento de los mismos. La "computadora electrónica" o "computadora digital" en el sentido de Turing (1950: 437) se constituye de tres partes: i) almacenamiento, ii) unidad de ejecución, iii) control.¹⁹

El almacenamiento es una memoria de información, y en la medida que la computadora hace cálculos una parte de su almacenamiento corresponderá con su memoria. La unidad ejecutora es la parte que realiza las variadas operaciones individuales involucradas en un cálculo. Lo que estas operaciones individuales son varía de máquina a máquina. El control es construido de tal forma que necesariamente las instrucciones deben ser obedecidas correctamente y en el orden adecuado.

Así, según Turing la información almacenada es usualmente dividida en paquetes de un tamaño pequeño moderado. En una máquina, por ejemplo, un paquete puede consistir en diez dígitos decimales. Los números son asignados a las partes del almacenamiento, en el cual varios paquetes de información son guardados de una manera sistemática. Una instrucción típica puede decir: "Sume el número guardado en la posición 6909 a aquel en 4302 y ponga el resultado de nuevo en la última posición de almacenamiento", pero esto no ocurre expresado en inglés en la máquina, sino que es

medida que el proceso de trabajo es más locuaz y menos "mudo", taciturno o tímido y donde se destacan cada vez más los aspectos lingüístico-relacionales y comunicativos. Por ello, para este autor, la crisis de la ley del valor que ocasiona la aplicación de la ciencia a la producción, se manifiesta como "una independencia con relación al tiempo de trabajo impuesto por el capital". En este sentido Eco (2000: 240) intenta explicar los tipos de cambio a partir de un "aserto meta-semiótico" basado en la autoridad de los presidentes de los bancos centrales como una "convención financiera" pero se ve obligado a reconocer que esto solo tiene validez para periodos muy breves, lo que ocasiona "gran embarazo para los especialistas en economía".

¹⁹ Nótese que la máquina automática de Marx se compone de tres partes: "el mecanismo motor, el mecanismo de transmisión y, finalmente, la máquina-herramienta o máquina de trabajo."(1984: 453)

codificado en una forma como "6809430217", donde "17" dice cual de las varias posibles operaciones será ejecutado sobre los dos números (1950: 437).

Para Turing la computadora digital fue inventada por Charles Babbage y lo utiliza como hecho para sostener que ello es evidencia de que el uso de la electricidad en las computadoras modernas ingresa como material auxiliar, y por tanto sin importancia teórica para la programación misma, ya que el “Motor Analítico” era puramente mecánico (1950: 439).

Teóricamente, la computadora digital busca imitar el comportamiento de cualquier máquina de estado discreto. Turing pone el ejemplo de una computadora que tuviera permitido 100 hojas de papel cada una conteniendo 50 líneas con espacio para 30 dígitos, la cual daría lugar a un número de estados de $10^{100 \times 50 \times 30}$, i.e. $10^{150,000}$. El logaritmo de la base 2 del número de estados es usualmente denominado “capacidad de almacenamiento” de la máquina.

En este sentido, sostiene el autor, es que debe comprenderse que una máquina “contiene 64 cintas magnéticas cada una con la capacidad de 2560, ocho tubos electrónicos con una capacidad de 1280, cuyo almacenamiento misceláneo alcanza alrededor de 300 haciendo un total de 174380” (1950: 441).

Esto es válido para la máquina Manchester que conoció dicho autor en 1949, o para una máquina “HP EliteBook 1050 G1 - 15.6- Core i7 8750H - 16 GB RAM - 256 GB SSD ” que salió a la venta en Amazon el 29 de julio del 2019 (Amazon, 2019a) y que será considerada en el siguiente capítulo sobre el proceso de formación de valor y de valorización.

Por lo anterior, las capacidades de procesamiento de datos y de almacenamiento –como principales elementos de la computadora digital- son la materia prima que se

requiere para producir valores de uso informáticos. Las líneas de código informáticas pueden ser escritas en hojas de papel pero requieren de esas capacidades artificiales de memoria mecánica para poder dotar a esas mismas líneas de su aparente poder performativo para operar la máquina computadora misma. Cuando la electrónica se aplica al manejo de paquetes de datos, constituye una de las fronteras abiertas de la programación informática y en la cual actualmente se están desarrollando las telecomunicaciones.

El carácter de materia prima de la capacidad de almacenamiento de la máquina computadora queda evidenciado en el análisis de You Lu et. al (2018: 264) sobre los problemas ocasionados por la congestión de la red de datos de los teléfono móviles . Para su experimento propone medir la congestión en el “cluster” y maximizar la utilidad en la función de los nodos. La variable clave aquí es la “caché de los clusters”, es decir, la memoria caché de los servidores a través de los cuales se distribuye la red, ya que esa memoria constituye la restricción para el procesamiento de los datos de la red móvil.

Así, su resultado está acotado a la medición del tiempo de respuesta y la cola en el tráfico para control de los datos (intensidad de tráfico) en los nodos de memoria caché.

Las pruebas experimentales del modelo las realizan con máquinas computadoras determinadas y con especificaciones del sistema informático involucrado, como el procesador, la memoria, los Mb/s de banda ancha, y paquetes de red de 64 bytes. Los resultados están en términos de Kb/s, ya esa es la unidad de control del modelo, es decir, en las unidades de velocidad de procesamiento y almacenamiento de datos.

Con esto se hace evidente que la aparente imaterialidad de las redes de datos móviles se materializa en las memorias caché hasta saturarlas y ocasionar congestión, consumir más energía eléctrica y retrasar la transmisión de los datos.

El que la programación informática parte de la capacidad de memoria para hacer que una máquina mecánica imite cualquier operación que pueda ser hecha por los humanos, tal y como lo establece explícitamente Turing (1950: 436), se comprende mejor a partir del ejemplo que utiliza Babbage para explicar lo que él denomina “división mental del trabajo” y constituye el fundamento de su “Motor Analítico”, la primera computadora digital.

Sostiene Babbage (2009: 154) que M. Prony, por la época de la Revolución Francesa, se le encargó la tarea de cambiar la base de los logaritmos utilizados en la balística militar. Para ello, e “inspirado” en el ejemplo de los alfileres de La Riqueza de las Naciones de Adam Smith, procedió aplicando las leyes de la división del trabajo. Buscó las ecuaciones que requería resolver y con la ayuda de aproximadamente 10 matemáticos las convirtió a operaciones fundamentales de suma y resta, y abrió talleres de 70-80 personas, de escolaridad mínima para realizar las múltiples operaciones básicas, en tanto que el grupo de los 10 especialistas solo se encargaba de verificar la validez de los cálculos contrastando resultados.

Esto, por un lado, es una evidencia para Babbage de que esas operaciones de suma y resta eran trabajo simple que podía ser sustituido por su “Motor Analítico”, y por otro lado, revela la importancia de la lógica simbólica de Boole en el desarrollo de este proceso, aspecto que es soslayado por autores como Lara (2012: 88-90), quien no menciona a dicho matemático en su historia de la computación, y se centra en Babbage

por considerar más importante sus aportes en relación con los códigos binarios que han permitido representar objetos más complejos.

Sin embargo la noción de objeto abstracto en matemáticas (Murillo, 2007: 29) y de máquina abstracta en computación tiene fundamento en la lógica simbólica y no en la binaria, aunque sea mediante la segunda que se realicen las operaciones automatizables de las tareas en que se dividen los procesos de trabajo a través de las unidades de procesamiento y almacenamiento de datos.

Es precisamente sobre esta diferencia en que yace la comprensión de que la materia prima en este proceso de trabajo son las unidades de memoria, mientras que los lenguajes de programación son el medio de trabajo a través de los cuales se transforma dicha materia prima.

Esto queda de manifiesto si se considera que no son los programadores informáticos quienes producen manualmente la máquina computadora, sino quienes se encargan de tomar su capacidad de procesamiento y almacenamiento de datos como materia prima- cuál reactivo limitante- que les permitirá, mediante algún lenguaje de programación, producir una cantidad finita de algoritmos y módulos en código informático en un tiempo determinado que quedarán registrados en esa memoria.

Esto es así, tanto si se utiliza un lenguaje compilado, y ese código fuente es convertido en código-objeto (como se explicará más adelante), como si esos archivos de código son almacenados, generalmente con grandes bases de datos, en memorias físicas de diversas capacidades (memoria USB, CD's, discos duros) o se mantengan en servidores en internet especializados para ello, como Github.

Por ello, el resultado de una jornada laboral de programación informática tendrá como resultado archivos de código fuente, con las líneas producidas y probadas para

generar los módulos en los que se divide todo sistema informático escritas en las unidades de almacenamiento, signaturas que antes no estaban ahí, es decir, no fueron hechas por el obrero fabril que ensambló la computadora, ni el que ensambló las memorias físicas de la máquina.

Como los nudos de los quipus incas, las tablas de Sumeria, los papiros egipcios, los códices mesoamericanos, o los petroglifos, los códigos informáticos son producidos en un medio físico (memoria de almacenamiento) a través de otro medio físico (memoria de procesamiento)

Por tanto, , es incorrecta la afirmación de Olivier Bondeau de que la producción “inmaterial”- como los sistemas informáticos- se caracterizan porque las materias primas y mercancías no son materia, ni siquiera substancia, ni energía, sino símbolos, códigos, signos lingüísticos y matemáticos, competencias y disposiciones (2004: 34). No son las materias primas, sino el medio de trabajo el que consiste en códigos simbólicos, y por tanto, el que podría ser investigado como signo semiótico, como sostiene Fumagalli (2010:188)

Más aún, como no se sostiene la idea de que las materias primas o el valor de uso informático mismo no sean materia, y como de una premisa falsa no se sigue algo verdadero, se debe rechazar la conclusión de que hay una tendencia a la desmaterialización de los medios de producción que dificulta establecer la frontera entre fuerzas productivas y los medios de producción (Blondeu, 2004: 35), ya que ese valor de uso informático si es material y existe bajo la forma de dispositivos

electromagnéticos, cintas perforadas, tubos de cilindro, entre otros²⁰, disponibles para ser signados con códigos informáticos.

De esta forma, la materia prima en el proceso de programación informática es la máquina computadora y los servidores (que son memorias en red), y contrario a la apariencia, los lenguajes de programación son el medio de trabajo, en tanto que la máquina abstracta-como la denomina Turing mismo a esas “ficciones matemáticas más que objetos físicos” (1950: 449)-, a través de los cuales el programador consume esa materia prima que constituye la memoria para producir nuevas líneas de código.

Esos nuevos registros en la memoria, una vez terminados, se constituirán en unidades mínimas de algoritmos denominadas “módulos”, los cuales pesan una determinada cantidad de bytes.

La máquina computadora del programador informático se consume productivamente con el impulso de los dedos del programador informático durante las horas de la jornada laboral para producir esos módulos, cuya integración ha sido denominada en esta exposición sistema informático o “valor de uso informático”, hasta que su memoria de procesamiento y almacenamiento de datos se convierte en un impedimento para la adecuada ejecución de las tareas en las que se divide este proceso de trabajo porque se ha vuelto muy lenta, relativamente, y por tanto se considera obsoleta.

3.2 El medio de trabajo

²⁰ La dificultad que enfrenta el autor, se puede especular, se origina en el efecto confuso del pseudo-concepto de “inmaterialidad”, pero no corresponde a esta exposición abordar asuntos metafísicos, ya que según Blondeau la mercancía se convierte en una abstracción metafísica (2004: 45)

Los lenguajes de programación le permiten al programador informático protocolos diferentes de comunicación con las máquinas computadoras. Se pueden distinguir dos tipos generales de lenguajes de programación informática (Downey, Eleker, y Meyers, 2002:1)

En primer lugar, se encuentran los lenguajes de bajo nivel, cercanos a la programación robótica y que se utilizan para configurar los componentes electromecánicos y cuánticos de las máquinas computadoras con los que se puede establecer comunicación desde una terminal de “Root”, es decir, desde la máquina misma, ya sea para reconfigurarla, o para instalarle algún sistema informático, como un sistema operativo que permita muchas otras funcionalidades

En segundo lugar, se encuentran los lenguajes de alto nivel, utilizados por los programadores informáticos para producir valores de uso como los sistemas operativos, que permiten una interfaz gráfica diferente a la de una terminal, y los utilizados por ellos mismos para producir sistemas informáticos, como por ejemplo los entornos de desarrollo integrados, o “IDE” por sus siglas en inglés, que permiten depurar el código, entre muchas funcionalidades para las tareas en las que se divide este proceso de trabajo.

Así, la máquina abstracta que funciona como medio de trabajo en la programación informática son precisamente estos lenguajes de alto nivel y se dividen a su vez en dos tipos (Downey, Eleker, y Meyers, 2002:2):

- i) Los lenguajes interpretados, como Python, que permiten al codificador comunicarse directamente con una terminal –canal de comunicación directo entre la máquina computadora y el programador

informático- mediante un lenguaje intermedio denominado intérprete que convierte el código fuente directamente a lenguaje máquina;

- ii) Los lenguajes compilados , como los C# o Java, que utilizan un compilador para convertir el código fuente producido en ese tipo de lenguaje en un tipo de archivo binario denominado “código-objeto”, el cual puede ser ejecutado pero no modificado y se le denomina “archivo ejecutable” o “.exe.

3.3 Los materiales auxiliares

El carácter del almacenamiento como materia prima, el de los lenguajes de alto nivel como medio de trabajo, y el de la electricidad como material auxiliar como según lo precisa el mismo Turing, queda claro si se considera el uso de servidores de desarrollo y de servidores de producción durante las etapas y tareas en las que se desarrolla dicho proceso de trabajo.

Conforme aumenta el tamaño de un sistema informático, aumentan los requerimientos técnicos, en términos de velocidad de procesamiento y capacidad de almacenamiento de datos, durante la ejecución de las tareas en las que se divide el desarrollo de sistemas informáticos, por lo que debe aumentarse la capacidad de esos servidores para poder procesar los códigos cada vez más complejos y los datos generados por el sistema.

Estos servidores deben mantenerse a temperaturas que no dañen los dispositivos de almacenamiento, como las cintas magnéticas, por lo que consumen grandes cantidades de electricidad, tanto para mantenerse en funcionamiento continuo como para sus sistemas de enfriamiento y ventilación.

Adicionalmente, el internet aparece como un material auxiliar para la mayor parte de los programadores informáticos, primero porque es el medio para comunicarse con los servidores donde alojan sus códigos y las bases de datos generados por los valores de uso informáticos, y en segundo lugar, porque en los “blogs” especializados en informática se encuentran muchas líneas de código y protocolos de comunicación entre distintos sistemas informáticos que consultan con mucha frecuencia cuando desde su propia computadora no logran dar con la solución requerida.

Sobre esto se volverá más adelante ya que entre las comunidades de programación libre y abierta es una práctica conocida como reciclaje de código, y en muchas ocasiones (como en el ejemplo que se expone más adelante) programadores contratados por corporaciones capitalistas solo ahí encuentran las soluciones que requieren en su jornada laboral.

3.4 El edificio

La noción de edificio como factor objetivo dentro del proceso de trabajo puede ser simplificado en este contexto a una mesa de escritorio y a una silla de oficina, objetos que pueden encontrarse en diversos espacios absolutos, como edificios con cubículos o dentro de la vivienda particular de cada programador informático.

Sin embargo, como espacio absoluto del proceso de trabajo, el edificio es el lugar donde se da la transformación material de las memorias de almacenamiento y procesamiento de datos, a través del mecanismo de impulso y respuesta del tecleo de manos de los programadores informáticos redactando el código fuente, proceso semejante al de los mineros picando y extrayendo las materias primas de la mina, al igual que la mayor parte de los otros procesos de trabajo.

En resumen, los medios de producción requeridos para producir sistemas informáticos son:

- a. Materia prima: la máquina computadora
- b. Medio de trabajo: los lenguajes de programación
- c. Materiales auxiliares: i) electricidad, ii) internet
- d. Edificio: oficina (silla, mesa)

Durante la jornada laboral, cada uno de los programadores informáticos dispone de una combinación finita, k , de estos medios de producción (a, b, c, d), los cuales son consumidos productivamente durante las horas de trabajo (HT) en las que ha sido contratado, lo cual puede ser formalizado estableciendo la combinación k como la combinación de elementos a, b, c y d que convergen en una cantidad z de unidades, o módulos, de valor de uso informático con una cantidad específica de horas de trabajo.

4. El factor subjetivo del proceso de producción de valores de uso informáticos

Se define la sucesión de tareas s en el orden temporal $(s, -s)$ como el conjunto de formas de organizar las tareas de concepción y ejecución en las que se divide el proceso de desarrollo de los sistemas informáticos requeridos para producir la cantidad z de unidades de valor de uso informático, es decir, z módulos de cualquier sistema informático en una cantidad finita de horas de trabajo (HT) con la combinación k de factores productivos a, b, c, d .

Si se acepta con Marx que la fuerza de trabajo ingresa como obrero parcial de un obrero colectivo (1984: 397), entonces el conjunto de todas las sucesiones de tareas s en las que se divide el proceso de trabajo en cuestión, realizados por obreros parciales como administrador de base de datos, codificador, probador o instalador, se puede denominar obrero colectivo $(s, -s)$ para referir a esas diversas formas de organizar un

mismo proceso de trabajo subdivido en tareas s que dan como resultado un sistema informático con z módulos.

Este obrero colectivo ($s, -s$) aporta la cantidad de horas de trabajo (HT) como parte del factor subjetivo de producción, y permite plantear el problema de la organización de las tareas en las que se divide la programación de los sistemas informáticos en la discusión contemporánea entre la programación corporativa y la programación libre, desde la perspectiva de la clase obrera en la lucha de clases (Lucaks, 1978: 45), o como dice el propio Marx al respecto de la duración de la jornada laboral: una lucha entre el capitalista colectivo, esto es, la clase de los capitalistas, y el obrero colectivo, o sea la clase obrera (1984: 282).

Las tareas s en las que se divide este proceso de trabajo son siete, dos pertenecientes a la concepción y cinco pertenecientes a la ejecución, cada una de las cuales se lleva a cabo, al decir de Charles Babbage (2009: 157), solo con la cantidad requerida de cada una, las cuales son medidas en horas de trabajo de un programador informático.

Además, existe una tarea s en los procesos bajo el poder de mando capitalista, conocida como el diseño del proceso de trabajo, tarea s que pertenece al “management científico” según la propuesta de Braverman (1998: 62) y en la cual se centra el análisis al respecto de la autonomización o no del trabajo en relación con el poder de mando del capital.

Lo anterior tiene que ver con las continuas discusiones al respecto del carácter ontológico de los programas informáticos. Para algunos autores, es un “artefacto” que proviene de una ingeniería (Riemer y Johnston, 2011: 15), mientras que para otros es

una mercancía (James y Lalonde, 2015: 22) y para otros una mezcla entre artesanía y bien público (Raymond, 2001: 100).

Estas diferencias no son baladí, ya que la organización del trabajo se diferencia si se incluye la tarea que tiene que ver con el “management científico”. Es decir, para quienes conciben los programas informáticos como mercancías, por ejemplo, tanto esa tarea de control como la de comercialización deben ser incluidas, pero en las teorías que enfatizan la producción del valor de uso, por encima de su comercialización, esas tareas no son consideradas, como se discutirá más adelante.

4.1 Las tareas de concepción

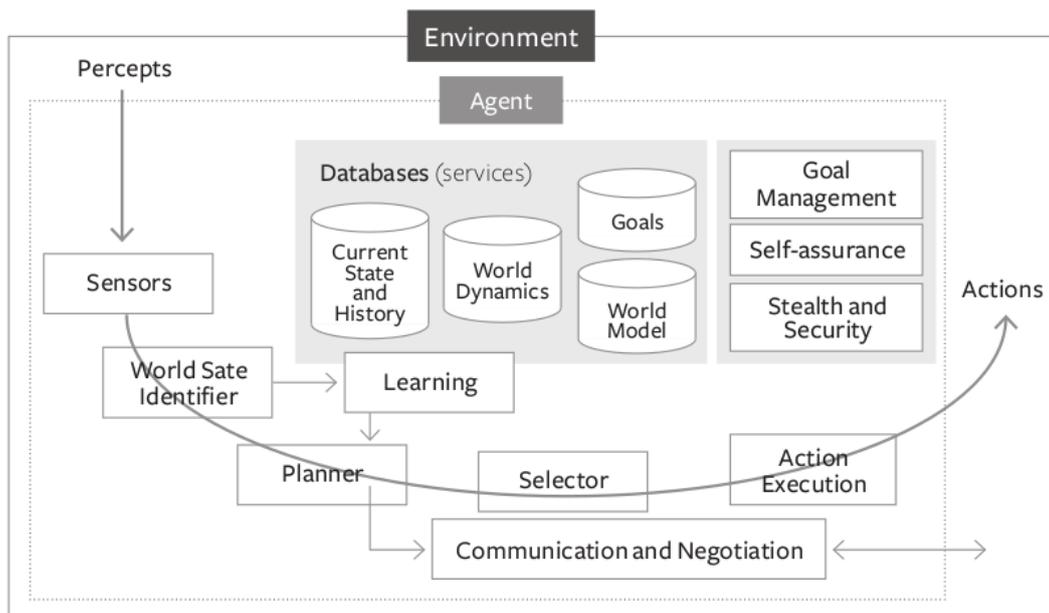
La primera de las tareas en las que se divide la concepción de programas sistemas informáticos es la arquitectura según la toma de requerimientos, es decir, el planteamiento del cliente o los usuarios al respecto de un problema que requiere ser ordenado en tareas más simples, de tal forma que pueda ser resuelto de manera automatizada y con un monitoreo de control en tiempo real.

Cuando estos valores de uso informáticos son producidos para ser comercializados como mercancías, será un comprador el agente económico que tomará la iniciativa de plantear un problema, mientras que será un especialista en programación informática, quien dividirá ese problema en tareas más simples y, mediante una sintaxis próxima a la programación robótica pero orientada a la informática, representará el mecanismo general de funcionamiento del sistema informático requerido para que el orden de ejecución de las tareas sea el deseado.

A modo de ejemplo, considérese la arquitectura propuesta por Kott (2018) para soldados mediante inteligencia artificial. Como puede verse en la Figura 1, una arquitectura informática es un tipo de diseño global con indicaciones estructurales que

reúne las tareas básicas que el programador informático deberá codificar durante el proceso de trabajo. En toda arquitectura informática es indispensable una base de datos y en este caso de un soldado automático, está combinado con una arquitectura de inteligencia artificial basada en un módulo de aprendizaje.

Figura 1. Arquitectura informática de un soldado robótico dotado con inteligencia artificial



Fuente: Kott(2018: 63)

Cuando el valor de uso informático es producido mediante prácticas de programación libre, muchos de los problemas son planteados por especialistas del campo y su solución requiere de la colaboración de muchos programadores. Tal es el caso de los orígenes mismos de GNU y Linux, y que en parte podría explicar la superioridad técnica de soluciones libres para los servidores, como es el caso de Apache.

Los defensores de la programación libre sostienen que la programación informática bajo el poder de mando capitalista –la programación corporativa- busca resolver problemas de usuarios que no entienden el funcionamiento de la informática y de las computadoras en general, lo que da lugar a “la paradoja de escuchar a los idiotas” (Lerner y Tirole, 2002: 206).

Por esta razón, dichas tecnologías privativas pueden resultar ineficientes en términos de arquitectura o enteramente aplicadas a la usabilidad de un tipo de usuarios que pueden denominarse como “analfabetas informáticos” porque no saben leer ni escribir códigos informáticos en ningún lenguaje, y desarrollan relaciones de dependencia con su proveedor de tecnologías digitales, porque tienen un nivel de comprensión nulo de los códigos sintácticos y semánticos en los que se basan esas tecnologías, y por tanto, solo pueden interactuar con algunos de sus códigos pragmáticos, los concebidos para un usuario funcional, tal y como los analfabetas funcionales estudiados por la teorías de la educación y el aprendizaje (Vygotsky, 1995: 132).

De esta forma, la primera de las tareas en la concepción de los sistemas informáticos combina el conocimiento de los lenguajes de programación, aunque no se haya puesto una sola línea de código, con una visión global de un proceso de trabajo que debe ser dividido en tareas más simples.

La segunda tarea en la que se divide el proceso de concepción de los sistemas informáticos es el diseño del sistema mismo y constituye un desarrollo de la arquitectura, pero con especificaciones técnicas semejantes al diseño de componentes electrónicos, solo que con objetos abstractos, es decir, programables mediante un lenguaje informático.

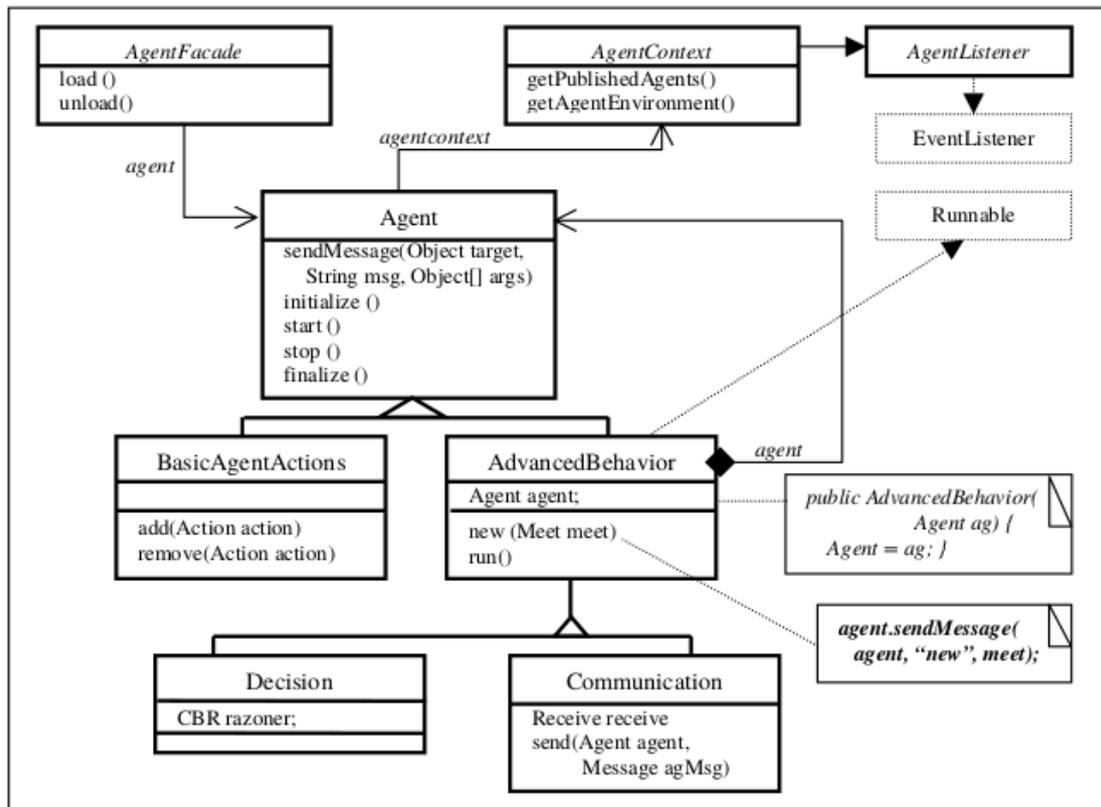
Los diseños de los sistemas informáticos son tan variados y diversos como programadores hay, pero generalmente se elaboran siguiendo patrones, es decir, relaciones frecuentes que establecen entre sí las tareas en las que se ha dividido un proceso de trabajo para resolver un problema.

Sin embargo, el proceso de diseño, al igual que el de la arquitectura, es complejo y supone cualidades, como la creatividad, para poder ser realizado de manera adecuada. Por ello existe una amplia variedad de diseños para un mismo tipo de valor de uso informático que dependerá de las cualidades subjetivas del programador encargado, pero también del uso que se le pretende dar en términos de cantidad de usuarios, módulos, usabilidad e incluso de vida útil.

Así, las características técnicas, como la modularidad misma del sistema informático o la granularidad en base de datos, dependen del diseño ya que es la que se encarga de especificar los dispositivos identificados en la arquitectura según módulos, interfaces, protocolos de comunicación, bases de datos, entre otros objetos abstractos disponibles mediante los lenguajes de programación informática para su consumo productivo durante las horas de trabajo que dura la jornada laboral.

Para comprender mejor esta tarea de diseño, diferenciándola de la arquitectura, considérese el siguiente caso referido a un sistema multiagente en un entorno de desarrollo de Java, un lenguaje de programación del tipo ejecutado, es decir, que el código fuente es convertido con un compilador en un código-objeto que no puede ser modificado sino solo ejecutado, como se muestra en la figura 2.

Figura 2. Diseño de una clase para un sistema multiagente de propósitos múltiples



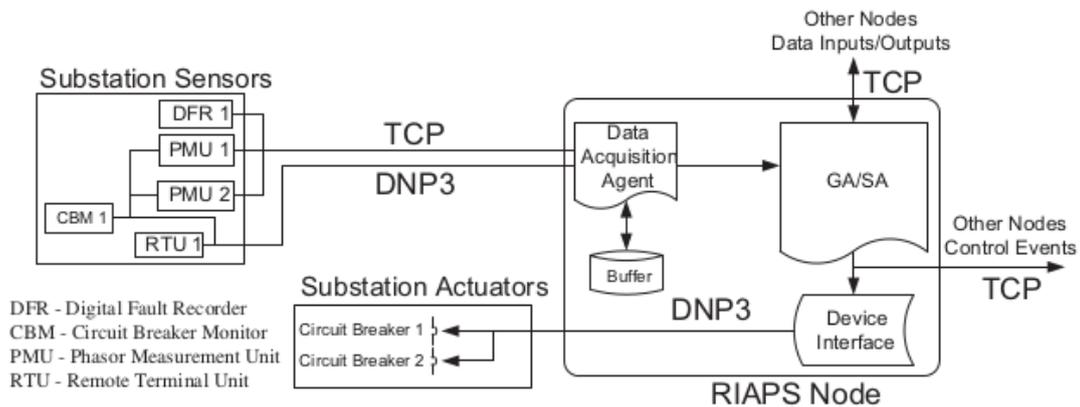
Fuente: Avancini y Amandi (2000: 5)

En la figura 2 se muestra una clase, objeto característico en este lenguaje, con el diseño de un agente para un sistema multiagente en Java. Como se puede ver, están especificadas acciones y relaciones concretas entre dichas acciones, las cuales deberán ser programadas y probadas por los obreros parciales encargados de estas tareas s como parte de un obrero colectivo ($s, -s$).

Esta noción de diseño también está presente en la programación robótica, es decir, aquellos que utilizan lenguajes robóticos, o de bajo nivel, para programar los sistemas incrustados en los dispositivos electrónicos, como impresoras, teléfonos móviles y las computadoras mismas. En este caso se diseñan las combinaciones de componentes electrónicos, entre otros objetos robóticos, requeridos para que la máquina

mecánica realice el proceso esperado. A modo de ejemplo de estos diseños para programación robótica considérese el presentado en la figura 3.

Figura 3. Diseño de un componente electrónico multiagente para programación robótica



Fuente: Jing y Chen (2017: 193)

En la figura 3 se muestra un diseño multiagente para programación robótica y es visible que sus elementos, en contraste con las funciones de la figura 2, son sensores y otros objetos robóticos, semejantes a los especificados en la figura 1 de una arquitectura informática de un soldado robótico dotado con inteligencia artificial.

En este sentido, debe enfatizarse que conforme avanza la concepción de un valor de uso informático, no solo aparecen los lenguajes de programación y su dominio especializado como medio de trabajo para el proceso de programación informática, sino que además aparecen los componentes electrónicos utilizados como dispositivos lógicos con funcionamiento mecánico, es decir, una combinación entre materias primas (máquina mecánica) y medios de trabajo (máquina abstracta) cuya combinación da como resultado una máquina dual, es decir, objetos como la computadora que reúnen tanto el

dispositivo electrónico (“hardware”) como el medio para hacer funcionar dicho dispositivo (un sistema operativo).

4.2 Las tareas de ejecución

La primera tarea s en el proceso de ejecución es la codificación de los módulos, clases, objetos o rutinas en los que se ha dividido el diseño de un sistema informático.

La codificación informática consiste en la elaboración de algoritmos que permiten resolver problemas, tal que unidos en módulos permiten resolver de manera automática tareas mecánicas.

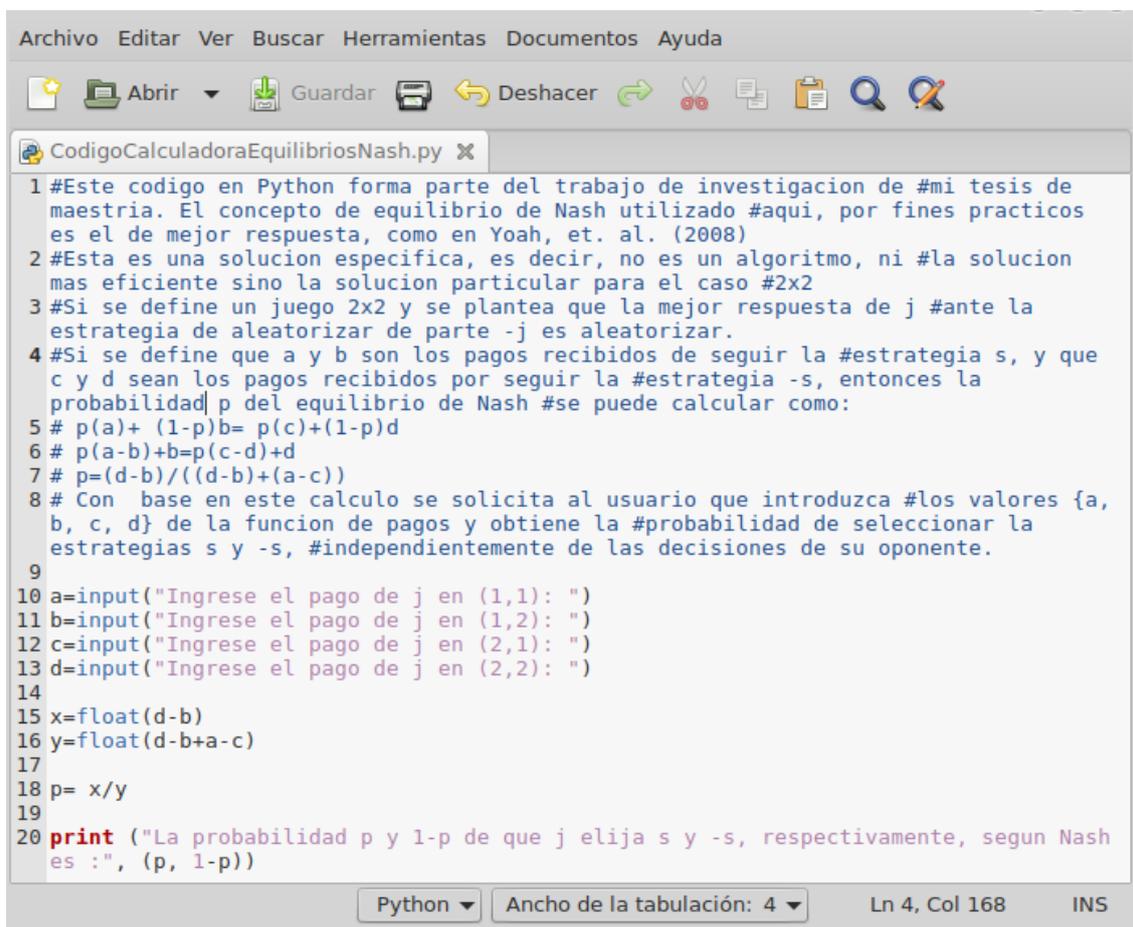
El carácter complejo de esta tarea s se pone de manifiesto a partir de que en su medio de trabajo se constituye el desarrollo de la representación formal de la lógica simbólica, mediante la lógica binaria, a través de la cual el ser humano ha establecido un protocolo de comunicación con las máquinas mecánicas.

Por consiguiente, la codificación requiere de un nivel de alfabetización en saberes como lógica, electrónica, protocolos de comunicación, sintaxis de lenguajes, así como una eficaz capacidad de abstracción para representar en un código específico una semántica interpretable por las máquinas mecánicas, que de como resultado las salidas planeadas en el diseño y que a la vez puedan ser integradas sin errores estructurales de acuerdo con la arquitectura del sistema informático como un todo.

A modo de ejemplo de esta tarea se presenta en la figura 4 un algoritmo, es decir, una serie de líneas de código en un lenguaje de programación que resuelve un problema específico mediante una solución general (Downey, Eleker,y Meyers, 2002: 146-147).

Al respecto, se sigue la ecuación propuesta por Zapata Lilo (2013: 10) para los juegos simétricos 2x2, para calcular los equilibrios de Nash y se utiliza el lenguaje Python como se muestra a continuación en un archivo tipo.py, es decir, que el “intérprete” sabe leer y escribir en el editor de textos Pluma:

Figura 4. Algoritmo en el lenguaje interpretado Python para calcular equilibrios Nash en matrices 2x2



```
Archivo  Editar  Ver  Buscar  Herramientas  Documentos  Ayuda
Abrir  Guardar  Deshacer
CodigoCalculadoraEquilibriosNash.py x
1 #Este codigo en Python forma parte del trabajo de investigacion de #mi tesis de
  maestria. El concepto de equilibrio de Nash utilizado #aqui, por fines practicos
  es el de mejor respuesta, como en Yoah, et. al. (2008)
2 #Esta es una solucion especifica, es decir, no es un algoritmo, ni #la solucion
  mas eficiente sino la solucion particular para el caso #2x2
3 #Si se define un juego 2x2 y se plantea que la mejor respuesta de j #ante la
  estrategia de aleatorizar de parte -j es aleatorizar.
4 #Si se define que a y b son los pagos recibidos de seguir la #estrategia s, y que
  c y d sean los pagos recibidos por seguir la #estrategia -s, entonces la
  probabilidad p del equilibrio de Nash #se puede calcular como:
5 # p(a)+ (1-p)b= p(c)+(1-p)d
6 # p(a-b)+b=p(c-d)+d
7 # p=(d-b)/((d-b)+(a-c))
8 # Con base en este calculo se solicita al usuario que introduzca #los valores {a,
  b, c, d} de la funcion de pagos y obtiene la #probabilidad de seleccionar la
  estrategias s y -s, #independientemente de las decisiones de su oponente.
9
10 a=input("Ingrese el pago de j en (1,1): ")
11 b=input("Ingrese el pago de j en (1,2): ")
12 c=input("Ingrese el pago de j en (2,1): ")
13 d=input("Ingrese el pago de j en (2,2): ")
14
15 x=float(d-b)
16 y=float(d-b+a-c)
17
18 p= x/y
19
20 print ("La probabilidad p y 1-p de que j elija s y -s, respectivamente, segun Nash
  es :", (p, 1-p))
Python  Ancho de la tabulación: 4  Ln 4, Col 168  INS
```

Fuente: Elaboración propia

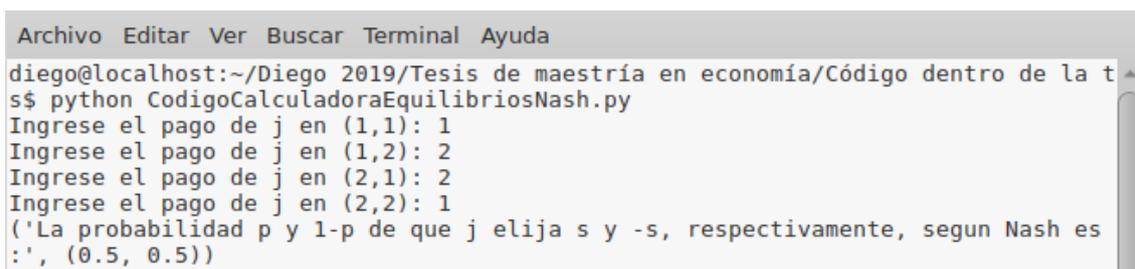
Como se muestra en la figura 4, un lenguaje de programación es un conjunto de palabras y símbolos reservados en un código que permite producir una máquina abstracta que imita a alguna máquina mecánica. En el caso del código aportado, las

primeras líneas aparecen en azul porque están anteceditas del numeral “#”, y por tanto el “intérprete” no las ingresa porque son comentarios.

Mediante las líneas de código de la figura 4 se logra que la máquina computadora imite una calculadora de equilibrios Nash, y para ello lo primero es definir los cuatros valores de la matriz 2x2 (a, b, c, d) como valores ingresados por un usuario mediante la palabra reservada “input” y luego se aplica la ecuación en números de tipo flotante, es decir, un tipo de decimal que la máquina mecánica sabe manipular.

Finalmente se utiliza la palabra reservada “print” a través de la cual se obtiene la impresión en la pantalla del resultado obtenido. En este caso, y por ser un lenguaje interpretado, este resultado se obtiene a través de una terminal, es decir, el canal directo de comunicación entre el ser humano y la máquina mecánica. Esta salida se muestra en la figura 5.

Figura 5. Salida obtenida mediante un algoritmo en Python que imita una calculadora de equilibrios Nash



```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
diego@localhost:~/Diego 2019/Tesis de maestría en economía/Código dentro de la t
s$ pythonCodigoCalculadoraEquilibriosNash.py
Ingrese el pago de j en (1,1): 1
Ingrese el pago de j en (1,2): 2
Ingrese el pago de j en (2,1): 2
Ingrese el pago de j en (2,2): 1
('La probabilidad p y 1-p de que j elija s y -s, respectivamente, segun Nash es
:', (0.5, 0.5))
```

Fuente: Elaboración propia.

En la figura 5 se muestran los valores ingresados y la salida en una terminal de la calculadora de equilibrios Nash programada a modo de ejemplo de algoritmo. Sin embargo, el proceso de desarrollo de sistemas informáticos es un proceso que involucra no solo esta tarea de codificación sino que el nivel de complejidad que involucra la

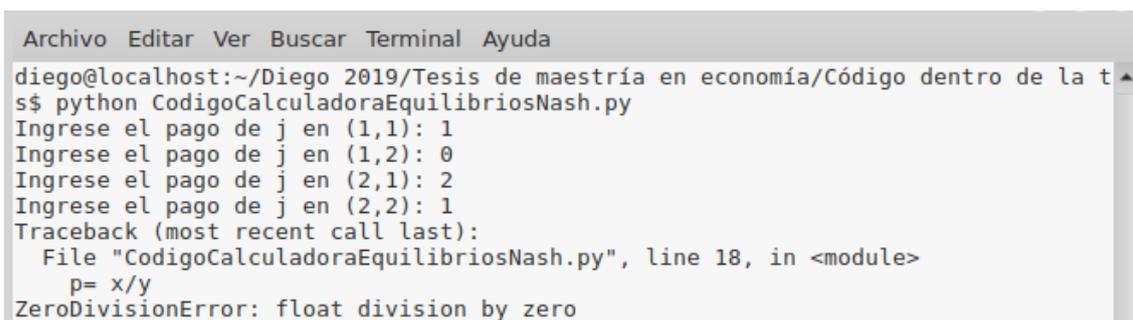
producción de un valor de uso informático queda de manifiesto en la organización misma de dicho proceso de trabajo, como se muestra a continuación.

La segunda tarea de ejecución son las pruebas, las cuales consisten en un conjunto de evaluaciones del código fuente, realizadas por lo general por un obrero parcial diferente al codificador, para determinar la estabilidad de los algoritmos, así como para verificar si cumple con los requerimientos funcionales especificados en el diseño y la arquitectura, esto es, que el sistema realice lo que se le pidió al codificador que hiciera. Es una tarea semejante a la primera en el sentido de que las pruebas de ese sistemas son programadas para ello, y buscan encontrar los errores que el otro obrero parcial no encontró o ni siquiera identificó.

A modo de ejemplo de la tarea s de realizar pruebas al código programado, considérese el resultado obtenido en la terminal y que se muestra en la figura 6.

Figura 6. Prueba del algoritmo desarrollado para calcular equilibrios Nash en matrices

2x2



```
Archivo Editar Ver Buscar Terminal Ayuda
diego@localhost:~/Diego 2019/Tesis de maestría en economía/Código dentro de la t
s$ python CodigoCalculadoraEquilibriosNash.py
Ingrese el pago de j en (1,1): 1
Ingrese el pago de j en (1,2): 0
Ingrese el pago de j en (2,1): 2
Ingrese el pago de j en (2,2): 1
Traceback (most recent call last):
  File "CodigoCalculadoraEquilibriosNash.py", line 18, in <module>
    p= x/y
ZeroDivisionError: float division by zero
```

Fuente: elaboración propia

En la figura anterior se muestra un error del algoritmo codificado para calcular equilibrios Nash y que consiste en que al calcular la probabilidad como una división, e ingresar un cero, se indetermina la función y por lo tanto no puede ofrecer resultado

alguno. Nótese que la terminal indica en este caso la línea en la que se encuentra el error, en este caso semántico, es decir, que la lógica implementada no puede resolver todos los casos de una matriz 2x2. También existen errores sintácticos, como utilizar una palabra reservada para definir una variable, o incluir una tilde si el lenguaje está en inglés, pero basta con este ejemplo para ilustrar en qué consiste esta tarea s.

La integración es la tercera tarea s de ejecución dentro del proceso de trabajo en el desarrollo de los sistemas informáticos. Consiste en codificar las interfaces y comunicaciones entre los módulos del sistema informático para que su funcionamiento global sea el esperado y pueda ser instalado en un entorno de producción, es decir, en un ambiente con usuarios.

Formalmente un sistema informático se puede definir como la integración de módulos cuya conjunción da como resultado las funcionalidades definidas en la arquitectura. Por ello un sistema de gestión de aprendizaje (LMS), como Moodle o LMS365, es la integración de un conjunto de módulos funcionales que permiten automatizar procesos de gestión educativa (contenidos, clases, evaluaciones, etc)

Un módulo puede estar codificado sin errores de funcionamiento interno, pero una vez integrado con los otros módulos ocasionar diversos tipos de problemas. Por ello en la integración se realizan pruebas que buscan asegurar que no existan resquicios, errores lógicos y procedimentales que permitan, por ejemplo, que usuarios no autorizados puedan acceder a la información o realizar cambios. A modo de ejemplo de un proceso de integración de un sistema en un entorno en el que se identifican errores, considérese la siguiente figura.

Figura 7. Correcciones de integración de un sistema multiagente en un entorno con usuarios

<pre> <bean name="Node" parent=".." singleton="false"> <property name="agents"> <list> <ref bean="Agent1" /> <ref bean="Agent2" /> </list> </property> </bean> <bean name="Agent1" parent=".." /> <bean name="Agent2" parent=".."> <property name="agentBeans"> <list> <ref bean="Bean2" /> </list> </property> </bean> </pre>	<pre> <bean name="Node" parent=".."> <!-- AWE3: value for "singleton" = "false" ↔ ↔ deleted --> <property name="agents"> <list> <ref bean="Agent1" /> <!--commented out <ref bean="Agent2" /> commented out--> <!--added--> <ref bean="Agent3" /> </list> </property> </bean> <bean name="Agent1" parent=".." /> <!--commented out <bean name="Agent2" parent=".."> <property name="agentBeans"> <list> <ref bean="Bean2" /> </list> </property> </bean> commented out--> <!--added--> <bean name="Agent3" parent=".."> <property name="agentBeans"> <list> <ref bean="Bean3" /> </list> </property> </bean> </pre>
--	--

Fuente: (Kuster et. al, 2014: 924)

Como puede verse en la figura 7, en la columna derecha se han realizado algunos cambios, en este caso referidos a un breve fragmento de un proceso de integración de un sistema informático. Las pruebas que se realizan durante el proceso de integración, y del sistema en su conjunto, se conocen como análisis de calidad (QA por sus siglas en inglés)

Debe hacerse notar que estas tareas de ejecución, (codificación, pruebas e integración) están vinculadas entre sí y requieren de coordinación y distribución de tareas de forma adecuada y por ello surge el obrero-capataz que diseña el proceso de trabajo para coordinar y controlar los cambios que se le realizan al código fuente. Antes

de atender este relevante asunto, es necesario mencionar las otras dos tareas de ejecución involucradas con el desarrollo de los sistemas informáticos, que son la instalación y el mantenimiento.

Una vez que un sistema informático ha sido probado e integrado, debe ser instalado en entornos para usuarios, como computadoras de escritorios o servidores web. Esta tarea, incluida dentro del conjunto de las tareas que se requieren para producir un módulo de valor de uso informático, se considera más simple que las anteriores e, incluso, se imparte en muchos centros de formación como conocimiento técnico.

También el mantenimiento es una tarea que se realiza una vez que el código fuente de un sistema informático ha sido integrado, probado e instalado, pero por diversas razones presenta problemas para los usuarios, los cuales tienen competencias diferentes en relación con las tecnologías informáticas. Aquellos usuarios que solo hagan un uso funcional del sistema en cuestión acudirán al técnico en mantenimiento, quien se encargará de solucionar las dificultades presentadas. Si el usuario tiene un nivel de competencia mayor, podría identificar el origen del problema que se le presenta e, incluso, solucionarlo por sí mismo o contactar directamente a los desarrolladores y hacerlo de su conocimiento.

Así, tanto la instalación como el mantenimiento son tareas indispensables para el valor de uso informático, más aún cuando la arquitectura misma considera al usuario analfabeta y dependiente de su proveedor para cualquier asunto relacionado con el sistema informático.

Hay que destacar que estas dos tareas, instalación y mantenimiento, de los sistemas informáticos, requieren de la laboriosa y rutinaria labor de encontrar los

procedimientos informáticos, en un sentido amplio, requeridos para que un determinado sistema pueda ser instalado y funcionar adecuadamente durante un periodo finito, o para resolver contingencias en su funcionamiento, o en el entorno del usuario con problemas, tal que el sistema pueda volver a su funcionamiento esperado.

Cuando estos problemas se originan sistemáticamente, y muchos usuarios presentan el mismo problema, se puede deber a errores de codificación o de diseño, algunas otras a errores de integración, y cuando son de instalación, los errores pueden estar relacionados con el entorno y la compatibilidad de sistemas informáticos.

Un ejemplo muy común, entre los problemas de instalación y mantenimiento se encuentra en la compatibilidad entre tecnologías (Apple, Microsoft, Linux), ya sea por sus lenguajes, por los paradigmas en los que son producidos (libre, corporativo), o por sus licencias.

Tal es el caso de Quiñones Orjuela (2011), un usuario de LMS Moodle, desarrollado mediante la programación abierta (como se verá en el siguiente apartado), que en el foro de dicha plataforma anuncia que esta “integrando Moodle en la nube (Azure -de Windows-) con un paquete que creó codeplex para subirlo a la nube” y aproximadamente una hora y treinta minutos después, comenta que lo logró y que si alguien necesita ayuda, que le pregunten. Entonces otro usuario, de nombre David Hernández, le replica: “¿Por qué no narras tu experiencia por aquí? Nunca está de más dejar un registro de las soluciones encontradas individual o colectivamente. Después de todo, para eso somos una comunidad ¿no?”. (Ver Anexo 1)

A partir de esta intervención, el primer internauta cuenta su procedimiento y da lugar a que otros usuarios comenten y pregunten, por lo menos hasta ocho meses

después de la publicación original, cuando el blog es cerrado a más comentarios porque el problema ha sido técnicamente resuelto.

Este ejemplo, además de contextualizar las tareas de actualización y mantenimiento de los sistemas informáticos, evidencia la existencia de las denominadas “comunidades de programación libre y abierta”, (FOSS, por sus siglas en inglés), un tipo de obrero colectivo ($s, -s$), que se basan en un conjunto de prácticas colaborativas, como el usuario en dicho ejemplo, y que constituyen un paradigma alternativo de organización del proceso de trabajo, esto es, de la división en las tareas s en el proceso de desarrollo de sistemas informáticos.

Este tipo de obrero colectivo ($s, -s$) constituye el principal argumento de los teóricos del capitalismo cognitivo en relación con lo que ellos denominan “intelecto general” como fundamento de la autonomización del trabajo ante el poder de mando del capital.

5. Las comunidades de programación libre y abierta como obrero colectivo ($s, -s$)

El hallazgo de la programación libre y abierta en el proceso de producción de sistemas informáticos constituye una singularidad en relación con otros procesos de trabajo. Sin embargo, de esto no se sigue que este tipo de obrero colectivo ($s, -s$) represente una fuerza productiva difusa intermedia entre los factores objetivos y los subjetivos sino que debe ser examinado con mayor detalle.

La programación libre y abierta, como paradigma de producción de sistemas informáticos, constituye un anti-paradigma en relación con los patrones y cánones que se siguen en la programación corporativa bajo el mando capitalista (Liviu, 2014: 50), y

aporta argumentos para considerar que el resultado de sus horas de trabajo sean valores de uso pero no necesariamente mercancías.

Las diferencias entre la programación corporativa, bajo el mando capitalista, y la programación libre y abierta son muchas. Ello permite discutir la relación entre el poder de mando capitalista sobre el trabajo y la ejecución de las tareas s durante la jornada laboral por parte de los programadores informáticos, la fuerza de trabajo y factor subjetivo, en este proceso de producción.

El proceso de producción, bajo una comprensión mecanicista y termodinámica, es concebido como la cantidad de trabajo, en un sentido físico, que deben ejercer los humanos dentro del proceso de transformación material de los medios de producción, (edificios, maquinaria, equipos, material, materias primas, materiales auxiliares), durante una cantidad determinada de horas hasta obtener una cantidad exacta de valores de uso, minimizando el uso de los medios de producción, pero principalmente el tiempo de trabajo.

Como precisa Cuesta Santos (2008: 3) el grado de desgaste físico y mental al que se ven sometidos los obreros parciales durante la jornada laboral para ejecución de sus tareas s en cada hora laboral se conoce como intensidad del trabajo, la cual se puede medir a través el radio r que se forma al expresar las tareas s como ciclo o radianes (Ver Anexo 2), es decir como si las tareas del proceso de trabajo se representaran como bolas en un espacio topológico en conjunción con el consumo de la canasta k de medios de producción.

Para cada rama de la economía capitalista existe una intensidad media, la cual es la que se exige a cada obrero parcial como parte del cumplimiento de su tarea en el proceso global. Así, este el radio r permite medir la intensidad del proceso de trabajo como la cantidad de tareas s realizadas en una hora de trabajo (HT) en un sentido

puramente físico, como en el trabajo impulsado por tracción animal, como el desgaste “del cerebro, músculo, nervio, mano, etc., humanos”.

En este sentido, el trabajo de la araña, de la abeja, del tejedor y del albañil son semejantes porque ambos son fuerzas transformadoras de la materia, y su acción consiste en esforzar los órganos que trabajan. Un gasto puramente fisiológico, orgánico, por parte de animales y humanos. Sin embargo, el trabajo humano es más que solo ese gasto orgánico del cuerpo humano, es actividad cerebral que no se ha verificado en otros animales²¹.

Marx propone (1984: 216) la subordinación de la voluntad que se manifiesta en atención como la característica específicamente humana dentro de la termodinámica de toda transformación de la naturaleza que es llevada a cabo durante el proceso de trabajo.

De esta forma, el trabajo humano es el único que además de implicar la transformación de la materia (física), y la producción de objetos complejos (telas de araña, panales), requiere del uso voluntario de las facultades cerebrales para construir un esquema mental²² del proceso y resultado de ejecutar las actividades (tareas) en las que consiste un trabajo humano mediante el uso de una combinación de medios de producción *k*.

Sobre esta característica es que asienta Braverman (1998: 94) la diferencia entre trabajo artesano (autónomo) y trabajo bajo el mando capitalista: al separar los procesos cerebrales de concepción de las tareas manuales de ejecución automatizada (y excluyendo la comunicación no verbal del cuerpo humano, como en las actividades artísticas), la clase capitalista impone su control en todos los ámbitos del proceso de

²¹ Braverman(1998: 32) sostiene que los productos de otros animales son instintivos, aunque quizá ese trabajo no ha sido subsumido al capital por la tozuda costumbre de los animales de no obedecer a sus depredadores.

²² El problema del signo interno Voloshinov (2009: 66) fue ampliamente discutido durante el siglo XX por diversas corrientes semióticas y semiológicas.

trabajo, vaciándolo de pensamiento y tornándolo en un puro movimiento de engranajes y palancas.

Pero el trabajo humano no solo es una transformación de la materia sino que da lugar a un producto con sentido humano (absoluto, relativo y relacional), en otras palabras, es producción cultural.

Las cualidades cerebrales de imaginar²³, crear, representar, así como la de concentrar la atención a voluntad, son necesarias pero en igual medida que las del cuerpo humano de sentir, percibir y subordinar sus movimientos con destreza y precisión para alcanzar objetivos puramente figurados en su cerebro, o de someterse a los de alguien más y seguir órdenes e instrucciones, propio de la comunicación no verbal como plantea Flora Davis (2016: 67).

Así, la mecánica de impulso y frecuencia que caracteriza la fuerza aplicada en las tareas durante la jornada laboral a través de medios de producción, es el punto de partida del concepto de trabajo simple, como aquella forma en la que la concepción y la ejecución están separadas, y por tanto cada uno de los trabajadores es un obrero parcial que realiza un segmento con una destreza que ha desarrollado solo para realizar esa tarea, como “gasto productivo del cerebro, músculo, nervio, mano, etc., humano” (Marx, 1984: 54).

Más aún, si esa tarea desaparece en una nueva organización del proceso de trabajo, ese obrero parcial se habrá vuelto obsoleto y su fuerza de trabajo habrá perdido su valor, es decir, padecerá histéresis²⁴.

²³ Pearl y Mackenzie (2018: 39) ponen en este nivel de importancia a la imaginación dentro de la revolución cognitiva relacionada con el desarrollo de la consciencia humana. Dice Adam Smith (2016: 61): “Por medio de la imaginación, nos ponemos en el lugar del otro, concebimos estar sufriendo los mismos tormentos, entramos, como quien dice, en su cuerpo”. Engels (1975: 78) enfatiza en el uso de herramientas dentro de la transformación del mono en ser humano.

²⁴ Escamilla (2014: 5) sostiene que hay dos mecanismos por los que sucede la histéresis: i) poder de negociación: que los trabajadores estén sindicalizado y o altamente calificados, (privilegio al momento de una situación adversa de la economía), y no están dispuestos a bajar sus salarios y por eso aumenta el

Por ello las diversas formas de organización del proceso de trabajo determinan la pertenencia o no de un obrero parcial dentro de un obrero colectivo, y las metodologías para el desarrollo de los sistemas informáticos constituyen los registros más fiables al respecto de la evolución histórica de las diversas formas de organización del proceso de trabajo informático. En su desarrollo se incluyen o excluyen obreros parciales dependiendo del estado histórico de la actividad laboral misma.

Las condiciones laborales generadas por cada una de las metodologías implementadas en la trayectoria histórica de la programación informática como oficio constituyen el fundamento del espacio relacional del proceso de trabajo, indisociable del espacio absoluto basado en la materialidad de los factores objetivos²⁵, y hace de la organización del proceso de trabajo algo más que el fenómeno puramente físico, en una dinámica de voluntades sujeta a las relaciones de poder en la escala macroscópica de la cotidianeidad y la rutina.

Precisamente es el espacio relacional del proceso de trabajo, durante la jornada laboral, en el que los cuerpos individuales de los trabajadores se convierten en un obrero colectivo (Marx, 1984: 425) en la medida que la interconexión del mecanismo global del proceso obliga a los obreros parciales, en su posición unilateral, “a funcionar con la regularidad inherente a la pieza de una máquina”. El resultado, sostiene Marx (1984: 589), es la transformación del obrero “en accesorio autoconsciente de una máquina parcial”.

Por consiguiente, se puede sostener que existe un conjunto de obreros colectivos $(s, -s)_i$ que contiene todas las prácticas colaborativas en las que se basa la

desempleo, ii) patrón de dependencia: a medida que los trabajadores permanecen desempleados, pierden habilidades.

²⁵ Harvey (2006: 143) plantea que la espacialidad en la teoría marxista se compone de un espacio absoluto, relacionado con lo material sensitivo como los valores de uso, un espacio relativo, vinculado con los intercambios comerciales, y un espacio relacional, fundamentado en los movimientos políticos y la expresión artística.

programación informática libre y abierta, y un conjunto de obreros colectivos $(s, -s)_j$ que contiene todas las metodologías corporativas en las que se basa la programación informática bajo el poder de mando capitalista o corporativa.

El carácter inherentemente colectivo de la programación informática queda de manifiesto en que siempre involucra a grupos de programadores que varían de los 2 a los 10, para sistemas pequeños y medianos, y equipos de más de 100 para sistemas grandes y muy grandes. Por ejemplo, Maruping, Venkatesh y Agarwal (2009: 378) analizan 110 obreros colectivos $(s, -s)_j$ con un promedio de 7.68 programadores cada uno.

En este sentido, la noción de Marx de obrero colectivo se muestra aquí con toda su capacidad descriptiva y analítica, como queda evidenciado con las “metodologías” de desarrollo de los sistemas informáticos, los cuales constituyen el hilo narrativo de la “historia de la informática” para los defensores de una ontología de la informática como una ingeniería, según McConnell (1998: 9)

6. Las metodologías para el desarrollo de los sistemas informáticos: autonomía y control

Braverman sostiene que la organización específicamente capitalista del proceso de trabajo se caracteriza por la separación sistemática entre las tareas de concepción, mal llamadas “trabajo mental”, y las tareas de ejecución, o imprecisamente denominadas “trabajo manual”, como mecanismo implementado por la clase capitalista como parte de su estrategia en la lucha de clases.

Esta separación es concebida por dicho autor como una degradación histórica del proceso de trabajo cuyo comportamiento es señalado por la ley general de la acumulación capitalista de Marx.

En este sentido, si los sistemas informáticos, o control numérico como lo denomina Braverman (1998: 135), constituyen la forma más desarrollada de la aplicación del conocimiento científico al proceso de producción para alcanzar los fines de la clase capitalista, entonces el proceso de trabajo mismo en el desarrollo de los sistemas informáticos debe manifestar dicho carácter, constituyendo el modelo a seguir por los demás procesos de trabajo.

Las diversas formas de organización de las siete tareas s anteriormente especificadas en el caso de los obreros colectivos $(s, -s)_j$, bajo el poder de mando capitalista, dan como resultado un conjunto de mercancías cognitivas, o capital mercantil, cuya circulación retornará al menos la ganancia media entre los capitales de su rama.

Estas formas de organización del proceso de trabajo se han desarrollado entre especialistas y analistas bajo la forma de las metodologías de desarrollo de sistemas informáticos. De hecho, la primera de metodologías, la de cascada, apareció a inicios de los 70 con el objetivo de sistematizar y formalizar las prácticas y patrones que hasta entonces se venían siguiendo en los laboratorios de tecnología en Universidades, en centros de investigación militar y en empresas del sector (Zmud, 1980: 49).

Liviu (2014: 41-51) describe comparativamente las 20 metodologías más populares (cascada, prototipos, iterativa e incremental, espiral, extrema, scrum, cleanroom, lean, dirigidas por pruebas, métodos cristal, entre otras) y destaca que la

programación libre en código abierto es una metodología descentralizada sin autoridad central, sin compensación para el equipo, sin contabilidad y con una alta tasa de éxito.

Según su descripción, miles de programadores trabajan en escribir, revisar y probar sistemas sin esperar ninguna compensación directa, la mayoría no se conoce cara a cara pero encuentran la manera de colaborar en armonía. Este tipo de programación tiene la ventaja de que muchos desarrolladores revisan el código y además aprovechan el ciclo horario al estar geográficamente distribuidos en el planeta.

En contraste, sostiene que las corporaciones cuentan con su propio paradigma metodológico, como el de Microsoft el cual está enfocado delibera y disciplinadamente para proyectos de tecnología donde la retroalimentación se recibe después del depuramiento (“deployment”) (Liviú, 2014: 50-51).

Por eso la comparación que hace Blondeau de los dos métodos generales de producción de los sistemas informáticos, siguiendo la tipificación de Raymond (2001: 21) con las categorías “bazar” y “catedral”, de los dos tipos generales de obrero colectivo ($s, -s$), en la rama de la computación, no es incorrecta: identifica, adecuadamente para el desarrollo de su época, el método corporativo con el desarrollo ágil (en su versión prototípica) y la opone a los modelos de gestión más comunes bajo las licencias tipo BSD y Apache, identificándolo con el modelo colaborativo.

Más aún, Blondeau acierta al destacar algunas de las ventajas técnicas de la programación libre y abierta (como los costos de eliminación de errores (“bugs”) del primer método y la separación de usuario, desarrollador e ingeniero que lo caracteriza), pero exagerando el alcance de los mismos hasta concebir dichas comunidades como “un espacio público de cooperación” (2004: 43)

Por ello es que para este autor la programación libre, particularmente Linux, se inscribe en una lógica del reembolso de los gastos ocasionados por la creación y distribución, incluyendo el tiempo de trabajo, en vez de una lógica de la ganancia (Blondeau, 2004: 40)

Según esta interpretación, esa ruptura con la lógica de la tasa de ganancia capitalista tiene el efecto sobre el proceso de trabajo de autonomizar al trabajador del poder de mando del capitalista propietario de los medios de producción, porque a partir de las licencias de Fundación GNU de los años 80, los códigos fuente de los lenguajes de programación son propiedad comunitaria (pública no privatizable) en diversos tipos de esas licencias.

Sin embargo, deben señalarse dos omisiones que cuestionan esta interpretación cognitivista del capitalismo contemporáneo: el control y los derechos de propiedad sobre los factores objetivos del proceso de producción.

En la primera omisión, los métodos de control implementados dentro del proceso de programación, es una variable que debe reconsiderarse a partir de la definición, antes destacada, de Turing de la máquina computadora al incluir el control como uno de sus partes constitutivas.

Así, el control debe considerarse desde el punto de vista de obrero colectivo $(s, -s)$ y desde dentro del obrero colectivo $(s, -s)_i$, aparentemente autonomizado del poder de mando del capital.

Por un lado, está el control numérico al que se refiere Turing, en relación con el funcionamiento mecánico del procesamiento y almacenamiento de los datos, pero por otro lado, está el control que es necesario ejercer sobre los obreros parciales que componen uno colectivo. La relevancia del control en el análisis es destacada por

Braverman (1998: 60) mismo, para quien fue ahí donde Taylor hizo su mayor aporte al “management científico”.

Maruping, Venkatesh y Agarwal (2009: 379) elaboran una exhaustiva documentación al respecto de la noción de control dentro de las prácticas corporativas de organización de este proceso de trabajo, y especifican dos modos generales de control.

El primero es el control formal, el cual consiste en el monitoreo y evaluación de acuerdo con los estándares en los que se subdivide: controles de producto y controles de comportamiento. El primero pone atención en los productos sin considerar el proceso para lograrlo, mientras que el segundo enfatiza en los procedimientos que deben seguir los miembros del equipo con el objetivo de alcanzar las metas del proyecto. Este tipo de control se asimila al control directo en otros autores (Gordon, Edwards y Reich, 1986: 18-19)

El otro modo de control es el denominado informal, el cual tiene que ver con las estrategias personales o sociales para regular los objetivos del empleado. La función del monitoreo es delegada al empleado en vez de al capataz (“manager”), pero se pueden utilizar incentivos sobre la conducta de auto monitoreo.

Las dos formas en que se divide el control informal son: el control de clan y el auto control. El primero consiste en una forma de socialización de los miembros de un equipo dentro de un conjunto específico de normas y valores que son evaluados por la organización. Mediante rituales y experiencias compartidas, los comportamientos aceptables son socialmente reforzados, de tal forma que los objetivos y el comportamiento son socialmente regulados colectivamente: el capataz (“management”) debe asegurarse del alineamiento de estas formas sociales y autoreguladas.

La segunda forma es el auto control y se relaciona con la autonomía individual para determinar las actividades necesarias y cómo realizarlas. En este caso, se estimula que los individuos organicen sus propios objetivos y luego auto regulen y auto monitoreen sus progresos para alcanzar sus metas. Por eso el uso de incentivos adecuados a este control son aquellos trabajos que demandan creatividad y actividad intelectual (Maruping, Venkatesh y Agarwal 2009: 379-380)

Por ello, esa tarea s que debe realiza el capataz (“manager”) de un proyecto es lidiar con diversas combinaciones de estas formas de control a lo largo del ciclo de vida un proyecto de programación informática.

De esta manera, la autonomía aparece en la literatura como un factor crítico dentro de los equipos de desarrollo ágil pero referida al comportamiento individual, para introducir flexibilidad al proceso para permitir al equipo atender un ambiente cambiante e impredecible (la falta de compromiso de uno o más obreros parciales, errores de diseño y dependencias entre tareas s de los obreros parciales) de las técnicas de la programación ágil (2009: 380)

Así, la noción de autonomía aparece relacionada con la forma de autocontrol al servicio de los intereses de la clase capitalista durante la jornada laboral de los obreros colectivos $(s, -s)_j$, lo que es coherente con la conjetura de esta investigación y constituye un mecanismo para eliminar la tarea s de control y vigilancia que se añade dentro de la organización específicamente capitalista²⁶.

Sin embargo, en la programación libre aparecen otras formas de gestión de la distribución de las tareas de ejecución. Entre estas figuras, la de los líderes carismáticos fue una de las que Lerner y Tirole (2002: 204) destacaron y sobre las que uno de los

²⁶ Este fenómeno de control sin vigilancia es descrito por Freud como “introyección de la autoridad” (1992: 119).

protagonistas, el CEO de Apache, respondió que es por “meritocracia” (Fielding, Hann; Roberts y Slughter, sf: 3)

De hecho, Ross (2013: 214) le reconoce a este paradigma de programación libre y abierta que la organización de sus proyectos busca facilitar la coordinación de esfuerzos dispersos geográfica y culturalmente mediante dos modalidades: i) administradores formalizados o electos (Mozilla, Apache); ii) estructura quasi-formal de comprometidos (implicados) que administran los insumos de la comunidad de colaboradores, dando forma y dirección al proyecto (Linux Kernel).

En esta segunda estructura destaca que la autoridad corresponde con la reputación, la cual se obtiene resolviendo retos y aspectos interesantes del proyecto, mostrando compromiso y competencia técnica, pero sostiene que si bien esta estructura ha sido señalada como más eficiente y productiva que aquella alcanzable en un ambiente corporativo, la ausencia de estructura es un inhibidor de ambas medidas en tanto implica una paralizante ausencia de coordinación.

Más aún, Ross (2013: 215) sostiene que esto ya lo había notado Braverman a propósito de que todo proceso de trabajo requiere algún nivel de habituación al mismo, y considera que la noción de trabajo auto-organizado casi siempre se acompaña de la concepción errada de que mayores niveles de democracia post industrial implica la minimización radical del trabajo alienado.

La segunda omisión que debe señalarse está relacionada con los derechos de propiedad sobre los factores productivos que caracterizan a los obreros colectivos $(s, -s)_i$ y que están fundamentados en las licencias de propiedad intelectual sobre los códigos informáticos. De hecho, esta característica es una singularidad de estas comunidades pero es impreciso considerar que todas tienen las mismas características

“copyleft”, como busca enfatizar Kyrou (2004: 85) para la producción artística, ya que como destaca el mismo Moulrier Boutang (2004: 115), este tipo radical de licenciamiento es propio de los sistemas informáticos que incorporan las cuatro libertades de la programación libre de Stallman y la Fundación GNU (2004: 45): de uso, de estudio, de distribución y de mejora.

De hecho, como encuentra Borja (2016: 56), para el año 2014 estas licencias GNU representaban solo el 40.54% de todas las inscritas, mientras que otras licencias compatibles con modelos de negocio capitalistas, como la MIT, la Apache, la BSD, la Pearl y la Microsoft Public Licence, ya sumaban el 51.78% del total.

Por ello, la autonomía potencial que se deriva de esta forma de propiedad debe ser matizada señalando que esta división interna del obrero colectivo $(s, -s)_i$, se traduce en comunidades de programación libre, abiertamente anticapitalistas porque no producen mercancías, y las comunidades de programación abierta, cuya particularidad es abrir su código a cualquiera que quiera colaborar, pero que venden bienes complementarios y contratan trabajadores asalariados para producirlos.

Así, la noción de libertad del paradigma de la programación libre, con Richard Stallman a la cabeza, es próxima a las reivindicaciones anarquistas y es abiertamente antagonista con el poder de mando capitalista sobre el trabajo y sobre la máquina. Esto constituye un paradigma en sí mismo que permite al programador informático servirse del trabajo de los otros a cambio de ceder el suyo a los demás. En este caso, el voluntarismo y otras cualidades que se precisan más adelante aplican con toda propiedad a este segmento del obrero colectivo $(s, -s)_i$.

Sin embargo, muchas críticas al respecto de las tesis de los teóricos del capitalismo cognitivo convergen en este punto. Por ejemplo Ross (2013: 203) centra su

argumentación en que constituye una formación gratuita de fuerza de trabajo porque su proceso de trabajo basado en el trabajo autodirigido de voluntarios mantiene un gran atractivo para el capital por ese voluntarismo de raíces lúdicas, entusiastas, académicas y de círculos de investigación.

La noción de programación en código abierto está inspirada en las prácticas de la programación libre, pero enfocada en compartir el código para realizar desarrollo colaborativo, esto es, emprender proyectos coordinados entre grupos de programadores con intereses o problemas a resolver afines, pero también mediante externalización, (“outsourcing”) o directamente en sistemas informáticos a la medida mediante programas en código abierto y obreros parciales realizando las tareas de acuerdo con las características de este tipo de sistemas, por lo que este segmento del obrero colectivo $(s, -s)_i$ ha motivado inversiones en estas tecnologías por parte de empresas capitalistas que desean reducir sus costos nominales.

Así, a pesar de que estas formas de propiedad expresan la singularidad de las comunidades de programación libre y abierta, como obreros colectivos $(s, -s)_i$, la misma se limita al medio de trabajo, y no incluye al resto de los factores objetivos del proceso productivo, pero además supone divisiones internas e términos de derechos de propiedad que constituyen una limitación considerable a la capacidad de esta autonomía para hegemonizarse y constituirse en una fuerza abiertamente antagonista con el modo de producción capitalista.

Por consiguiente, es inexacto sostener que el auto control que permite la autonomía individual dentro de los obreros colectivos $(s, -s)$, y las formas de propiedad sobre el medio de trabajo que ha favorecido la autonomía de los obreros colectivos $(s, -s)_i$ impliquen una ruptura con el control y vigilancia de un capataz,

sino que se han ido constituyendo en otra forma de subordinación a la voluntad de uno o varios miembros de la clase capitalista, una obediencia sin vigilancia para cumplir con las instrucciones y exigencias de intensidad durante las horas de trabajo (HT) de la jornada laboral en la producción de módulos y sistemas informáticos en código abierto, ya sea desarrollados a la medida o bajo otros modelos de negocio específicamente capitalistas.

6.1 La autonomía en términos de contenido, remuneraciones y horas de trabajo

Dentro de los teóricos del capitalismo cognitivo, destaca la definición ofrecida por Fumagalli (2010: 202) de trabajo autónomo como aquel cuya prestación laboral otorga márgenes de discrecionalidad al trabajador en relación con su contenido, con la remuneración y con el tiempo de trabajo. Por eso sostiene que es menos alienado que el trabajo fordista pero con mayor incertidumbre por la competencia del mercado.

Al respecto de la noción de contenido, como primer elemento característico de la autonomía en el proceso de trabajo, ya se ha mencionado el cuestionamiento de Carchedi (2014: 70) de que como los capitalistas poseen los factores objetivos del proceso de trabajo y, también, poseen los medios objetivos de transformación de los valores de uso mentales, por tanto ellos deciden que conocimiento debe ser producido, cómo y para quién.

En este sentido, y como se ha destacado respecto de las arquitecturas informáticas, son los capitalistas quienes detentan el poder para definir los problemas que deben ser resueltos de acuerdo con sus fines. Así, son propietarios de los medios

mentales de las transformaciones mentales: el conocimiento producido debe corresponder con la racionalidad del capital y no la del trabajo²⁷.

Como se ha especificado anteriormente, esta crítica es válida para los obreros colectivos $(s, -s)_j$ pero no necesariamente para todos los obreros colectivos $(s, -s)_i$, es decir, dentro de las comunidades de programación libre y abierta, ya que sus prácticas entran en una contradicción con el poder de mando del capital a partir de la década de los 80 del siglo pasado mediante la singular figura de Richard Stallman y sus proyecto GNU, sistema operativo que no circula como mercancía, sino que es distribuido sin precio monetario ni la dinámica excluyente de tecnologías como Apple o Microsoft.

Sin embargo, como se precisó en el apartado anterior, esto motivó la creación de otros proyectos menos radicales como Linux, donde los códigos fuente de sus lenguajes de programación son propiedad comunitaria (pública no privatizable) en diversos tipos de esas licencias, pero son utilizados para producir otros sistemas informáticos que si circulan como mercancías, como es el caso de los servicios de RedHat, o el caso de Apache, donde se han generado modelos de negocio basados en la venta de módulos a la medida, fenómeno observado en los sistemas de gestión de aprendizaje (LMS) en código abierto, como Moodle o Claroline.

En relación con el tema de las remuneraciones que destaca Fumagalli, se volverá sobre ello en el siguiente capítulo, cuando se trate el proceso de valorización. Sin embargo, en este capítulo es necesario destacar, tanto la pregunta que formulan Lerner y Tirole (2002: 216) de “Por qué las personas participan? ¿Por qué hay proyectos de

²⁷ Para este autor, en esto consiste la lucha de clase cognitiva: que la falsa consciencia del trabajo no es una reflexión distorsionada de la realidad sino aceptar la racionalidad del capital como propia (Carchedi, 2014: 70).

código abierto en primera instancia? ¿Cómo reaccionan los vendedores comerciales ante el movimiento de código abierto?”, como la respuesta de Fielding, Roberts y Slughter (sf:2), quienes recuperan los dos argumentos centrales de Raymond: la cultura de compartir y colaborar, y el modelo del artesano cuyas motivaciones incluyen crear trabajos admirables por su belleza.

Los desarrolladores están motivados mediante el reconocimiento de sus contribuciones por los demás. Más aún, en el caso de la comunidad Apache que investigan, destacan que tiene una estructura asamblearia de red que incluye a colaboradores tipo "commiter", lo cual es un privilegio porque entregan los cambios al código directamente al repositorio de código fuente, y eluden el paso por un intermediario para que los cambios sean incluidos en el proyecto.

Estos colaboradores se encargan de gestionar los subproyectos y para ser miembros de la Fundación Informática Apache (ASF por sus siglas en inglés) deben ser nominados y elegidos por los integrantes de acuerdo con sus colaboraciones. Así, gozan de prestigio porque pueden ser nominados para director de ASF y con ello tomar decisiones sobre la gobernanza corporativa. Según estos autores, no hay una relación entre la remuneración y los aportes a la comunidad Apache, y además destacan que sus patrones capitalistas tampoco se los reconocen.

El carácter singular de estos obreros colectivos $(s, -s)_i$, como Apache, aparentemente autónomo en términos de remuneraciones, puede ser cuestionado a partir de lo que la teórica del capitalismo cognitivo, Morini (2007: 42) denomina feminización del proceso de trabajo.

Este fenómeno consiste en que las tareas se distribuyen a semejanza de las del cuidado, crianza y otras de carácter doméstico, sexualmente atribuidas a las mujeres²⁸, noción también presente en Fumagalli (2010: 218), quien justifica esta contradicción entre una aparente autonomía y un aumento de la extensión y la intensidad de la jornada laboral, porque actualmente es un periodo de transición y por ello las manifestaciones tienen ese carácter contradictorio.

En este sentido, Erneroglu (2011: 43) también destaca que en el capitalismo contemporáneo el proceso de trabajo ha adoptado características de género y por ello el trabajo femenino es la base de su funcionamiento actual.

En este caso, hay una extensión de la jornada laboral más allá del horario fijado por ley, lo que favorece que los proyectos puedan realizarse en menos días calendario, es decir, se realizan los proyectos con menos días calendario pero trabajando más horas cada día. No es un aumento de las fuerzas productivas (como por un mejoramiento en las computadoras utilizadas) ni de la intensidad del trabajo sino una prolongación de la jornada laboral mediante trabajo paralelo.

Contra este argumento de Fumagalli, Vázquez y García (2017: 510) consideran que la extinción formal de la jornada laboral, prolongándose durante las 24 del día es una forma de neotaylorismo de origen japonés, y que es incorrecto e impreciso, afirmar que eso se deba a la configuración técnica de la informática ya que, por un lado la externalización, característica en las empresas del sector, siempre ha sido una ventaja en general, y el colonialismo es la primera muestra histórica moderna de ello y, por otro lado, que lo particular de la informática es que las comunidades de programación libre y

²⁸ Crowley, et. al. (2010: 437-439) destacan el caso de las enfermeras, quienes son amenazadas con el despido para intensificar la inseguridad laboral y consideran que la frustración laboral da lugar a conflictos familiares, abuso de drogas y agotamiento.

abierta demostraron que la integración 24/7 es posible y más eficiente que las formas de integración 24/7 corporativas.

También es válida la crítica de Ross (2013: 215) de que si bien el voluntarismo de estas comunidades no es tan alienante como su contrapartida corporativa, es un error considerar que este proceso de trabajo manifiesta una verdadera democracia radical a nivel industrial, pues solo expresa los límites contemporáneos que dicha tecnología posibilita (2013: 215)

Finalmente, sobre la autonomía en relación con el tiempo de trabajo, las ventajas técnicas que la programación libre y abierta mantiene sobre la programación corporativa bajo el mando capitalista son precisamente evidencia, de que lejos de que la cantidad de horas de trabajo haya dejado de ser el centro gravitatorio del proceso de producción, está en el centro

Precisamente los obreros colectivos $(s, -s)_i$, como los que soportan a Apache y a Linux, desplazan a los obreros colectivos $(s, -s)_j$, como los de Netscape y Microsoft, porque son capaces de realizar las 7 tareas consumiendo una combinación k de factores objetivos del proceso de trabajo, y produciendo los z módulos del sistema informático v , en menos horas de trabajo (HT^i) que las (HT^j) que se requieren bajo el paradigma privativo.

Así, si bien son verdaderas las ventajas técnicas de los sistemas informáticos libres y abiertos sobre sus pares privativos que destaca Blondieu (modularidad, ley de muchos ojos, reciclaje de código, prácticas comunitarias, como se detalla en el siguiente apartado), de ello no se sigue que el carácter emancipador de las experiencias de clase de los obreros colectivos sea lo predominante, por cuanto la autonomía que “disfrutan” los obreros parciales dentro de estos obreros colectivos $(s, -s)_i$ se fundamenta en una

mayor eficiencia en términos de requerir una menor cantidad de horas de trabajo para producir un valor de uso informático en comparación con las horas de trabajo consumidas por los requeridos por obreros colectivos $(s, -s)_j$.

7. Fundamentos económicos de la productividad de los obreros colectivos $(s, -s)_i$

Desde el punto de vista del valor de uso, y del proceso de trabajo que se estudia en este capítulo, la productividad es una medida de la cantidad de valores de uso producidos en una determinada cantidad de horas de trabajo, dada una intensidad media del trabajo y un desarrollo histórico dado de las fuerzas productivas. En este apartado se tratará de la productividad en este sentido, la cual está en relación directa con la fuerza productiva del trabajo que interviene en el proceso de producción: los factores objetivos (materias primas, maquinaria, herramientas, combustibles) y los factores subjetivos (fuerza de trabajo)²⁹.

Las cuatro principales características distintivas del obrero colectivo $(s, -s)_i$, en tanto que factor subjetivo del proceso de producción, que le permite reducir la cantidad de horas de trabajo (HT) y mejorar la calidad de sus sistemas informáticos son: i) la modularidad; ii) el reciclaje de código, iii) la ley de los muchos ojos, iv) las prácticas comunitarias. Existen otras características, pero exceden los alcances de esta investigación.

²⁹ Desde el punto de vista del valor, y del proceso de formación del valor y la valorización que se estudia en el siguiente capítulo, una hora de trabajo socialmente necesaria para la producción de cualquier mercancía añade siempre la misma cantidad de valor, independientemente de la cantidad de valores de uso producidos en esa hora, por lo que si aumenta la fuerza productiva del trabajo, y por ende se producen más valores de uso por hora, entonces la cantidad de valor en cada mercancía es proporcionalmente menor. En este sentido es que la productividad es el recíproco del valor: los aumentos en la productividad disminuyen la cantidad de valor añadido a cada mercancía individual, aunque no la masa de valor producido.

7.1 La modularidad

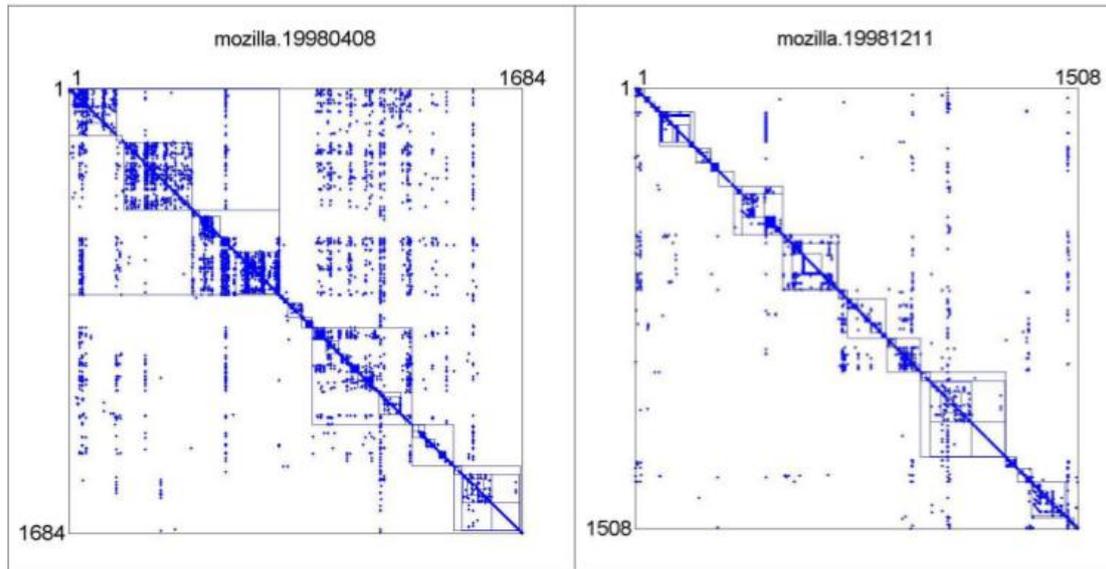
La modularidad es una característica de las arquitecturas de sistemas informáticos para subdividirse en partes más pequeñas y simples, tal que el funcionamiento inadecuado de una de las partes no interrumpa el funcionamiento correcto de la demás. Esta característica ha sido ampliamente señalada como uno de los principales factores técnicos que explican el éxito de casos como Linux y Apache, y que ha sido examinado y documentado para el caso de Netscape/Firefox a finales de los años 90.

Así, la noción de modularidad, ya muy presente en el análisis de Braverman, se manifiesta en relación con el concepto de dependencias y que se refiere a las interferencias que se introducen entre sí los programadores informáticos al ir realizando los cambios en los archivos, por ejemplo dentro de las ramas (“branches”) en Github, lo que supone un problema en la planificación de la asignación de tareas.

Más allá de la gran cantidad de empresas que empezaron a seguir la estrategia de liberar su código fuente, el caso de Mozilla también destaca por haber sido ampliamente investigado, fenómeno que se repite con más frecuencia que en relación con los sistemas informáticos propiedad de las corporaciones capitalistas, ya que al estar protegidos por licencias que hacen pública su información, sirven como un objeto de investigación, y más allá de ofrecer resistencia, se ven beneficiados por dichas investigaciones.

Precisamente MacCormack, Rusnak y Baldwin (2006) estudian el caso de la modularidad de Mozilla antes y después de liberar su código y ser intervenido siguiendo una arquitectura más modular, tal y como se muestra en la siguiente figura:

Figura 8. Modularidad del código de Mozilla antes y después de migrar a programación abierta



Fuente: MacCormack, Rusnak y Baldwin (2006: 33)

En la figura 8 se muestra la comparación de las dependencias entre los módulos, interfaces y otras partes de la arquitectura del navegador web de Mozilla. En el cuadro de la izquierda se notan patrones mucho más dispersos y puntos de gran concentración.. Las arquitecturas y diseños concebidas para sistemas informáticos bajo propiedad corporativa, como en el caso del código de Mozilla antes de liberar su código, favorecía la existencia de dependencias de los programadores encargados de las otras tareas (diseño, codificación, pruebas, integración, y mantenimiento) con los capataces para que estos pudiera ejecutar su tareas del control y vigilancia dentro de los obreros colectivos $(s, -s)_j$, bajo el poder de mando capitalista.

Al cambiar de paradigma, y liberar su código, este tipo de controles numéricos establecidos estructuralmente en lo interno de estos sistemas informáticos desaparece

de su configuración misma. Aparentemente, el poder de mando capitalista ha sido eliminado incluso desde una perspectiva técnica. Esto se origina en que al crear arquitecturas y diseños más modulares, los sistemas informáticos libres y abiertos, permiten que sean los voluntarios quienes decidan cuáles módulos producir, cuándo y cómo

Sin embargo, la modularidad es una característica también de la ingeniería comercial de sistemas informáticos, no solo del paradigma de programación libre y abierta, tal y como queda documentado por Zmud (1980: 46) quien la consideró en su momento, como una práctica moderna dentro del proceso de desarrollo. De ahí, que como consideran los mismos MacCormack, Rusnak y Baldwin (2006: 6) para los sistemas informáticos libres y abiertos la modularidad es una condición necesaria de este método de desarrollo para tener éxito.

7.2 El reciclaje de código

La segunda característica que permite reducir las horas de trabajo, es el reciclaje de código, y del cual la figura 8 es un claro ejemplo. El reciclaje de código se refiere a la práctica entre los miembros de estas comunidades de programadores de utilizar el código fuente de otros programadores, quienes anteriormente se hayan enfrentado a un determinado problema, y para ello recurren a plataformas especializadas, como Stackoverflow o Github, o como el caso del foro de Moodle mencionado anteriormente, de donde consultan dicha solución y la prueban en su propio problema.

El reciclaje de código es una práctica característica de este tipo de proyectos, porque como con el caso de Netscape/Firefox, deben hacer públicos los código fuente, lo que los convierte en objetos de continuas investigaciones y desarrollos, fenómeno que no acontece con los sistemas informáticos privativos, porque precisamente buscan

ocultar lo más posible esa información, como es el caso de Microsoft y particularmente de Apple. Por ello, precisamente es que entre la investigación empírica sobre los sistemas informáticos, hay mayoritariamente datos de casos sobre sistemas libres o abiertos.

El reciclaje de código es una práctica de obreros colectivos $(s, -s)_i$ favorece el desarrollo en paralelo, porque aprovecha, por un lado, el trabajo pretérito de otros programadores informáticos y, por el otro, aprovecha las potencialidades creativas en el desarrollo de prototipos de otros programadores.

Así, cuando por características de arquitectura o del diseño, o incluso por la distribución horario-territorial de los obreros parciales, favorece que se dé el desarrollo en paralelo, aparece el problema de los emparejamientos, que es cuando las modificaciones de un programador afectan los archivos que requiere otro programador.

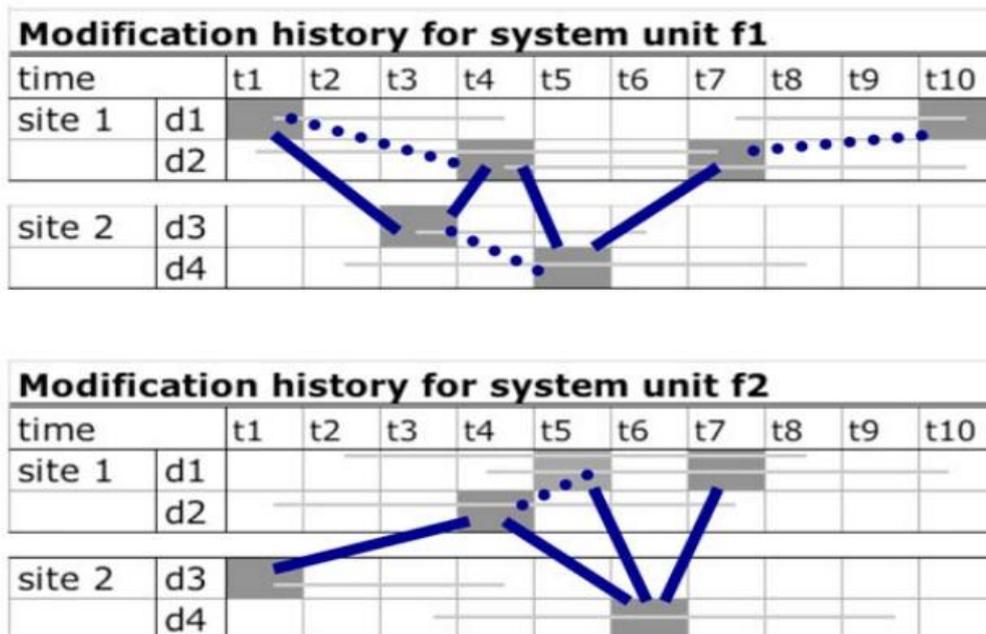
En estos casos se revela la necesidad de planificar este proceso para los obreros parciales, fenómeno que bajo el poder de mando del capitalista se pretende resolver a través de la tarea s de ejercer el control y vigilancia. Esto es más frecuente en los proyectos grandes y largos de desarrollo de los sistemas informáticos y resulta aún más importante para el desarrollo global de los sistemas informáticos del tipo desterritorializado (“offshore”) y externalizado (“outsourcing”) por la variedad geográfica.

Ye, Nakakoji y Yamamoto (2008: 182) analizan el emparejamiento de programadores como un patrón recurrente durante el proceso de desarrollo de los sistemas informáticos y destaca que la dependencia entre las tareas de los desarrolladores para plantear que dichas dependencias (recíprocas) se relacionan con los costos de comunicación y coordinación, por lo que controlarlas es controlar las

segundas. Cambios minúsculos en la arquitectura implican múltiples cambios en las dependencias entre desarrolladores. Cuando dos desarrolladores modifican una unidad crean un potencial conflicto y dependencia entre esos dos desarrolladores, lo que da lugar a la necesidad de coordinación. De esta forma, el emparejamiento o acoplamiento de programadores se refiere a la dependencia de tareas causada por la modificación conjunta de algún sistema unitario por múltiples desarrolladores.

Con base en esto, ofrecen una herramienta de optimización en la planificación de la organización de las tareas, que denominan “matriz de asignación de tareas y de dependencia de tareas”, donde una celda (ij) representa un archivo i y uno j que son cambiados simultáneamente en algún momento (2008: 183), tal y como se muestra a continuación:

Figura 9. Matriz de asignación de tareas para programas informáticos deterritorializados de gran escala



Fuente: Ye, Nakakoji y Yamamoto (2008: 186)

En la figura 9 se muestra el historial de modificaciones de cuatro programadores informáticos, obreros parciales realizando tareas, en las unidades de dos sistemas. Este tipo de situaciones, en las que la coordinación y colaboración entre obreros parciales son concebidas por los autores como dependencia sociotécnica y se refiere a la dependencia modular elaborada a partir del flujo de trabajo y el historial de desarrollo para determinar las dependencias particulares entre desarrolladores.

Esto les permite definir el acoplamiento entre desarrolladores como una función del tiempo (time slider), que es la distancia entre la modificación 1 y la 2. La variable t_s sirve para medir la intensidad del acoplamiento también para aproximar la convención de compromiso (2008: 184-185)

Utilizan su métrica para establecer valores para el proyecto del servidor Apache HTTP para el periodo 2000-2008 y encuentran que el acoplamiento de desarrolladores y el acoplamiento espacial es casi el mismo, y esto se debe a que es un proyecto de código abierto en el que esto sucede muy frecuentemente (2008: 188- 189)

Este hallazgo revela, por un lado, que el tiempo de trabajo es la unidad de medición intrínseca de los procesos de programación informática y, por otro lado, que el capataz encargado de vigilancia y control en los obreros $(s, -s)_j$ es una imposición de la clase capitalista que ni siquiera logra disminuir la cantidad de horas de trabajo requeridas, ya que este servidor Apache HTTP está desarrollado por un obrero colectivo $(s, -s)_i$, es decir, autogestionado sin la necesidad de esa figura autoritaria, y con una productividad en términos de valor de uso mucho mayor.

7.3 “La ley de muchos ojos”

La tercera característica, denominada “La ley de muchos ojos” por Raymond (2001: 30) en un texto múltiples veces citado bajo el título “La Catedral y el Bazar”,

consiste en que si hay suficientes ojos, todos los errores (“bugs”) son atendidos, y que Lerner y Tirole (2002: 229) intentan explicar bajo el concepto de economías de escala.

Esta característica del paradigma de programación libre y abierto presenta la dificultad de formalización que involucra una cantidad indeterminada de horas de trabajo (HT) que integra el obrero colectivo $(s, -s)_i$ ya que no puede ser medido por las agregación de las horas de trabajo (HT) que aporta cada obrero parcial en su tarea s .

Apache es un ejemplo claro de ello, ya que se sostiene por las colaboraciones en horas de trabajo (HT) materializadas en líneas de código por parte de millones de programadores informáticos cada año, a diferencia de los obreros colectivos $(s, -s)_j$ de 7 a 10 obreros parciales bajo el poder de mando de un capataz que personifica subjetivamente los intereses de la clase capitalista a través del ejercicio del control sobre el proceso de trabajo, y a los cuales se les contabiliza diariamente la cantidad de horas de trabajo (HT) que aportaron al proyecto mediante el entorno integrado de desarrollo (IDE)

En este sentido, Zayout y Hajjdiab (2013: 2426) investigan si el uso de entornos integrados de desarrollo (IDE, por sus siglas en inglés) reduce las horas de trabajo de un equipo de 13 programadores informáticos de diferentes niveles de experiencias.

Sus resultados no solo confirman la prevalencia, 40 años después, de los problemas de atrasos en las entregas, altos costos de desarrollo, clientes insatisfechos y altos costos de mantenimiento que en los 80 ya había destacado Zmud (1980: 46), sino que además ofrece evidencia de que en el Visual Estudio se gasta mucho tiempo buscando material en internet y en ayuda local.

Esto ocasiona que el depuramiento del código (“debuggin”) puede consumir hasta el 80% del tiempo de los programadores en ciertas circunstancias, y es difícil diferenciar entre experimentar y aprender durante este proceso. Muchos errores (“bugs”) parecen simples al principio, pero pueden consumir tiempo significativamente, y los programadores reportan que frecuentemente las soluciones para los errores (“bugs”) solo pueden ser encontradas en las comunidades de desarrolladores en internet, por lo que estas soluciones suelen ser aplicadas sin saber exactamente cómo lo hacen (Zayout y Hajjdiab, 2013: 24-29).

Mas aún, las propias reflexiones de autor del ejemplo anterior constituyen una evidencia de que el tiempo de trabajo, y en particular las horas de trabajo que dura la jornada laboral, en términos medios o socialmente necesarios, tiene un interés científico, más allá del valor ideológico que podría achacársele al marxismo por fundamentarse en una teoría del valor trabajo, aunque la misma no fuera originalmente formulada por Marx (Riesznik, 2014: 6).

De esta forma, los esfuerzos de la clase capitalista por controlar todos los aspectos de este proceso de trabajo, aún aplicando el control numérico mismo dentro de la programación informática a partir de los IDE, siguen resultados infructuosos, y por eso han cambiado su estrategia, participando ellas mismas dentro de estas comunidades para tratar de apropiarse de sus recursos, como se revisará cerca del final del siguiente capítulo.

7.4 Las prácticas comunitarias

La cuarta característica de los obreros colectivos $(s, -s)_i$ que les permite ser más eficientes en términos de reducción de horas de trabajo en relación con los

directamente bajo el mando capitalista, se fundamenta en las prácticas de colaboración a partir de las cuales organizan su proceso de trabajo.

Las tres características anteriores, consideradas desde el punto de vista de las experiencias emancipatorias en las que se funda la noción de autonomía para el pensamiento marxista desplegado en esta exposición, tienen como fundamento precisamente estas prácticas comunitarias, ya que las mismas contravienen no solo el interés de obtener una ganancia por parte de la clase capitalista, sino que además se oponen antagónicamente a las aspiraciones de apropiación privada de los saberes de dicha clase y al mismo tiempo, al control y vigilancia sobre los procesos de trabajo que intentan imponer los capitalistas como clase.

De esta forma, la modularidad es implementada por los obreros colectivos $(s, -s)_i$ precisamente para favorecer las colaboraciones voluntarias y entusiastas de obreros parciales bajo prácticas comunitarias. También el reciclaje de código constituye una práctica colaborativa de aprovechamiento del trabajo pretérito de otros obreros parciales y colectivos.

La ley de muchos ojos, soportada técnicamente por la modularidad y el reciclaje de código, es fundamentalmente una práctica comunitaria de colaboración que busca resolver problemas comunes a los que se enfrentan obreros parciales como parte de los obreros colectivos, tanto $(s, -s)_i$ como $(s, -s)_j$, es decir, en una dinámica de solidaridad que incluye también a los obreros parciales bajo el poder de mando capitalista.

Es por ello que estas prácticas comunitarias, son un factor destacado entre las investigaciones empíricas a cargo de capataces, “managers”, encargados del control de los obreros colectivos $(s, -s)_j$ bajo el poder de mando capitalista como es el caso,

abordado en la sección anterior, sobre el uso de entornos de desarrollo integrado (IDE) para controlar la intensidad del trabajo en los programadores informáticos.

También es el caso mencionado en el apartado sobre las tareas de instalación, de una versión de Moodle en la tecnología privada Azure de la Corporación Microsoft, de precisamente una práctica comunitaria que consiste en compartir el código y su experiencia como obrero parcial, lo que permite que muchos otros usuarios, con diferentes niveles de competencia informática, puedan resolver sus propios problemas particulares.

La noción de este tipo de prácticas comunitarias también fuera de los trabajos cognitivos (artísticos, intelectuales, creativos) son identificadas.

Díaz Polanco (2011: 23) destaca diversas prácticas comunitarias en los territorios de Oaxaca y Chiapas, como la conservación del genoplasma agrícola, in situ, la cual es una estrategia en la que es vital la participación de las comunidades (saber tradicional y prácticas colectivas) y que tiene la doble ventaja de conservar los hábitats y preservar una mayor cantidad de genoplasma a menor costo.

En este sentido, debe reconocérseles a los teóricos del capitalismo cognitivo, como Fumagalli, su insistencia en incluir la noción de autonomía dentro del pensamiento económico, ya que como destaca Veraza (2007:51) sobre la tradición de autores operaístas en general, resulta un tema subversivo vinculado al interés de esta escuela italiana en insistir en el valor de uso y el comunismo asociado a él.

Sin embargo, también es menester destacar el economicismo subyacente tras el planteamiento de la autonomización de la fuerza de trabajo ante el poder de mando del capital, fundamentado en estas características subjetivas de los obreros colectivos $(s, -s)_i$ que les permite minimizar su cantidad de horas de trabajo, ya que como se ha

argumentado anteriormente solo el segmento radicalizado políticamente a través de las licencias GNU es abiertamente anticapitalista y no busca producir mercancías sino valores de uso, al contrario del otro segmento que se encuentra bajo alguna modalidad de subordinación al poder de mando del capital, ya sea de contenido, de remuneración o de horas de trabajo.

Sin embargo, esta argumentación no es suficiente para refutar la definición de Fumagalli de trabajo autónomo como una fuerza social autónoma con la forma triádica: cooperación, comunicación y autocontrol (o control social) y cuya dialéctica constituye el trabajo cognitivo al que está tendiendo el trabajo vivo (2010: 192).

En este sentido, para este autor la discusión radica en verificar si el trabajo es cognitivo y si la fuerza de trabajo es intelectualidad de masas, y si este último puede ser un sujeto políticamente hegemónico (Fumagalli, 2010: 196)

En primer lugar, debe retomarse el planteamiento ensayado al inicio de este capítulo de que es falso que en el proceso de programación informática la principal fuerza productiva sea la fuerza de trabajo, como intelectualidad de masas o no, ya que sin las materias primas, memoria de almacenamiento y procesamiento de información, las horas de trabajo podrían dar como resultado código informático escrito en hojas de papel, incapaces por sí mismas de operar una máquina mecánica de cualquier tipo.

En segundo lugar, porque en las investigaciones mismas de programación informática se ha encontrado que las restricciones físicas propias de las materias primas en este proceso de trabajo resultan determinantes.

Por ejemplo, Schmidt et. al (2017: 200) encuentran que en la aplicación de las metodologías ágiles en el desarrollo de sistemas informáticos (Scrum, Kamban, programación extrema), si bien los programadores compilan código una vez escrito en

unos cuantos segundos o minutos, construir un prototipo físico de máquina donde se instale el código tardas semanas e incluso meses, porque las características de esos sistemas están limitados por la potencia computacional, y no solo por la experiencia de los programadores y su imaginación.

Por eso sostienen que en la producción de valores de uso puramente materiales, como robots, las restricciones físicas causan dificultades en presentar incrementos del producto, y en responder a los cambios rápidamente. De ahí que las prácticas ágiles son muy poco usadas en el desarrollo de productos físicos como vehículos, entre otros (Schmidt, et. al, 2017: 203)

Es decir, que las características específicas del proceso de programación informática se ven constantemente limitadas por las restricciones físicas de las máquinas computadoras, y más aún cuando se trata de implementar las metodologías de desarrollo de sistemas informáticos en otro tipo de valores de uso. Esto constituye evidencia de las limitaciones de las capacidades de los obreros colectivos ($s, -s$) para hegemonizarse ante la grosera corporalidad de la materia misma que el ser humano transforma durante la jornada laboral.

Sin embargo, debe reconocerse que la organización del proceso de desarrollo de los sistemas informáticos no es fácilmente controlable mediante las técnicas de otros procesos de trabajo.

Por ejemplo, Ahmad, Markkula, y Oivo (2013: 16) analizan la aplicación de la metodología Kanban, utilizada en muchas industrias, a la programación informática y encuentran limitaciones relacionadas con las distancias físicas y las diferencias entre los lenguajes humanos, que conducen a desperdicios en el proceso de desarrollo de sistemas informáticos.

De esta forma, sí hay evidencia de las prácticas comunitarias y colaborativas que han podido ser privatizadas y que dificultan el proceso de automatización de las tareas relacionadas con el desarrollo de sistemas informáticos.

Sin embargo, debe destacarse que el término de “autonomización” es un concepto simplista que enfatiza la ruptura en vez del proceso, como le cuestiona Husson (2003:1) y debe ser reconsiderado a partir de la categoría de clase para el pensamiento marxista.

Por ello, se juzga necesario ofrecer un marco analítico adecuado para incorporar la noción de autonomía fundamentada en estas prácticas comunitarias desde una perspectiva de clase que complemente este abordaje desde la sociología política marxista.

8. División del trabajo y lucha de clases

La comprensión de que el proceso de trabajo está atravesado por la lucha de clases está manifiesta explícitamente por Charles Babbage (209: 250), para quien la amenaza latente de interrupción del proceso de producción de valores de uso por huelgas y sabotajes es un costo que se debe incluir en el precio

La amenaza de paralización del proceso de trabajo constituye en una fuente de poder en sí de los trabajadores y su magnitud es directamente proporcional a la fuerza productiva del trabajo del obrero colectivo: entre más aumente su productividad, en términos de valor de uso, la amenaza de paralización representa un costo mayor para la clase capitalista, la posición que defiende Babbage.

Por ello es que el poder de negociación de los trabajadores aparece vinculado, en primer lugar, con acciones conflictivas, como huelgas y sabotajes, de ruptura súbita, así

como de lucha y rebeldía en las que solo aplica el análisis concreto de la situación concreta³⁰.

En la medida que las características de la acción conflictiva y sus consecuencias no pueden ser previstas apriorísticamente³¹, resulta pertinente considerar con Modonessi (2010: 168-169) que las experiencias de clase pueden mapearse en configuraciones subjetivas de antagonismo (insubordinación), autonomía (emancipación) y subalternidad (subordinación). Cuando hay acumulación de tensiones antagonistas y autonomistas, y las tensiones subalternas y antagonistas entran en proceso de ruptura, la fuerza subjetiva de producción puede entrar en contradicción con las relaciones de poder y acontecer una confrontación que paralice el proceso de trabajo.

Si se acepta que la clase trabajadora en sí, definida por su posición en relación con los medios de producción, se encuentra en un proceso de aprendizaje histórico³², colectivo y colaborativo, caracterizado por contradicciones e indeterminaciones, rupturas y discontinuidades y cuya historicidad se pueden registrar como combinaciones desiguales en las coordenadas de subalternidad-antagonismo-autonomía (Modonessi, 2010: 18), entonces las coaliciones de clase³³ son manifestaciones de la conciencia de clase en el proceso histórico *t* de la clase para sí en la medida que su poder en sí, puramente económico, se convierte en un ejercicio de un poder para sí o *potentia* (Dussel, 2006 : 27).

³⁰ Mao Tse-Tung (1968: 345) destaca el principio de análisis estratégico de Lenin precisamente cuando reflexiona sobre la contradicción y el antagonismo.

³¹ De hecho para Babbage (2009: 244) las huelgas son casi siempre perjudiciales para las partes, primero por el aumento en los precios y segundo porque la mayor parte de las mejoras en la maquinaria son consecuencia de las huelgas entre los trabajadores.

³² Esta es aproximación de autores marxistas de la acción colectiva como Carpenter y Matthews (2003: 6) para comprender las conductas colectivas intra-clase entre trabajadores y sus divisiones étnicas y culturales promovidas por la clase capitalista.

³³ Roemer (1978: 157) sostiene que las coaliciones, burguesa y proletaria, surgen de la percepción mutua de su poder, el proletario en su diario vivir, el capitalista en estado táctico, reevaluando las nuevas opciones.

En este sentido, el proceso de trabajo no solo es más que físico sino que además forma parte de la historicidad de las clases subalternas como experiencia cotidiana concreta, como jornada laboral de 8 horas de trabajo diarias (HT) y como proceso de construcción de identidad colectiva dentro de la jerarquía de la división del trabajo.

El proceso de trabajo es indisociable de cada subjetividad individual y su voluntad para ejecutar las tareas que le corresponden o rebelarse. El alcance conflictual dependerá de las configuraciones subjetivas de subalternidad, antagonismo y autonomía en cada situación concreta³⁴.

En resumen, la voluntad subjetiva, individual y colectiva, para subordinar las capacidades corporales, cerebrales, sensitivas y mecánicas para ejecutar las tareas laborales está en relación directa con su alimentación (mediante una canasta de subsistencia w), descanso y destreza de su cuerpo, pero además está determinado por las experiencias de subjetivación política durante la jornada laboral, en la cual las jerarquías entre trabajadores atraviesan ciclos políticos³⁵ e interfieren con los procesos de producción de valores de uso, incluidos los informáticos.

Por consiguiente el carácter potencialmente autonómico de los obreros colectivos $(s, -s)_i$ no depende exclusivamente de su propiedad colectiva sobre el medio de trabajo, sino además de la configuración de experiencias de emancipación, insubordinación y subordinación en las que se desarrolla su actividad laboral, y por consiguiente depende de esos factores subjetivos el que sus prácticas adquieran un

³⁴ Por ejemplo, Althusser considera que cuando la lucha de la clase obrera no tiene la posibilidad de enfrentarse contra los efectos de la explotación derivados del aumento de la productividad se pasa a limitar esos efectos, luchando contra los ritmos de trabajo, contra las arbitrariedades extremas, contra las horas extraordinarias, contra la supresión de puestos de trabajo, contra el “paro de productividad” (2004: 28).

³⁵ En este sentido, Mandel (1986: 46) recupera la conjetura de Trotsky de que la dinámica de los ciclos económicos está atravesada por los ciclos de lucha de clases, los cuales deben integrarse al análisis pero reconociendo su relativa autonomía.

carácter anticapitalista, o que tiendan a someterse cada vez más al interés capitalista por la tasa de ganancia y la subordinación en términos de contenidos, remuneraciones, tiempos de trabajo y modos de ejecución de las tareas durante el proceso de trabajo.

9. Flexibilidad modular, control numérico y analfabetismo informático

Braverman fue un crítico mordaz de las tecnologías informáticas, incipientes en su época, porque consideró que formaban parte del “management científico” en tanto que estrategia de la clase capitalista para controlar todos los aspectos del proceso de trabajo y debilitar con ello el poder de negociación de los trabajadores.

En primer lugar, destacó que la programación informática era “un concepto revolucionario para la industria” por la confiabilidad en sus operaciones y la facilidad para realizar reparaciones mediante reemplazo modular (Braverman, 1998: 135-136).

En segundo lugar, y de manera más general, consideró que el efecto del “management científico” en la clase trabajadora era una degradación que consistía en una pérdida de competencias y conocimientos sobre el proceso de trabajo al servicio de una maquinaria cada vez más compleja, lo que ocasionaba que la clase trabajadora creciera solamente en ignorancia e incapacidad... que se convirtiera en una fuerza de trabajo al servicio a la máquina (Braverman, 1998: 134)

La trayectoria histórica seguida por la programación informática hasta nuestros días confirma que tanto el reemplazo modular como una segmentación tecnológica entre la fuerza de trabajo son fenómenos que cobran más relevancia en los mercados de trabajo y la economía mercantil en general.

En este sentido, puede considerarse que esos dos aspectos destacados por Braverman sobre la informática han sido incorporadas por la clase capitalista dentro

de su estrategia en la lucha de clases mediante el control numérico bajo la forma de: i) “flexibilidad modular”; ii) y de “analfabetismo informático”.

En el primer caso, la máquina envía señales de cada acción de cada obrero parcial durante las horas de la jornada laboral: no solo no hay posibilidad de sabotaje individual o colectivo, sino que esas señales adoptan la forma de bytes de información generados por fracciones de segundo, y en este sentido es semejante al control directo, que es como una cámara de seguridad externa, pero que en el control numérico se torna en una cámara de vigilancia interna ineludible.

Esto ocasiona que la cantidad de tráfico de datos aumente vertiginosamente, ya que no solo se registra el producto final, el sistema informático finalizado como código-objeto, sino todo el proceso de modificaciones parciales, los tiempos de cada uno de los programadores informáticos, es decir, genera su propia base de datos. Plataformas como Github permiten a los desarrolladores administrar esas bases de datos y utilizarlas como herramientas de planificación en el modelo abierto de programación informática, es decir, como repositorios. En esto, el mecanismo de vigilancia es dichas comunidades parece semejante al que imponen los entornos integrados de desarrollo (IDE).

De esta forma, el control numérico posibilita la flexibilidad modular, ya que el mecanismo de vigilancia en el que se soporta el control numérico permite la flexibilidad modular como criterio de organización del proceso de trabajo.

En el segundo caso, se da una segmentación de los usuarios de sistemas informáticos según su nivel de competencia tecnológica, es decir, la clase capitalista promueve intencionalmente el analfabetismo informático de los usuarios para obtener información en tiempo real de todas sus actividades cotidianas, incluida la jornada laboral, con el fin de ejercer control burocrático sobre los ciudadanos y medir los

tiempos de trabajo con miras a automatizar todos los procesos y sustituirlos mediante robots dotados de inteligencia artificial, tema abordado por los teóricos del capitalismo cognitivo a través del confuso término analítico, de “prosumidor”.

9.1 La flexibilidad modular como estrategia de debilitamiento del poder de negociación de los trabajadores informáticos

En el apartado anterior se destacó que la paralización del proceso de trabajo constituye un poder en sí de los obreros colectivos en general, $(s, -s)$, y que su magnitud es proporcional a la cantidad de valores de uso.

La estrategia seguida por la clase capitalista para debilitar este poder de negociación de los trabajadores informáticos, o “cognitariado” término utilizado por teóricos del capitalismo cognitivo para referirse a la clase a la que pertenecen los programadores informáticos (Moulier Boutang, 2012: 185), puede definirse como flexibilidad modular a partir del “ principio de Babbage” (2009: 137-138) y la noción de “reemplazo modular” presente en Braverman (1998: 136) y se fundamenta en la coexistencia y competencia entre los obreros colectivos $(s, -s)_i$ y los obreros colectivos $(s, -s)_j$ que permite la amenaza de sustitución de cualquier obrero parcial³⁶, e incluso del obrero colectivo en cualquier fase del proceso de producción³⁷, por otros ubicados en cualquier parte del mundo mediante externalización ante los episodios de insubordinación locales y nacionales.

³⁶ También Moulier Boutang (2012: 83) señala el aniquilamiento del sabotaje individual como acción conflictual del obrero parcial, mediante técnicas como el control de calidad.

³⁷ Autores como Guillén (2013: 295) y Marazzi (2014: 45) enfatizan la financiarización, impulsada por la aplicación de las tecnologías informáticas a las finanzas, como el fenómeno que posibilitó esta competencia a muerte entre la clase trabajadora a nivel internacional, e incluso para el segundo autor, defensor de las tesis del capitalismo cognitivo, esta aplicación de la informática tuvo efectos estructurales en la relación capital y trabajo.

La flexibilidad modular permite comprar solo la cantidad exacta de fuerza de trabajo, según el principio de Babbage, y reemplazarla sin afectar la intensidad del proceso de trabajo para cumplir con los tiempos requeridos por las corporaciones capitalistas para asegurarse una ganancia con la venta de la mercancía cognitiva.

Esto se origina en que ante el envío de información en tiempo real sobre las actividades de cada programador y las señales vinculadas con potenciales afectaciones del cumplimiento de los plazos, es posible contratar a otro obrero parcial que realice la tarea s en cuestión, ya sea que redacte esa parte del código, o que aplique las pruebas, o que realice la integración, sin que se afecte la intensidad del proceso de trabajo ni el plazo de entrega, es decir, anulando totalmente el poder de negociación de los trabajadores originado en su control de los tiempos de trabajo.

Braverman sostiene que la computación permite introducir el control numérico en el proceso de trabajo. En este sentido, los entornos integrados de desarrollo (IDE por sus siglas en inglés) se consideran el método más sofisticado de control numérico.

Todo tipo de control (directo e indirecto) se define a partir de un mecanismo de vigilancia, el cual recopila información que permite determinar si un proceso de trabajo se realiza con las especificaciones de ejecución, tal que tenga como resultado el valor de uso en la cantidad exacta de horas de trabajo asignadas a cada obrero parcial dentro de un obrero colectivo.

El control burocrático, por ejemplo, se define a partir de la vigilancia civil de la conducta individual del trabajador fuera de las horas de trabajo a través de las oficinas de recursos humanos, encargadas de la selección y el reclutamiento de la fuerza de trabajo (Gordon, Edwards y Reich, 1986: 241).

El control numérico, en cambio, se define a partir de un mecanismo en tiempo real de vigilancia individual automatizada (Braverman, 1998: 136).

Cada uno de los mecanismos de vigilancia asociados a un tipo de control se basa en el sistema de comunicación básico, es decir, un emisor, un receptor, un canal, una señal y un código. El emisor es el vigilado y el receptor el vigilante.

En el caso de control directo las señales y mecanismos son la observación inmediata durante la jornada laboral, mientras que en el control técnico esas señales adoptan la forma de índices de intensidad del trabajo en cada ciclo .

El control burocrático adopta la forma de índices de conducta de la subjetividad individual en la configuración subalternidad-autonomía-antagonismo, es decir, ausentismo, su cumplimiento de metas individuales en relación con el promedio, su participación en huelgas o conflictos laborales, en relación con los incentivos de involucramiento como opciones de compra de acciones de la empresa, bonificaciones individuales, servicios de salud adicionales.

El control numérico, al introducir una vigilancia en tiempo real de todas las actividades realizadas por el programador durante las horas de trabajo y capturar toda la información que este genere, abre la posibilidad de que en el futuro incluso esta tarea s de codificación pueda ser, no solo estandarizada, sino automatizada. El asunto de interés para esta investigación radica en que si el proceso de trabajo es bajo el mando capitalista se introduce la vigilancia a través del IDE, pero si el proceso es colaborativo, la vigilancia es pública y por tanto no es utilizada solo por una clase para luchar contra la otra, sino por ambas clases.

Por ejemplo, el carácter abierto de Github le permitió a programadores chinos eludir la censura para quejarse de las largas jornadas laborales a las que se ven

sometidos: 12 horas diarias durante seis días a la semana por empresas como Alibaba, Huawei, Bytedance, DJI, entre otras (Liao, 2019) como trocando sus herramientas de trabajo en armas dentro de la lucha de clases.

Así, a pesar que los mecanismos de vigilancia y control de los obreros colectivos $(s, -s)_i$ y $(s, -s)_j$ parecen semejantes, con las herramientas utilizadas en el desarrollo en código abierto, como Github, la información es completamente pública, y por tanto se utiliza para medir la magnitud cualitativa y cuantitativa de su aporte al desarrollo del sistema abierto, y se utiliza como señal de estatus técnico dentro de las redes de programadores, pero también como arma en manos de la clase obrera dentro de la lucha de clases.

De esta forma, la información que genera la vigilancia impuesta por el control numérico es utilizada de dos maneras: una privada por parte de las corporaciones capitalistas dueñas de esos valores de uso informáticos y otra comunitaria colectivista por los obreros colectivos $(s, -s)_i$, cada una con su forma de propiedad específica y con fines diferenciados.

Esta diferencia queda patente en la web 2.0 con las redes sociales privadas como Facebook, Twitter o Instagram, donde la popularidad de sus usuarios es explotada al máximo mediante la captura de datos a través de la publicidad y luego vendida a diferentes compradores.

Sin embargo, la confusión ocasionada por la generación de información debido al control numérico con tecnologías privativas y tecnologías libres y abiertas, ha ocasionado que incluso autores críticos de las tesis del capitalismo cognitivo, como Fuchs (2015: 34), interpreten erróneamente este secuestro de la información como

generación de plusvalor por parte de los “prosumidores”, usuarios aparentemente explotados.

De esta manera, el control numérico permite las mediciones más precisas sobre la productividad (en términos de valor de uso) individual y colectiva durante el proceso de trabajo porque se fundamenta en una vigilancia automatizada en tiempo real, y a la vez permite el reemplazo modular de los obreros parciales, características que se han extendido fuera del campo de la informática mediante los valores de uso informáticos mismos.

En este sentido, puede sostenerse que el control autonómico no es la modalidad subjetiva correspondiente con la flexibilidad modular propiamente dicha, esto es, que sea un proceso de desarrollo completamente flexible modularmente, ya que aunque cada obrero parcial debe asumir un control autonómico de su propio proceso de trabajo, es decir, vigilarse a sí mismo, el control numérico está presente para informar a los capataces de cualquier desviación en la conducta esperada por las clases capitalistas.

9.2 Tipología de usuarios de valores de uso informáticos según su nivel de alfabetización informática

Los obreros colectivos $(s, -s)_i$, de las comunidades de programación libre y abierta son heterogéneos en relación con las licencias de propiedad intelectual, pero además en su composición técnica, geográfica, demográfica e institucional.

Mufatto (2006: 57) recopiló las características demográficas de estas comunidades para el 2001, y entre los rasgos más destacados encuentra que eran mayoritariamente hombres, (98%,) y más del 70% oscilaba entre los 22 y 27 años. Además, la mayoría tenía un grado universitario, pero solo el 10% PhD. Adicionalmente, destaca que el 60% eran profesionales de las tecnologías de la

información, y un 30% estaba vinculado con alguna institución académica, y de estos 60% eran estudiantes. Finalmente, señala que más del 80% vivía en Norte América y en Europa, pero advierte que esta composición cambiaría en términos de la distribución de la edad y la distribución geográfica por la incorporación de China y la India.

Algunas de estas características son confirmadas por Borja (2016: 105-119), quien encuentra que las instituciones educativas son un agente ineludible en el “ecosistema” de la programación libre, pero no registra si hubo una redistribución geográfica entre los usuarios, ya que de hecho, fuera de Europa, solo Estados Unidos y Australia se encuentran dentro de los 10 países que más utilizan y contribuyen al desarrollo de estos sistemas.

Si bien estos datos son insuficientes para ofrecer una tipología empírica de la composición técnica de los obreros colectivos $(s, -s)_i$, esta puede formularse a partir de la competencia tecnológica en el dominio de lenguajes de programación y comprensión del funcionamiento de los dispositivos electrónicos y máquinas computadoras, en consonancia con la denuncia de Braverman sobre la degradación de la fuerza de trabajo ante la máquina computadora.

En primer lugar, se encuentran los usuarios especialistas, que se compone de todos los ingenieros en sistemas computacionales, desarrolladores de sistemas informáticos y técnicos en sistemas y redes. Se consideran especialistas porque su medio de trabajo son los lenguajes de programación mismos, como Python, C#, Java o PHP, es decir, sistemas informáticos para producir sistemas informáticos (incluidos los videojuegos).

Este primer tipo de usuarios son los obreros parciales que se han considerado para las tareas requeridas para el desarrollo de sistemas informáticos y que constituyen el trabajo productivo desde el punto de vista de la clase capitalista.

En segundo lugar se encuentran los usuarios competentes, que se compone de aquellas personas que sin ser especialistas en programación informática poseen conocimientos sobre alguno, o varios, lenguajes de programación suficientes como para producir sus propios códigos y leer el código elaborado por otros, y por ello se consideran con algún nivel de alfabetización informática.

Este tipo de usuario puede o no, ser especialista en otras áreas del conocimiento diferentes a la programación informática (arquitectos, ingenieros, químicos, biólogos, matemáticos, estadistas, economistas, sociólogos, medicina) y en dicho caso, utilizan los lenguajes de programación como material auxiliar en sus investigaciones dentro de su campo específico, y por tanto, utilizan estos lenguajes para sus propios problemas y comparten las soluciones con otros, creando comunidades de programación libre basadas en aportes de no especialistas pero con un dominio profesional de algún lenguaje en específico, como R, que es el caso de sistemas para producir datos en cualquier disciplina, o como PHP (en el que está escrito Moodle), que sirve para producir cursos online de cualquier disciplina.

Finalmente, se encuentra el usuario pasivo o analfabeta informático, cuyos conocimientos van desde el desconocimiento de los procedimientos básicos en la manipulación de dispositivos electrónicos, como encender y apagar una computadora o un teléfono inteligente, hasta el dominio de los códigos pragmáticos de los sistemas de aplicaciones en código-objeto desarrollados para diversos fines, que van desde el entretenimiento (como los videojuegos o las redes sociales) hasta la manipulación de los

mismos para realizar sus actividades laborales, como enviar y recibir correos electrónicos, redactar cartas en editores de textos hasta seguir las instrucciones de uno o varios sistemas de aplicaciones y completar formularios predeterminados en varias plataformas³⁸.

En ese caso en el que sus conocimientos se limitan a seguir instrucciones y completar formularios para actividades laborales, realizan un gasto de fuerza de trabajo simple realizando tareas humanas que el programa, convertido en código-objeto, no puede hacer a través de la computadora ni de otro tipo de máquina y por tanto, se consideran tareas humanas.

Este es el caso de cajero de banco, que solo debe registrar la información del cliente en la plantilla y contar el dinero, el cajero de supermercado, que solo debe registrar la información en el sistema y contar el dinero, o el conductor de Uber, cuya plataforma utiliza tecnologías soportadas por Linux y funciona como un IDE de localización espacial y por tanto solo debe enviar las señales georeferenciales cruzadas con las del cliente-usuario para verificar que se realizó dicha tarea (Calo y Rosenblat, 2017: 1631).

Este fenómeno es denominado por Aneesh (2009: 357) control algocrático, el cual es el mecanismo basado en algoritmos, indispensable en los proyectos de desarrollo modulares, horizontales, con múltiples lugares y países, y en los protocolos de la red mismos. Su red de información está compuesta de tres capas: la infraestructura física, la infraestructura lógica y la capa de contenido.

³⁸ Xing (2014: 1186) utiliza la categoría “usabilidad” para referirse a la función inversa del nivel de las habilidades técnicas del usuario: entre más usable un sistema informático, menos habilidades técnicas requiere el usuario para interactuar con él. Los sistemas informáticos en código abierto le dedican menos horas de trabajo que los comerciales en la usabilidad ya que la comunidad supone cierto nivel de habilidad mientras que en los comerciales no.

De esta forma, estos tres tipos de usuarios de los sistemas informáticos se encuentran en posiciones de clase diferenciadas y con experiencias de emancipación, insubordinación y subordinación muy diferentes. Los usuarios más competentes crean valores de uso informáticos para controlar en tiempo real a los menos competentes, y eliminar los posibles episodios de insubordinación –y por tanto de experiencias de clase antagonistas- pero también crean sistemas informáticos para vigilarse a sí mismos y a otros programadores informáticos.

Por ello el término “prosumidor”, como aquel que produce y consume los valores informáticos es impreciso porque generaliza posiciones y experiencias de clase contradictorias entre sí, por lo que debe rechazarse y proponerse más bien en términos de autonomía, antagonismo y subalternidad en relación con la dominación que la clase capitalista ejerce sobre estos distintos tipos de usuarios.

Más aún, solo a partir de más investigación sobre las experiencias de clase de estos usuarios se podrá ofrecer una descripción más precisa al respecto de los procesos de subjetivación política de todos los obreros colectivos $(s, -s)_j$ para determinar cuáles de sus configuraciones tienen un predominio de experiencias emancipatorias, es decir, de autonomía en un sentido marxista, sobre las antagonistas y las subalternas, tal que se pueda concluir si efectivamente la tendencia histórica es hacia su autonomización del poder de mando de la clase capitalista o no.

10. Conclusiones

En primer lugar, se determinó que la materia prima para la producción de sistemas informáticos es la memoria de almacenamiento y procesamiento de datos, y no material metafísico inseparable del cuerpo de la fuerza de trabajo. En este mismo sentido, se encontró que los valores de uso informáticos son como firmas en un

lenguaje formal interpretable por máquinas de diversos tipos (computadoras, teléfonos inteligentes, robots, impresoras, etc) incorporados a las memorias de almacenamiento mediante la memoria de procesamiento a partir de la mecánica impulsada por los dedos de los programadores informáticos, dada una frecuencia o intensidad, durante las horas de trabajo que dure la jornada laboral.

En segundo lugar se encontró que las comunidades de programación libre y abierta si representan un paradigma de organización del proceso de desarrollo de los sistemas informáticos diferente, y en ocasiones antagonista, en relación con la división específicamente capitalista de dicho proceso de trabajo.

Se propuso el término de obrero colectivo, $(s, -s)_i$ y $(s, -s)_j$, siguiendo el concepto de Marx para analizar dichos paradigmas desde la perspectiva de la clase obrera, y se discutió la noción de control y autonomía para el “management” de programación informática, con lo cual se pudo precisar que la categoría de autonomía, o autonomización, presente en los teóricos del capitalismo cognitivo es confusa porque se funda en el economicismo de que el autocontrol por parte de los obreros parciales es más eficiente en términos de minimizar la cantidad de horas de trabajo requeridas para producir un valor de uso informático, por lo que desplaza el control y la vigilancia de los capataces impuestos por la clase capitalista.

Además, se identificaron al menos cuatro cualidades productivas en los obreros colectivos $(s, -s)_i$ que explican su eficiencia en términos de reducción de las horas de trabajo en relación con los obreros colectivos $(s, -s)_j$, directamente bajo el poder de mando capitalista, que son: i) la modularidad, ii) reciclaje de código, iii) la “ley de muchos ojos”, y iv) prácticas comunitarias de colaboración.

Se identificó que existen divisiones internas dentro del obrero colectivo $(s, -s)_i$ vinculadas con los derechos de propiedad con los que protegen su medio de trabajo, los lenguajes de programación, de la sistemática expropiación capitalista de toda forma de riqueza común, colectiva o comunitaria. Aquí, la autonomía de los trabajadores está vinculada con dicha forma de propiedad, ya que es ella la que les permite producir contenidos, obtener remuneraciones y disponer de tiempos de trabajo no determinados por la clase capitalista, por lo que esta división interna puede constituirse en un debilitamiento de su poder de negociación en la medida que las corporaciones capitalistas participan de dichas comunidades y buscan dirigirlas según sus intereses.

Se propuso la paralización del proceso de producción de valores de uso informáticos como el fundamento de su poder de negociación como una relación inversa a su productividad en términos de valores de uso, dada una intensidad del trabajo según ciclos que se repiten durante la jornada laboral y un estado dado de las fuerzas productivas. En este sentido, dicha paralización se describió como episodios de insubordinación de la fuerza de trabajo, en tanto que parte de las clases subalternas, que configura experiencias de clase antagonistas con el poder de mando de la clase capitalista.

Se propuso el concepto de flexibilidad modular para referir la estrategia capitalista seguida dentro de la lucha de clases para disminuir el poder de negociación de los trabajadores informáticos mediante la vigilancia en tiempo real que posibilita el control numérico y el reemplazo modular de los obreros parciales, e incluso del colectivo, originado en la coexistencia y competencia entre obreros colectivos $(s, -s)_i$ y $(s, -s)_j$ dentro de la división internacional del trabajo.

Además, se destacó que el carácter público y abierto de las herramientas de los obreros colectivos $(s, -s)_i$, a pesar de someterlos al control numérico, es una arma en sus manos dentro de la lucha de clases.

Finalmente, se propuso una tipología para los usuarios de valores de uso informáticos basado en su nivel de alfabetización informática, definida a partir del conocimiento de lenguajes de programación, y que permite rechazar el término de “prosumidor” como sujeto del “cognitariado” porque vela sus diferentes posiciones y experiencias de clase, y con ello oculta los potenciales efectos que el control numérico puede implicar al introducirse en otros procesos de trabajo.

Capítulo 4. Análisis de la formación del valor y la valorización de un sistema informático de gestión de aprendizaje mediante un ejemplo práctico

1. Introducción

En este capítulo se expone el análisis del proceso de formación de valor y de valorización para un sistema de gestión de aprendizaje con cinco módulos básicos y para 100 usuarios, con una vida útil de dos años mediante un ejemplo práctico con datos observados no experimentales para aproximar cómo se puede distribuir el tiempo necesario y el plustrabajo en una hora de trabajo a partir de una “plusvalía aparente”(definida a lo largo del capítulo)³⁹.

La discusión se articula alrededor de la noción de trabajo productivo en los servicios, como ha planteado Blondeu (2004), y está atravesada por las dificultades que

³⁹ Este concepto de “plusvalía aparente” se asemeja mucho al de explotación que Joan Robinson (1969: 281) quien la define cuando "el pago del trabajo es menor que el salario apropiado" debido a una negociación desigual entre las fuerzas de empleadores y empleado. Sin embargo, a diferencia de dicha autora, aquí si se considera que la fuerza de negociación es la causa de la explotación y no la ausencia de elasticidad perfecta en la oferta de trabajo o en la demanda por mercancías como propone ella.

supone el problema de la transformación de los valores en precios dentro del debate contemporáneo de la expresión monetaria del tiempo de trabajo (MELT, por sus siglas en inglés), pero se limita a proponer un ejercicio práctico sin entrar en los abordajes robustos que se vienen desarrollando en las últimas décadas.

En este sentido, se propone “una hora de trabajo informático” como unidad de medida de la tasa de explotación, bajo la conjetura de que precios y salarios se constituyen en una estructura que reproduce las relaciones entre clases sociales en su posición inicial, es decir, fuerza de trabajo como clase desposeída y capitalistas como clase propietaria de los medios de producción.

Con este procedimiento, se procura seguir el planteamiento de Marx sobre la “ley general del valor” para estimar en horas de trabajo las relaciones de valor involucradas durante el proceso de trabajo, en la medida que “es posible considerar como diversas fases sucesivas del mismo proceso laboral a los diversos procesos de trabajo particulares, separados en el tiempo y el espacio” (Marx, 1984: 227)

Para la estimación del precio de venta del valor de uso LMS se han consultado tres cotizaciones en línea de plataformas con precios públicos, con la debilidad que esto supone, ya que tanto las empresas como los contratistas de la rama manifiestan obtusas reticencias a brindar dicha información para fines no comerciales, por lo que se sigue la recomendación de Braña y Barreda (2011: 29) de no mentir para obtener datos empíricos o experimentales.

Para la estimación del precio de costo se ha seleccionado Berkeley, California (EE.UU), Ciudad de México, México y Heredia, Costa Rica, tres territorios que se articulan en la división internacional del trabajo en la rama informática como eje

“Norte-Sur” en América , aspecto reconocido tanto por autores de la escuela francesa de la regulación como por teóricos del capitalismo cognitivo.

Para los cálculos de todos los precios que se presentan a continuación se ha tomado el tipo de cambio del peso mexicano para el 26 de julio del 2019, es decir, 19.01 pesos por cada dólar estadounidense (Banxico, 2019).

Para la conversión del colón costarricense a pesos mexicanos, se toma como referencia el tipo de cambio de venta del dólar estadounidense a 575,69 colones por dólar (BCCR, 2019), de tal forma que se tiene un tipo de cambio de 30.28 colones costarricenses por cada peso mexicano.

Finalmente, para realizar los cálculos por hora de trabajo Erneroglu (2011: 42), en su análisis de la dinámica laboral en el neoliberalismo, denomina salario viviente como la magnitud de 2080 días de trabajo (52 x 40) expresadas en valor suficiente para consumir la canasta básica que determina la línea de la pobreza para una familia de 4 miembros.

Sin embargo, en esta investigación se asume que por año se trabajan 50 semanas, es decir, 2000 horas anuales, y que por tanto son 166.67 horas de trabajo al mes. Si se considera que se trabaja solo 8 horas diarias –como fue establecido en la Constitución mexicana de 1917– entonces al mes se trabajan 20.83 días. Esta cantidad es debatible, pero representa un buen referente considerando que existen días feriados diferenciados en cada país bajo estudio, y que, como sostiene Marx al respecto de la duración de la jornada laboral: “Entre derechos iguales decide la fuerza” (1984: 282).

2. Precio y valor en el debate contemporáneo de la expresión monetaria del tiempo de trabajo en la rama informática

La divergencia entre precios y valores de las mercancías es uno de los postulados más polémicos y criticados de la teoría del valor trabajo, tanto en la interpretación clásica como en la crítica de la economía política, y que se encuentra en el centro del debate de la transformación de los valores en precios a partir de la expresión monetaria del tiempo de trabajo (Guerrero, 2000: 149)

Este debate contemporáneo se ha centrado en dos teorías: la Nueva Interpretación (en la versión de Dumenil y Foley); y la Interpretación Temporal y de Sistema Único (representada por Freeman y Carchedi), aunque no son las únicas como detalla Kim (2010: 283).

A este respecto existen muchas consideraciones, por ejemplo Moseley (2016: 16) sostiene que estas dos interpretaciones carecen de una rigurosa teoría del plusvalor, pero no conviene a esta exposición centrarse en este ampuloso debate sobre ellas, sino proponer un procedimiento práctico para utilizar precios de mercado observados para realizar estimaciones semejantes las que hace Marx en la sección tercera de El Capital dedicado a la “Producción del Plusvalor Absoluto” bajo la noción intuitiva de “plusvalía aparente” como ese excedente entre el precio de venta y el precio de costo.

A lo largo de casi tres décadas se han publicadas diversos trabajos de carácter empírico con abordajes robustos a partir de matrices insumo producto, desde la propuesta pionera de Ramos Martínez (1995: 25) hasta trabajos como la evidencia empírica sobre China, Japón y Korea en 2009 expuesta por Tsoulfidisy Paitaridis (2016: 18) a favor de la interpretación clásica por sobre la Interpretación Temporal y de Sistema Único mediante la medición de las desviaciones de los precios de producción respecto de los precios observados, según la desviación de las intensidades de capital en cada industria en relación con los promedios nacionales.

En este mismo enfoque empírico se encuentra el planteamiento de Fuchs (2017 : 72) quién retoma el debate de la MELT en la creación de valor en la economía informacional para la economía política marxista. Este autor analiza el caso de la economía de información de Alemania a partir del sistema simple de interpretación temporal (TSSI por su nombre en inglés) como una versión de la teoría marxista del valor fundada alrededor de los planteamientos Andrew Kliman y Alan Freeman.

Según Fuchs (2017: 73) los sistemas informáticos comerciales, las mercancías cognitivas, no solo son producidos y copiados, sino que además requieren actualizaciones y formas de trabajo de apoyo y mantenimiento. En la división internacional del trabajo digital existen varias formas de organización del trabajo, incluidos los programadores informáticos, quienes son altamente explotados en términos de demanda de su tiempo. Por ello, sostiene que la ley del valor digital crea nuevas formas de explotación, así como contradicciones que permiten nuevas esferas no comerciales de producción alternativa y solidaridad.

Además, apunta que el problema de cómo medir la productividad de Microsoft se puede abordar considerando que esta empresa libera actualizaciones de su sistema operativo, de tal forma que la productividad se puede medir por el número de actualizaciones, versiones o diferentes tipos del sistema informático. De esta forma, concluye que sí es posible dar con una medición estandarizada de la productividad (2017: 83).

Pese a la pertinencia de estas consideraciones, su abordaje empírico (con datos de la industria alemana para el periodo 2001-2011) es diferente al aquí propuesto, ya que este autor rechaza los análisis sectoriales para el caso de la tecnología informática, porque como ha sostenido en publicaciones previas, la misma solo es posible por la

existencia de un amplio trabajo explotado, como los trabajadores esclavizados en África para extraer los minerales requeridos en la producción de los dispositivos electrónicos (“hardware”), así como de los altamente explotados trabajadores industriales en China, y otros países que realizan el ensamblaje de partes, en centros de trabajo precarios y con condiciones de riesgo en los países en desarrollo (Fuchs y Sevignani, 2013: 284)

Otros autores, como Starosta (2012) y Carchedi (2014), ofrecen un abordaje que puede considerarse sectorial, pero omiten estas consideraciones empíricas de cómo manipular los precios para aproximar el valor. Pese a esto, la comprensión de ambos sobre el capital constante en el sector informático es más conveniente a la disertación aquí ensayada, a pesar de que se enfocan en la industria de los videojuegos, el primero, en la formación del precio de las mercancías cognitivas y, el segundo, en el valor de uso mental dentro de la lucha de clases cognitiva.

De esta forma, para ofrecer al menos un ejercicio práctico sobre el problema de que no se pueden tomar los precios de los valores de uso informáticos como si fueran sus valores, se sigue la recomendación de Althusser (2004: 22) en el sentido de identificar cómo es que el dinero adelantado por los capitalistas, de la rama informática en este caso, se convierte en capital, es decir, aproximar las relaciones de explotación capitalista a partir de la relación entre el precio de venta de los valores de uso informáticos y los salarios nominales de los programadores, es decir, estimar una “plusvalía aparente”.

Con esta consideración se busca evidenciar la relación directa que enfrentan determinados obreros de la rama informática y los representantes de la clase capitalista que los contratan, y no la relación global de explotación que se obtiene con los cálculos

matriciales sobre la expresión monetaria del tiempo de trabajo (MELT) a partir de los precios de producción.

Esta relación directa entre explotados y explotadores en la rama informática se estructura a partir de las condiciones dadas de la economía mundial, es decir, salarios y precios que no son fijados individualmente sino que están acotados por los límites que la acumulación de capital les impone a través del plusvalor relativo, lo cual constituye evidencia en contra las teorías del capitalismo cognitivo aquí discutidas.

Por ello, se asume la interpretación de Valle Baeza (1996: 64) sobre la relación entre salario y productividad, en tanto que la segunda debe definirse “como el recíproco del valor medio de una canasta de mercancías”, de lo que se sigue que el salario que reciben los programadores informáticos debe corresponder con una canasta de mercancías destinadas para su reproducción como facción de clase y, por tanto, proporcionalmente equiparable a los salarios nominales que reciben otras facciones de clase dedicadas a otros procesos de trabajo, como el ensamblaje de los dispositivos electrónicos, que son clasificados como trabajo simple.

Esta consideración se enfrenta a la noción de productividad de Fuchs antes planteada, porque postula el valor de la fuerza de trabajo de una rama, como la informática, a partir del plusvalor relativo, es decir, como una relación social fundamentada en las fuerzas productivas en cada periodo histórico, y no como una relación individual entre la cantidad de veces que se realiza la tarea s en una hora de trabajo y el salario percibido por cada obrero parcial.

De esta forma, se propone la hora de trabajo como unidad de medida de la estructura relacional a través de la cual el precio de venta y el salario reproducen las condiciones de explotación capitalista en cada ciclo del capital productivo para una

determinada empresa. Esto no resuelve las dificultades vinculadas con la expresión monetaria del tiempo de trabajo en el problema de la transformación de los valores en precios, pero favorece la discusión con los teóricos del capitalismo cognitivo, objetivo central de esta investigación.

Además, se acoge el planteamiento de Sandoval (2007: 89) de que hay una carencia conceptual en la definición de la composición técnica del capital porque no se pueden mezclar las herramientas, ya que como el mismo Marx estableció, lo único común es el trabajo.

Finalmente, la dificultad que supone el problema de la transformación de los valores en precios, y el enfoque contemporáneo a partir de la expresión monetaria de una hora de trabajo, se presenta también en el hecho de que los precios de los factores productivos, estimados en este apartado, no pueden asumirse como si fueran sus valores, y que Rodríguez (2008: 426) plantea bajo la pregunta de si el capital constante es el precio devengado en el periodo anterior ($t-1$), el vigente durante el proceso de producción y circulación (t), o el de reposición ($t+1$)

Sobre este cuestionamiento, Herrera (2017: 58) propone un método basado en las horas de trabajo para calcular el trabajo pretérito para los sectores de la reproducción social a partir de los esquemas de reproducción de Marx y sus supuestos, por lo que es un abordaje teórico, semejante a la inducción hacia atrás en la teoría de juegos (Yoav y Leyton: 2008: 38), cuya aplicación empírica exigiría medir las horas de trabajo en los procesos de trabajo previos a la programación informática según la propuesta de Fuchs y Seignani (2013), como la manufactura de los dispositivos electrónicos y las industrias extractivas, procedimiento fuera del alcance de esta investigación.

Para atender el cuestionamiento inicial, se retoma el procedimiento de Marx (1984: 241) de separar el producto del valor, capital variable más plusvalor, del valor del producto, que incluye el capital constante, y utilizar el primero como magnitud de valor añadido en el proceso de trabajo, es decir, apelar a la dualidad del trabajo mismo en el sentido de que la creación de valor y la transferencia de valor de los medios de producción al valor de uso informático suceden de manera simultánea.

En consecuencia, la argumentación que sigue a continuación se limita a las relaciones en términos de valor, es decir, a la tasa de plusvalor, y a la composición de valor como parte de la composición orgánica de capital, para el caso de un sistema informático de gestión de aprendizaje (LMS) de acuerdo con una cantidad dada de módulos y de usuarios, y no las condiciones generales de explotación de la rama informática de acuerdo con sus precios de producción.

3. Precio de venta de los valores de uso informáticos

El problema de la transformación de los valores en precios, en el debate contemporáneo, se presenta en toda su dimensión al considerar el precio de venta de las mercancías, cognitivas en este caso, ya que la evidencia empírica en la que se soporta la conjetura de la no vigencia de la ley de valor de los teóricos del capitalismo cognitivo es que dichos precios no están determinados por la cantidad de horas de trabajo requeridos para su producción, lo cual se origina, según Moulrier-Boutang (2012: 178), en que la información digitalizada tiene un costo marginal que tiende a cero al ser fácilmente copiable y reproducible.

Para responder a esto, y de acuerdo con el enfoque de introducir el análisis de clase desde la perspectiva de la clase obrera, el procedimiento ensayado en esta tesis es que los capitalistas fijan el precio de venta de las mercancías cognitivas a partir de la

suma de los costos de producción y adicionándole una ganancia, que para el 2014 rondaba el 20% como ganancia media para las GAFAM (Google, Apple, Facebook, Amazon y Microsoft) (Gautier y Lamesch, 2020:13), según dos estrategias:

- i) Vendérsela a muchos compradores bajo la apariencia de una licencia de uso, libre o privativa, en cuyo caso el precio adopta de monopólico basado en los derechos de propiedad intelectual;
- ii) Vender su mercancía a un único cliente, es decir, como un desarrollo a la medida.

En el caso de la primera estrategia de las empresas capitalistas en la fijación del precio de venta, el sistema informático es ofrecido como código objeto y puede ser copiado muchas veces hasta establecerse una dinámica de acumulación capitalista, aparentemente rentística, distribuyendo el precio de venta total entre varios muchos compradores, tal y como hacen corporaciones multinacionales como Microsoft con su LMS365, y la mayor parte de los sistemas privativos, como Blackboard Learn.

En la segunda estrategia el código-objeto es entregado directa y exclusivamente para el consumo del cliente con la restricción que la empresa vendedora no puede vender ninguna réplica de dicho sistema informático, por lo que el precio de venta total corresponde con el precio pagado por ese único cliente.

De esta forma, el que un sistema sea empaquetado, o desarrollado a la medida, no modifica el proceso de valorización sino la cantidad de plusvalor apropiado en la circulación por cada empresa.

A este respecto, y con el caso específico de los videojuegos, Starosta (2012: 382) ha argumentado que los sistemas informáticos ingresan como medio de producción en subsecuentes procesos productivos y por tanto su valor se consume lentamente, por lo

que no tiende a cero ni desaparecen de los medios de producción, sino que es un valor que está contenido en el prototipo que es copiado y se va transfiriendo poco a poco a sus réplicas, como en todos los casos de capital constante fijo.

Para el caso específico de los sistemas de gestión de aprendizaje LMS, tanto en el modelo de fijación de precios basado en licencias como en el desarrollo a la medida, la cantidad de módulos incluidos en el prototipo, así como la cantidad y el tipo de usuarios esperados es fundamental para las tarifas cobradas.

En primer lugar, la cantidad de horas de trabajo requeridos para producir el prototipo que será copiado varía de acuerdo con la cantidad de módulos, tal que entre más módulos, más horas de trabajo.

El aumento en la cantidad de horas de acuerdo con la cantidad de módulos se explica en que la arquitectura y el diseño deberán ser más complejos en relación con el manejo de la base de datos y la gestión de las interfaces entre módulos, lo cual se dificulta aún más conforme aumenta la cantidad y el tipo de usuarios esperados, ya que entre más peticiones de respuesta generen dichos usuarios, suceden demoras en el sistema mismo así como vulnerabilidades en términos de seguridad de la información y problemas en el tráfico y manejo de datos como en el ejemplo comentado en el capítulo anterior al respecto de las redes móviles (Lu et. al, 2018).

Por ello, la arquitectura y el diseño de un sistema de gestión de aprendizaje desarrollado para ser vendido en copias a varios clientes y uno a la medida de un cliente requieren cantidades de horas de trabajo diferentes en todas las tareas, y por tanto ocasionará que sus precios de venta sean, también, diferentes.

En consecuencia, al estar orientados para soportar cantidades y tipos de usuario de manera masiva, el producir arquitecturas y diseños apropiados para ello implica

adelantos de capital mucho mayores que los desarrollados a la medida, lo que a su vez implica más horas de trabajo en codificación, pruebas e integración. De ahí que las licencias de uso busquen asegurar la transferencia del valor total del prototipo en las copias vendidas, como argumenta Starosta, lo cual le asegura a la clase capitalista el retorno del capital adelantado.

En segundo lugar, el uso de tecnologías privativas o libres puede implicar variaciones en la cantidad de horas de trabajo para producir un sistema de gestión de aprendizaje, ya que desarrollándolo en código abierto permitirá que los programadores pueden reciclar el código, así como tipos de pruebas y soluciones a problemas de integración e instalación de otras plataformas en código abierto, disminuyendo la cantidad de horas de trabajo dedicadas a dichas tareas sin afectar la calidad, lo que conduce a que su precio de venta pueda ser inferior a uno producido con tecnologías privativas.

Debido a ello, las licencias privativas de los sistemas de gestión de aprendizaje buscan impedir que otras empresas reciclen su código. Por eso dichas licencias privadas constituyen una barrera de entrada técnica para la competencia, es decir, como fuente de plusvalor extraordinario, lo que favorece que se constituyan en monopolios.

Pero de eso no se sigue que sean precios puramente rentísticos, ya que las tarifas cobradas por las licencias privadas responden a mayores cantidad de horas de trabajo, como se argumentó anteriormente, pero además deben pagarse los servidores donde se aloja el código y la base de datos, así como la instalación y el mantenimiento, tareas que consumen más horas de trabajo entre menor sea el nivel de alfabetización informática de sus usuarios.

Esto se origina en que toda la información producida por los usuarios, estimados en cantidades masivas, requieren enormes capacidades de almacenamiento y procesamiento, por lo que las tarifas cobradas por estas licencias también deben ser suficientes para asumir esos costos monetarios.

Así, esta relación positiva entre requerimientos en términos de módulos y usuario de un sistema de gestión de aprendizaje con la cantidad de horas de trabajo requeridas para producirlos, constituye la evidencia de que es la cantidad de horas de trabajo en programación informática las que se dedican para desarrollarlos, y no una fuerza abstracta e “inmaterial” como el intelecto general, lo que constituye el líquido amniótico del proceso de formación de valor y valorización de los sistemas informáticos y, por tanto, en la formación del precio de venta.

Pese a las potenciales divergencias cuantitativas entre valores y precios, el precio de venta debe ser competitivo y esto quiere decir que debe tener al menos la cantidad de horas de trabajo socialmente necesarias, ya sea que sigan una estrategia de licenciamiento privativo de sus sistemas informáticos o que sigan una estrategia de desarrollo a la medida.

En este sentido de lo que trata el problema aquí abordado es sobre cómo las tareas de programación informática – en tanto que servicios- se convierten en trabajo productivo, según Blondeau (2004: 32-33)

Por ello el énfasis se establece en que la estructura de precios y salarios, basada en las horas de trabajo como reactivo limitante, es una condición necesaria para la acumulación de capital en la medida que la clase capitalista tiene un poder de negociación, limitado por el poder de negociación de la clase trabajadora, para fijar los precios y los salarios, pero que les asegura a los primeros que para todo valor de uso

producido, su precio de venta sea estrictamente mayor que su costo, del cual los salarios constituyen el principal componente para las mercancías cognitivas.

Si bien este postulado parece axiomático, esta investigación se centra en el análisis de la explotación, y la misma muestra que la relación entre precios y salarios tiene una estructura tal que reproduce las clases en su posición inicial, es decir, los capitalistas como clase propietaria y la fuerza de trabajo como clase desposeída, tal y como argumenta Marx en su análisis de la reproducción (1984: 711).

De esta forma, el interés en relación con el precio de venta en esta tesis no consiste en si es igual al valor o no, o su relación con el precio de producción, sino en cómo dado un precio en el mercado la clase capitalista organiza el proceso de trabajo a partir del consumo productivo de una determinada cantidad de horas de trabajo, tal que su precio de costo sea menor y por tanto se asegure una ganancia, un excedente económico en términos de Baran y Sweezy 1980:13) y de esa forma, un plusvalor aparente.

En este sentido, lo que se propone es utilizar los precios de venta y costo observados para estimar cuántos minutos en una hora de trabajo dedica el programador informático en la formación del valor destinado a reproducir su fuerza de trabajo y cuántos minutos dedica a la valorización en beneficio de la clase capitalista.

Por consiguiente, desde la teoría del valor trabajo se sostiene que la formación del precio se realiza a partir de la cantidad de horas añadidas al valor de los medios de producción, tal que dichas estrategias de fijación del precio de venta tienen a las horas de trabajo consumidas productivamente en la producción de la mercancía cognitiva como su fundamento, aunque a la clase capitalista se le presente como estrategias comerciales. Como dice Marx: “No lo saben, pero lo hacen” (1984:90).

En lo que sigue de la exposición, se seguirá con la estrategia de fijación de precios cuando se elaboran sistemas informáticos a la medida.

4. El precio de venta de los sistemas de gestión del aprendizaje (LMS)

Para proceder con el análisis de la formación de valor y la valorización en los sistemas de gestión de aprendizaje aquí propuesto es necesario determinar la relación entre la cantidad de módulos y los precios de venta para el caso específico de los sistemas de gestión de aprendizaje, tal que a partir de las especificaciones técnicas de los requerimientos básicos de una de estas mercancías cognitivas se pueda estimar la cantidad de horas necesarias para producirlo, asunto que se abordará en el siguiente apartado.

Desde el punto de vista pedagógico Ouadoud , Yassin Chkouri y Nejjari (2018: 31) destacan el enfoque modular⁴⁰ en la arquitectura de los sistemas de gestión de aprendizaje (LSM), dentro de los paradigmas de enseñanza (pedagogía tradicional, conductismo, cognitivismo y constructivismo social), porque favorece su adaptación a diferentes contextos debido a la estructura granular de sus bases de datos, permitiendo una completa transversalidad entre usuarios (docentes, estudiantes y administradores) y la personalización deseada.

Desde el punto de vista informático, Kasim y Khalid (2016: 56) sostienen que los LMS utilizan un modelo de referencia de objetos de contenido compatible (SCORM), el cual controla cómo el contenido de aprendizaje en línea y los sistemas de gestión del aprendizaje se comunican entre sí. SCORM orienta a los programadores sobre cómo escribir su código tal que puedan ser utilizados con otros programas de aprendizaje electrónico (e-learning).

⁴⁰ Aunque no sea el único enfoque de programación dentro de los paradigmas más destacados según Lula et. al. (2014: 51), el imperativo y el orientado a objetos, la modularidad se utiliza para diversos fines.

Para estos autores, los LMS se componen solo de tres tipos de componentes: i) las herramientas para el aprendizaje de habilidades (crear actividades y herramientas de aprendizaje como presentaciones, evaluaciones, asignaciones), ii) herramientas de comunicación para permitir la interacción entre maestros y estudiantes, iii) herramientas de productividad para sistemas de gestión, como subir documentos, registros de actividades de estudiantes, encuestas (Kasim y Khalid , 2016: 56)

Adicionalmente, distinguen entre dos tipos de LMS, los de código abierto como Moodle, Sakai, ATutor, Claroline, MyGuru2 y MyLMS, y los comerciales como Blackboard, SuccessFactors, SumTotal, Litmos, Angle Learning, Geo Learning, Cornestone y Connect Edu (Kasim y Khalid , 2016: 57)

Desde el punto de vista institucional, Sánchez Rodríguez (2009: 219) sostiene que un sistema LMS debe contar con al menos un módulo para cada una de las siguientes cinco funciones que destaca 1) herramientas de distribución de contenidos; 2) herramientas de comunicación y colaboración sincrónica y asincrónica; 3) herramientas de seguimiento y evaluación; 4) herramientas de administración y asignación de permisos; 5) herramientas complementarias.

Para este autor, los precios de los LMS se fijan con base en las “licencias” que suelen ser permisos de uso, de acuerdo con la cantidad total de estudiantes, pero solo permiten instalarlas en un servidor (asociado a un IP), lo que dificulta las actualizaciones, o tener copias de seguridad. Además, afirma que muchos LMS empezaron con licencias libres, pero cambiaron a privadas cuando lograron penetrar el

sector y cuyas tarifas, oscilaban de los 6000 dólares estadounidenses a los 68000 dólares anuales, incluido el soporte (Sánchez Rodríguez, 2009: 221)⁴¹

Poulova, Simonova y Manenova (2015: 1304-1307) estiman que para Blackboard Lean el costo por usuario es 10 dólares estadounidenses por año, aunque reconoce que esto varía según la cantidad de usuarios y las funciones solicitadas. Más aún, destacan que no opera bien en todos los sistemas operativos, tal parece que ha sido optimizado solo para Microsoft NT, además de que su precio es muy elevado, entre 200 mil y 400 mil dólares estadounidenses anuales dependiendo de las funciones utilizadas⁴².

Los precios de venta vigentes en julio del 2019 para los sistemas de gestión de aprendizaje, de acuerdo con las funciones ofrecidas por los proveedores, se resumen en la siguiente tabla:

Tabla 1. Precios según la cantidad de funciones disponibles para los sistemas de gestión de aprendizaje (LMS), al tipo de cambio peso/dólar del 15 de julio del 2019

Empresa	Funcionalidad ofrecidas	Precio
iSpring Solutions	Gestión de contenidos	Suscripción de acuerdo con cantidad de usuarios. 6957.66 pesos mensuales por 100 usuarios.
	Gestión de usuarios	
	Gestión de prácticas	
	Reportes	
	Personalización del sistema	

⁴¹ El precio no se ha convertido a pesos mexicanos para evitar confusiones con las estimaciones realizadas a lo largo de la investigación y la de otros autores. El 15 de julio del 2009, el tipo de cambio peso/dólar era de 13.65 (Banco de México, 2019), por lo que dichas tarifas equivalían a 81900 pesos y 928200 pesos, respectivamente.

⁴² El tipo de cambio peso/dólar al 15 de julio del 2015 era de 15.75 (Banco de México, 2019), por lo que dichas tarifas eran de 157.5 pesos por usuario por año, mientras que el precio del sistema completo iba de los 3150000 pesos hasta los 6300000 pesos por la compra de funcionalidades a la medida.

	Portal del aprendiz	
	Integración	
	Opción de personalización	
Talent LMS	Gestión de cursos	Suscripción de acuerdo con cantidad de usuarios. 2452.29 pesos mensuales por 100 usuarios.
	Entrega de certificados	
	Personalización para empresas	
	Personalización de marca	
	Sistema estandarizado	
Accord LMS	Solución de problemas en la nube	Suscripción de acuerdo con cantidad de usuarios, 3326.75 pesos mensuales por 200 usuarios
	Almacenamiento ilimitado y banda ancha	
	Características básicas y para empresa	

Fuente: Elaboración propia. Datos iSpring Solutions (2019), Talent LMS (2019) y Accord LMS (2019)

Como se muestra en la Tabla 1, los precios varían de acuerdo con la cantidad de usuarios y con las funcionalidades disponibles, pero debe aclararse que corresponden con licencias de uso, es decir, son sistemas estandarizados destinados al uso masivo de usuarios. Por ello, los precios que se consignan son por suscripción y no a la medida, como se propuso proceder en esta investigación.

Pese a esto, estos datos sirven como referente para los precios de venta de los sistemas de gestión de aprendizaje a la medida (LMS) ya que el problema al que se

enfrenta el comprador es utilizar estos precios de suscripción o contratar un desarrollo a la medida.

Los LMS son valores de usos estandarizados en el sentido de que sus componentes internos tienden a ser los mismos e involucran módulos relacionados con funciones muy semejantes entre las ofertas disponibles, como la gestión de cursos y contenidos, almacenamiento, personalización o gestión de usuarios, por lo que la oferta de desarrollo a la medida no puede sobrepasar estos precios porque se quedaría sin compradores.

Así, el problema al que se enfrenta el capitalista de la rama informática es determinar si su precio de costo es suficientemente bajo como para poder producir un sistema a la medida a ese precio de venta o menor.

Desde esta perspectiva, y considerando que no solo las máquinas computadoras enfrentan un periodo aproximado de depreciación moral de dos años (como se argumenta en el apartado de costos), sino que los sistemas informáticos mismos enfrentan periodos de esa magnitud o menores (por ejemplo los sistemas operativos de los teléfonos tienen actualizaciones anuales), entonces puede sostenerse que el precio de venta de un LMS a la medida, con una vida útil de dos años, será de 15 8243.84 pesos si dispone de características semejantes al de iSpring Solutions, de 58 854.96 pesos para uno como el ofrecido por Talent LMS y de 79 842 pesos para uno con la cantidad de funciones y usuarios como el de Accord LMS.

De esta forma, en una negociación entre dos capitalistas, uno del sector informático y otro del sector educativo, para el desarrollo de un LMS a la medida para 100 usuarios, o máximo 200 usuarios, oscilará entre los 58 854.96 pesos y los 15 8243.84 pesos.

Si se comparan estos precios en dólares estadounidenses, para eliminar el efecto de la depreciación del peso mexicano, con los aportados por las fuentes anteriormente destacadas, el de iSpring Solutions se encuentra por encima del margen inferior del rango obtenido por Sánchez Rodríguez (2009) aunque muy lejos del obtenido por Poulouva, Simonova y Manenova (2015) para el caso del Blackboard Learn, el cual es destacado como uno de los más onerosos del mercado producido con un obrero colectivo $(s, -s)_j$.

Más aún, como el mismo Sánchez Rodríguez (2009) destaca, algunos de los sistemas que consultó fueron desarrollados en código abierto, es decir, mediante la fuerza de trabajo de un obrero colectivo $(s, -s)_i$, y solo después la empresa cambió la licencia, por lo que no es inverosímil que un capitalista pueda producir un sistema de gestión de aprendizaje con cinco módulos para competir con el de iSpring Solutions en términos de calidad pero no de precio.

De hecho, el uso de tecnologías libres y abiertas reduce el costo monetario de producción en relación con el uso de los sistemas privativos, tanto en el uso de lenguajes de programación sin costo monetario, como porque favorecen que los sistemas empaquetados se distribuyan sin precio y con licencias GPL, basadas en las cuatro libertades promovidas por la Fundación GNU, o con licencias Apache o BSD, o incluso que haya cambios de licencia durante el ciclo de vida de un programa informático, como parte de las estrategias de las empresas capitalistas para apropiarse del código abierto y la comunidad que los soporta, como en los casos destacados anteriormente por Sánchez Rodríguez (2009) de algunos LMS.

En este sentido, Vendome et. al. (2016: 1553) estudian los patrones de cambio de estas licencias a partir de los códigos alojados en Github y encuentran que el patrón

de migración más común es de las GPL 3-0+ a las Apache 2.0, es decir, hay una tendencia a aumentar las licencias en código abierto pero comerciales, y por tanto, compatibles con el modo de producción capitalista.

Por ejemplo, Moodle y Claroline, sistemas de gestión del aprendizaje en código abierto, se distribuyen como sistemas empaquetados con módulos genéricos pero sin costo monetario, es decir, cualquier programador con los conocimientos suficientes, puede usar dicho sistema y compartirlo, e incluso modificarlo a conveniencia, por ejemplo para elaborar un LMS con 5 módulos para 100 usuarios mediante el reciclaje de código, pero sin copiar textualmente las líneas protegidas bajo licencia libre.

De esta forma, existe la opción de contratar programadores independientes, o incluso una empresa capitalista que utilice tecnologías libres y abiertas, para que desarrolle los módulos con las especificaciones deseadas, en cuyo caso, la cantidad de horas de trabajo se revelan determinantes para la formación del precio de venta, y en este sentido, como sustancia de su valor.

Sin embargo, es necesario determinar la cantidad de horas de programación informática requeridas para producir un LMS con al menos los cinco módulos destacados por Sánchez Rodríguez (2009) para poder determinar su costo de producción.

5. Cantidad de horas de trabajo en programación informática requeridos para producir un sistema de gestión de aprendizaje para 100 usuarios y cinco módulos

Para continuar con el análisis de la formación del valor y de la valorización es necesario determinar cuántas horas de trabajo se requieren para producir estos sistemas

de gestión de aprendizaje (LMS), y para ello es indispensable retomar las características específicas del proceso de trabajo de programación informática.

En primer lugar, la capacidad de abstracción requerida para la arquitectura y el diseño no necesariamente se obtienen de dominar un lenguaje de programación, aunque saber leerlo y escribirlo si es indispensable para obtener una arquitectura óptima y un diseño eficiente. Pese a esto, para un sistema de gestión de aprendizaje de cinco módulos básico y con una capacidad para 100 usuarios, como máximo 200 no se requiere de un investigador especialista en computación o un arquitecto en redes computacionales, sino un desarrollador de sistemas informáticos, el cual a su vez podría desempeñar tareas como las pruebas o la integración, que requieren una visión global del sistema y la interacción entre módulos e interfaces.

En segundo lugar, el obrero colectivo ($s, -s$) requerido debe estar compuesto por al menos dos programadores, uno que se encargue de la codificación y otro de las pruebas, ya que de lo contrario aumenta la probabilidad de errores invisibles para el codificador desde su perspectiva. Estas dos tareas, codificación y pruebas, son las que más horas de trabajo requieren, ya que durante las mismas se escribe el código fuente, el cual debe ser probado para evitar los diversos tipos de errores (“bugs”) que podrían afectar el funcionamiento estructural del sistema de gestión de aprendizaje. Por el carácter modular del sistema en cuestión, cada uno de los módulos podría ser asignado a un codificador diferente y desarrollarse en paralelo. Esto reduciría el plazo de entrega pero no disminuiría la cantidad de horas requeridas para cada módulo.

El reciclaje de código podría reducir la cantidad de horas de trabajo si el LMS en cuestión no tuviera fines comerciales, ya que las licencias libres prohíben el plagio con dicho fin. De esta forma, para un contratista que utilice tecnologías libres y

abiertas, dicha práctica favorecerá la calidad de su valor de uso informático, ya que los codificadores podrán usar de referencia dichas líneas de código, pero no podrán plagiarlo completamente para reducir la cantidad de horas de trabajo.

En tercer lugar, el proceso de integración del código de un sistema con estas características requiere de conocimientos de los entornos donde será instalado, de los protocolos de comunicación con otros sistemas, como correo electrónico, así como el manejo de las interfaces de otros sistemas, como navegadores y sistemas operativos, los cuales suelen estar programados en otros lenguajes de programación. Esta tarea podría ser realizada por un tercer obrero parcial con conocimientos en sistemas computacionales y desarrollo web, o ser realizado entre el codificador y el encargado de las pruebas, tomando en consideración la polivalencia exigida al “cognitariado” y que el sistema en cuestión es pequeño.

Finalmente, la instalación y el mantenimiento requieren de dominio técnico de sistemas operativos y administración de servidores, y muy poco de los lenguajes en los que han sido programados los módulos. En este sentido, se requiere de un tipo de fuerza de trabajo menos especializada y cuyo aporte en horas de trabajo es mucho menor.

De esta forma, y considerando que el mantenimiento se cobra por aparte en los desarrollos a la medida, los procesos de programación informática en el desarrollo de un módulo para LMS de complejidad media-baja, es decir, para un rango 100 a 200 usuarios y alguna compatibilidad con otros LMS, la cantidad de horas de trabajo se resumen en la siguiente tabla:

Tabla 2. Estimación de horas requeridas para desarrollar un módulo de LMS

Arquitectura	20 horas
--------------	----------

Diseño	40 horas
Codificación	160 horas
Pruebas	160 horas
Integración	40 horas
Instalación	4 horas
Total	424 horas

Fuente: elaboración propia con base en consultas a programadores informáticos en Solvo (2019)

Como se muestra en la Tabla 2, un módulo para LMS podría requerir 424 horas de trabajo, lo que para 5 módulos da un estimado de 2120 horas. Esta cantidad, si bien puede variar dependiendo de la pericia y experiencia de los programadores contratados es verosímil y sirve como referencia para cotizaciones de este tipo de sistemas informáticos.

En relación con lo anterior, cabe destacar que existe un marcado celo profesional entre los programadores para revelar sus tiempos exactos, lo cual es comprensible, dado que entienden a la perfección que dicha información es estratégica, no solo en términos competitivos, sino además en relación con la automatización de su proceso de trabajo.

De hecho, estas dificultades para estimar la cantidad de horas requeridas para el desarrollo de un módulo de un sistema informático en general, y para LMS en particular, no son menores y se encuentran en el centro del debate entre los que defienden que dicho trabajo es propio de artesanos, como Lessing (2004: 14) y Raymond (2001: 82), y los que defienden que su ontología es propia de una ingeniería, como Northover et. al. (2008: 108) y Sommerville (2011: 28).

Más aún, constituye el argumento de Cafentzis (2016: 37) contra Vercellone y su comparación entre los programadores informáticos y los trabajadores de la externalización (“putting-outsystem”) de los siglos XVI y XVIII, ya que el mismo era un mismo sistema de artesanado en el cual no había supervisión, es decir, la disputa era por el control del proceso de trabajo y las horas de trabajo, pero se terminó cuando las herramientas de trabajo se hicieron propiedad de la clase capitalista, lo que supuso la desaparición de dicha clase de artesanos.

Para los autores que defienden que la programación informática es una ingeniería, todas estas dificultades sobre el cálculo de horas de trabajo necesarias para desarrollar un sistema informático, el control del proceso y la supervisión, y que podrían situarla como una “tecnología artesanal”, responden al despliegue histórico de los paradigmas de la disciplina de acuerdo con los retos emergentes hasta alcanzar el estatus de ingeniería como tal, aunque no lo haya logrado todavía (Northover et. al., 2008: 108).

Sommerville (2011: 12) reconoce que en 50 años de historia de métodos para la programación informática no se ha constituido ningún método de desarrollo que estandarice costos, horarios, conflictos de dependencias, para todos los tipos de sistemas informáticos. Sin embargo, defiende que existen cuatro fundamentos en la ingeniería informática que aplican para todos los tipos de sistemas: i) utilizar un plan del proceso de trabajo con plazos; ii) el sistema no debe desperdiciar recursos de ningún tipo; iii) administrar las expectativas de los usuarios para ajustarse al presupuesto y calendario; iv) siempre se debe reciclar código en vez de escribir nuevo.

Esta concepción de la programación informática como una ingeniería supone, además, la incorporación de dos tareas s más de las consideradas en la tabla 2: a)

gestión del proyecto de desarrollo informático o management; b) la comercialización, por lo que debe cuestionarse si esas horas de trabajo forman parte del trabajo socialmente necesario, es decir, si son trabajo productivo o no como ha planteado Blondeu (2004).

Según Braverman (1998: 287) las tareas de realización y apropiación de plusvalor por el capital son trabajo improductivo, porque como sostiene Wright (1996: 712) , los trabajadores que las desempeñan se encuentran en una posición “contradictoria de clase” al agenciar los intereses de la clase capitalista, a pesar de pertenecer a la clase trabajadora.

Desde esta perspectiva, tanto la tarea de gestión del proyecto de desarrollo informático, como la de comercialización de las copias bajo licencia son trabajo improductivo.

El carácter improductivo de la tarea de gestión del proyecto de desarrollo informático, en tanto que consiste en controlar los avances de los programadores y coordinar que las interdependencias entre la manipulación de código no afecte los plazos de entrega, queda evidenciado por la existencia misma de la programación libre y abierta con sus modelos de gestión basados en la autoridad técnica, como Linus Torvald en Linux o Richard Stallman con GNU, y mediante los métodos ensamblarios de red que caracterizan a Apache y a Mozilla, ya que muestran que dicha tarea constituye una imposición de vigilancia, por interpósita persona, de la clase capitalista sobre el obrero colectivo $(s, -s)_j$.

Una argumentación que permite mostrar el carácter improductivo de esta tarea la ofrece Fernández (2013: 77-78) con “el juego de la inspección”, donde un capataz puede supervisar o no a un empleado, mientras que éste puede trabajar o realizar otras

actividades, de tal forma que si el salario es bajo, el capataz no tiene incentivos para supervisar y el trabajador los tiene para realizar otras actividades, pero si dicha remuneración aumenta se reducen las probabilidades de que el empleado no trabaje porque “un salario más alto actúa como un poderoso instrumento de disuasión de actividades irregulares”.

En este caso, trabajadores que introyecten la autoridad capitalista como autocontrol, y por tanto operen como trabajadores independientes, pero no autonomizados del poder de mando capitalista, constituye un fin perseguido por la clase propietaria mediante salarios más altos que el salario mínimo, no porque objetiven más valor, sino porque elimina la tarea s de vigilancia de los capataces dentro de los obreros colectivos $(s, -s)_i$ y con ello, el costo asociado a su remuneración, y cuya magnitud monetaria puede ser redistribuida entre los obreros parciales bajo la apariencia de salarios más altos, y justificado ideológicamente con relatos, como el de la sociología weberiana o del management, articulados alrededor de las diferencias en los niveles educativos y la cantidad de responsabilidades dentro de la jornada laboral.

En relación con la tarea de comercialización, a pesar de que constituye un trabajo indispensable para el capital comercial, es improductiva, en este caso por ser exclusivamente una defensa de la imposición de los derechos de propiedad privada de la clase capitalista sobre el resto de la sociedad, ya que como argumenta Stallman (1985:4) al compartir los programas libres entre amigos se demuestra que la publicidad comercial no es necesaria para difundir el sistema.

De esta manera, esta estimación de 2120 horas de trabajo necesarias para producir los cinco módulos de un LMS se enfrenta al secreto empresarial, el secreto profesional y a las dificultades mismas que supone la estandarización de la

programación informática misma, pero permite continuar con el ejemplo práctico planteado para este capítulo de acuerdo con la distinción establecida entre las tareas productivas y las improductivas que componen este proceso de trabajo.

5.1 Expresión monetaria de una hora de trabajo informático a partir del precio de venta

Si se asumen las 2120 horas de trabajo estimadas en el apartado anterior, se puede afirmar que para iSpring Solutions cada hora de trabajo en apariencia estaría objetivando 74.64 pesos de valor/hora; para Talent LMS cada hora de trabajo estaría objetivando 27.76 pesos de valor/hora, y para Accord LMS cada hora de trabajo estaría objetivando 37.66 pesos de valor/hora. Pero para ello, debe producirse como mercancía y venderse en el mercado.

Este resultado es aparentemente paradójico desde la argumentación de la expresión monetaria de una hora de trabajo (MELT), comentada al inicio del capítulo, ya que esta expresión debería ser unívoca y no estos tres valores distintos.

Esta objeción es válida para análisis globales de la economía capitalista en su conjunto, como en la solución de Guerrero (2011: 11), es decir, para el proceso global de la economía capitalista que aborda Marx en el tomo III de *El Capital*, aunque ya autores como Fuchs (2017) han propuesto abordajes para una única rama, como el caso de la informática en Alemania, tal y como se destacó al inicio del capítulo.

Corresponde a esta investigación proponer un ejemplo práctico para un eslabón dentro de una cadena productiva, como es el caso de un sistema informático, considerando que, desde el punto de vista de la teoría de la explotación, lo relevante es cómo esta aparente cantidad de valor objetivado por hora de trabajo constituye una

estructura relacional tal que asegure la existencia del plusvalor, en este caso, un plusvalor aparente obtenido directamente de la ganancia.

En este sentido, permanece la dificultad de que la diferencia entre el precio de venta y el precio de costo da lugar a la ganancia y no al plusvalor, motivo por el cual se ha optado por la noción de “plusvalor aparente”, ya que permite que al especificar las horas de trabajo necesarias para producir el sistema informático de gestión de aprendizaje (LMS) sea posible estimar el plustrabajo, es decir, la cantidad de horas excedentes de la jornada laboral.

Con esto, se aproxima las relaciones de explotación no en términos de valor como en los abordajes contemporáneos de la MELT, sino de horas de trabajo, es decir, planteando éstas como unidad de medición, semejante al procedimiento seguido por Marx en el capítulo VII La tasa de plusvalor del tomo I de *El Capital* (1984: 264).

Si se acepta que esta relación precio/hora de trabajo sirve como aproximación de expresión monetaria del tiempo de trabajo (MELT) para estimar el plustrabajo, entonces, se puede estimar una tasa de plusvalor a partir del plustrabajo si se asume la contratación de una empresa que utilice tecnologías libres y trabajadores autónomos para desarrollar un sistema LMS con cinco módulos, y se acepta que dicho precio no puede sobrepasar los precios de los sistemas empaquetados bajo licencia identificados anteriormente, tal que la tasa de objetivación de valor sea de 74.64 pesos, como en el precio de iSpring Solutions.

6. El bloque continental americano Norte-Sur como eje territorial del análisis de los costos de producción

Para la estimación del precio de costo de los sistemas informáticos se ha considerado en tres espacios geográficos: 1) Berkeley, California (Estados Unidos); 2) Coyoacán, Ciudad de México; 3) Heredia, Costa Rica

Este procedimiento se fundamenta, en primer lugar, en el planteamiento de Lipietz (1997: 17), quien propone una coexistencia entre paradigmas de organización del trabajo ante la emergencia de bloques continentales luego de la desintegración del bloque soviético. Según él, el neotaylorismo y el paradigma kalamariano, caracterizado este último por la implicación negociada de los trabajadores, se consolidaron en el periodo posterior al fordismo, en tanto que el toyotismo solo es un punto medio entre ambos porque añade al modo de regulación y al régimen de acumulación, como niveles de análisis de la organización del trabajo.

En este sentido, Estados Unidos constituye el centro del bloque continental en América, mientras que México y Costa Rica (así como el resto del subcontinente) forman parte de su periferia, cuyas paradojas pueden resumirse en que está alineado sobre un eje vertical de flexibilidad creciente, pero en un paradigma industrial uniformemente tayloriano (fordismo más automatización). Esto da lugar a que la competencia se establezca a costa de bajos salarios y de una pauperización de la mano de obra, lo que según Lipietz otorga a México una ventaja absoluta sobre Estados Unidos por su trabajo semicalificado en la mayoría de sus ramas (1997: 33).

En segundo lugar, se procede con este bloque en el eje Norte-Sur, en términos de la división internacional del trabajo, porque es reconocido por teóricos como Fumagalli (2010: 199), quien sostiene que el aumento en el norte de estas figuras “autónomas” se compensa por un aumento de los asalariados en el sur, o como Moulier Boutang (2012:

186), para quien existe una interdependencia entre Estados Unidos y los países del sur debido a su liderazgo en materia de capitalismo cognitivo, por ser el taller del mundo.

También Cardoso y Vercellone (2016: 47) argumentan que hay una tendencia a la "relocalización" de las actividades descentralizadas durante la crisis del fordismo hacia países de salarios bajos, porque ciertas fases del trabajo cognitivo pueden ser reubicadas hacia países del Sur, lo que ocasiona "Nortes en el Sur" y "Sures en el Norte".

En este sentido, uno de los principales argumentos de los teóricos del capitalismo cognitivo, relacionado con el tema del trabajo productivo, se refiere a las características del "cognitariado", el proletario del trabajo cognitivo definido como aquel que puede apropiarse ampliamente de los instrumentos de trabajo y evadirse en actividades no mercantiles o controladas por el capital (Moulier Boutang , 2004: 118).

Según este Moulier Boutang, (2012:183-185), este cognitariado, o clase creativa, está situada en una posición de explotación, donde se les expropia su capacidad de innovación, es decir, una explotación no en el sentido marxista. Precisamente debido a esto, este autor sostiene que las políticas de precios son superadas por el modelo de bajo costo, que supone precios muy bajos cuyo objetivo es obtener abonos, como es el caso en que la tinta es más cara que la impresora, o los teléfonos para las compañías telefónicas .

Finalmente, la selección de este eje continental permite ofrecer información actualizada y pertinente sobre las condiciones tecnológicas relativas entre México y Costa Rica para el diseño de políticas públicas que favorezcan una mejor inserción en la división internacional del trabajo.

6.1 Condiciones generales del empleo y la ocupación en California, Ciudad de México y Costa Rica

A nivel general, el salario mínimo federal en Estados Unidos para empleos no calificados era de 137.82 pesos la hora, mientras que en California, el salario mínimo era de 209.11 pesos la hora en 2019 (Department of Labor, 2019). El desempleo promedio en Estados Unidos fue de 3.7% en junio del 2019, y el de California 4.2% (Bureau of Labor Statics, 2019a).

En el caso de México, el salario mínimo vigente en 2019 fue de 102.69 pesos por jornada diaria, es decir, 12.84 pesos por hora en una jornada de 8 horas, salario válido para la ciudad de México (aunque no para la franja fronteriza, donde fue de 176.72 pesos diarios: 22.09 pesos por hora) (Secretaría del Trabajo y Previsión Social, 2019). La tasa de desempleo en México durante junio del 2019 fue de 3.5% en el territorio nacional, mientras que en la ciudad de México fue de 4.9% (Instituto Nacional de Estadística, Geografía e Información, 2019a: 14)

Finalmente, en el caso de Costa Rica, el salario mínimo nacional por jornada diaria de 8 horas fue de 342.07 pesos mexicanos en 2019, por lo que la hora de trabajo equivalió a 42.75 pesos mexicanos (Ministerio de Trabajo y Seguridad Social, 2019: 2). La tasa de desempleo en Costa Rica se ubicó en 11.9%, mientras que en la zona urbana, donde se ubica Heredia fue de 12.3% y en la zona rural fue de 10.8% (Instituto Nacional de Estadísticas y Censos, 2019: 14)

Con base en estos datos, puede estimarse que para los trabajos simples, no calificados, por hora en Estados Unidos es 10.73 veces el salario mínimo que en ciudad de México, y de 3.22 en relación con el costarricense, mientras que el salario mínimo en

la república centroamericana es 3.33 veces el salario mínimo mexicano, lo que muestra las asimetrías en términos de remuneraciones salariales en el eje Norte-Sur en cuestión.

6.2 La seguridad social

Las asimetrías en remuneraciones identificadas en el apartado anterior también se verifican en términos de los sistemas de seguridad social en los tres territorios considerados para el mismo año. En Estados Unidos existe un sistema privado, con algunas excepciones en algunos condados donde se brinda asistencia gratuita, pero no es el caso de Alameda (Berkeley), en California, donde solo se ofrece asistencia gratuita para personas de bajos ingresos y vulnerabilidad social que no puedan trabajar (Alameda Alliance for Health, 2019).

La plataforma “eHealthInsurance”, certificada por el programa de salud del gobierno estadounidense (healthcare.gov), y utilizando el código postal de Berkeley, ofrecía tres cotizaciones en 2019: 1) Óscar, con un costo mensual de 4 277.25 pesos, 2 129.12 pesos aportados por el empleador y el resto por el empleado, es decir, un costo de 12.77 pesos por hora de trabajo; 2) Kaiser Permanente, el cual tiene el mismo precio que el anterior y la misma estructura de pago; 3) United Health Care con un costo mensual de 4 752.5 pesos y una contribución del empleador de 2 186.15 pesos mensuales (eHealth, 2019), es decir, un costo de 13.12 pesos por hora de trabajo.

Para incorporar la comprensión del concepto de “autonomía” de los teóricos del capitalismo cognitivo, respecto de la ruptura con la relación salarial en términos de seguridad social, se consideraron los planes individuales de salud completamente pagados por el trabajador ofrecidos en dicha plataforma : 1) Kaiser Permanente, con un costo mensual de 5 037.65 pesos y un deducible de 15 0179 pesos; y 2) Blue of California, con un costo mensual de 7 242.81 pesos y 13 307 pesos de deducible

(eHealth, 2019). Por tanto, el costo de aseguramiento por hora de trabajo era de 30.23 pesos con Kaiser Permanente y de 43.45 pesos por hora de trabajo con Blue of California.

En el caso de la ciudad de México, se tomó la seguridad social del IMSS y sus diferentes modalidades para 2019. La tasa efectiva de contribución, sumando el aporte obrero, el patronal y el del estado, correspondió a un 24.6% del salario base de cotización, compuesto por un 16.6% de aporte de los patronos, un 2.6% por parte de los trabajadores, y un 5.4% del gobierno federal (Instituto Mexicano del Seguro Social, 2019a: 5).

En el caso de los trabajadores independientes inscritos en el régimen voluntario, (cobertura de retiro y vejez, más no la de cesantía), los montos de contribución se establecen sobre la base de un aporte nominal anual vigente desde marzo del 2019, los cuales conforme aumenta la edad, aumenta el costo. Para personas de 0 a 19 años era de 3 900 pesos anuales, 4 600 pesos de 20 a 29 años, 4 900 pesos de 30 a 39 años, 6 800 pesos de 40 a 49 años, 7 150 pesos entre 50 y 59 años, y más de 60 años 10 350 pesos (Instituto Mexicano del Seguro Social, 2019b).

Así, un programador informático, en el rango etario de 20-29 años, rango en el que se encuentran la mayoría de colaboradores de en las comunidades de programación libre y abierta (Muffato, 2006: 57), enfrenta un costo de aseguramiento por hora de trabajo de 2.3 pesos.

En el caso de Costa Rica se considera el seguro de enfermedad y maternidad, SEM, de la Caja Costarricense de Seguridad Social, y el aseguramiento voluntario del trabajador independiente para el caso de trabajadores autónomos. En el caso del trabajador asalariado, la suma de los aportes del empresario, el obrero y el estado suman

36.67% de salario reportado, aportando un 26.33% el patrono y 10.34% el trabajador (Caja Costarricense de Seguridad Social: 2019a).

En el caso del trabajador independiente se establece por escalas, con una tasa fija del 12% en total del ingreso reportado por el asegurado voluntario, tal que entre más aumenta el ingreso del trabajador más aporta y menos contribuye el estado.

Así, un ingreso de menos del salario mínimo paga un 2,89% el trabajador y un 9.11% el estado, cuando oscila entre 1 y 2 salarios mínimos el aporte del trabajador es del 4.33% y el del estado 7.67%. Cuando el ingreso del trabajador oscila entre 2 y 4 salarios mínimos su aporte es de 6.24% y el del estado 5.76%. Cuando el ingreso del trabajador va entre 4 y 6 salarios mínimos, su aporte es de 8.02 % mientras que el del estado de 3.98%. Finalmente, para ingresos mayores a 6 salarios mínimos el trabajador aporta el 10.69% y el estado el 1.31% (Caja Costarricense de Seguridad Social, 2019b).

Para el registro de trabajadores independientes en Costa Rica, se toma el salario mínimo por ley, es decir, para un trabajador no calificado que se registre como independiente, su aseguramiento tiene un costo de 1.85 pesos por hora de trabajo.

Con base en esta información, y en los supuestos hechos al inicio del capítulo de que el aporte patronal mensual se realiza para pagar 20.83 días de trabajo, 8 horas diarias, es posible señalar para el caso estadounidense la contribución social para los asalariados que perciben salario mínimo aumenta el costo en 13.31 pesos, mientras que en el caso mexicano el aumento es de 2.13 pesos por hora, y en el caso costarricense de 11.26 pesos por hora.

El caso de los trabajadores independientes, el costo de aseguramiento por hora de trabajo en Berkeley, California, es de 30.23 pesos, en ciudad de México de 2.3 pesos por hora de trabajo, y en Heredia, Costa Rica, de 1.85 pesos para quienes ganen salario

mínimo, pero en el caso de un programador informático, dados los salarios para el sector que se presentan más adelante, el costo de aseguramiento asciende a 4.74 pesos por hora de trabajo, lo que muestra las grandes diferencias entre los sistemas de salud.

6.3 El valor de la fuerza de trabajo en la programación informática

Las condiciones salariales para los programadores informáticos en el eje continental bajo estudio presentan asimetrías “Norte-Sur” como en los trabajos no calificados, lo cual queda evidenciado en los niveles salariales entre países y en relación con los trabajadores no calificados (como se detalla en la sección anterior), y más aún en la especialización en la que se encuentra la fuerza de trabajo en este sector.

Así, en el caso de Estados Unidos, tanto la división salarial entre programadores como las categorías de especialización son más marcadas que en México y en Costa Rica, lo que refleja que efectivamente dicho país lidera la industria de manera centralizada, externalizando las tareas más simples y automatizables hacia los países del sur, con salarios más bajos.

De manera global, en Estados Unidos el sector de la información aportó casi el 5% del crecimiento del producto interno bruto (PBI) durante el primer trimestre del 2019 (Bureau of Economic Analysis, 2019). La fuerza de trabajo en dicho sector fue alrededor de 2.8 millones de personas, de 161 millones de ocupados, es decir, el 1.76 % del total de la ocupación (Bureau of Labor Statics, 2019b)

En el sector específico bajo estudio, los programadores ocupados en el desarrollo de sistemas informáticos ascendían a un total de 405 330 trabajadores en Estados Unidos. En promedio se estima que ganaban un salario anual de 2 167 140 pesos, por 260 días de trabajo y un total de 2 080 horas laboradas, con un salario por hora de 1041.94 pesos, mientras que en el caso de California, la ocupación fue de 88

910 personas, con un salario por hora de 1 203.71 pesos, y un salario anual de 2 503 617 pesos (Bureau of Labor Statics, 2019c).

Esto quiere decir, que solo en Estados Unidos, un programador informático ganaba, en promedio, 7.56 veces lo que un trabajador en una ocupación no calificada, mientras que en California, en promedio, era 5.75 veces. Sin embargo, si se toman las 2 000 horas de trabajo efectivas por año, es decir, descontando 2 semanas de vacaciones, entonces el salario por hora de un programador informático era de 1 083.57 pesos para Estados Unidos, mientras que para California sería de 1 251.81 pesos.

Es indispensable señalar que pese a estos promedios, existe una marcada diferencia dentro de la división de tareas entre programadores informáticos en Estados Unidos, ya que están separados en 10 oficios con sus respectivas tasas salariales anuales (que no incluyen a los auto-empleados) con un salario medio anual y grado académico como se detalla a continuación:

- i) Científicos de la investigación en computadoras e información, 2 243 180 de pesos y grado de maestro;
- ii) Arquitecto en redes computacionales, 2 072 090 de pesos y grado de bachiller;
- iii) Programadores computacionales, 1 596 840 pesos, y grado de bachiller;
- iv) Especialista en soporte computacional, 1 007 530 pesos, sin grado específico;
- v) Analista de sistemas computacionales, 1 672 880 pesos, y grado de bachiller;
- vi) Administradores de bases de datos, 1 710 900 pesos, y grado de bachiller;
- vii) Analista de seguridad informacional, 1 862 980 pesos, y grado de bachiller;

- viii) Administradores de redes y sistemas computacionales, 1 558 820 pesos, y grado de bachiller;
- ix) Desarrollador de sistemas informáticos, 1 996 050 pesos, y grado de bachiller;
- x) Desarrolladores web, con 1 311 690 pesos, y grado de analista asociado (Bureau of Labor Statics, 2019d).

Para el territorio de California, según Simon, Klowden y Karo (2018: 13), debe destacarse que concentraba en 2016, el 27% de toda la industria de la programación informática orientada a los videojuegos de Estados Unidos (abordada tanto por los defensores del capitalismo cognitivo como sus críticos, como se ha mencionado anteriormente), y el 90% de la empresas eran pequeñas (menos de 30 empleados), aunque dicha industria no está compuesta exclusivamente por desarrolladores.

Más aún, es una industria que históricamente ha sido controlada por pocas corporaciones (Sony Interactive Entertainment, Nintendo, Microsoft, Game Studios y Tencent), la reciente distribución vía internet de juegos para teléfonos inteligentes y redes sociales virtuales ha permitido una mayor participación de programadores independientes.

Esto permite considerar que el empleo en el sector informático en California es primordialmente asalariado, ya que otras investigaciones como la de Bernhardt y Thomason (2017: 13) encuentran que en California para 2015 solo el 8.5% de la fuerza de trabajo era auto-empleada. Pese a esto, del total de autoempleados, el sector de la información pasó de representar 2.1% en el 2000 a 2.5% en 2015, es decir, un aumento coherente con el resultado anterior respecto de las posibilidades del internet y las redes sociales virtuales para la producción independiente de videojuegos en California. Esto

revela que no es cierto que en el “cognitariado” en Berkeley, CA, sea independiente o “autónomo” en términos de la relación salarial capitalista.

Para el caso de México, las personas ocupadas en 2019 en el sector de la tecnología informática y las comunicaciones eran 753 mil y estaban divididas en 12 oficios:

i) 4 oficios de dirección y coordinación: Directores y gerentes en informática; Directores y gerentes en comunicación y telecomunicaciones; Coordinadores y jefes de área en informática; y Coordinadores y jefes de área en comunicación y telecomunicaciones;

ii) 3 como profesionistas: Desarrolladores y analistas de software y multimedia; Administradores de bases de datos y redes de computadora; Ingenieros en comunicaciones y telecomunicaciones;

iii) 5 como técnicos: Supervisores de técnicos eléctricos, en electrónica y de equipos en telecomunicaciones y electromecánicos; Técnicos en instalación y reparación de equipos electrónicos, telecomunicaciones y electrodoméstico (excepto equipo informático); Trabajadores en instalación y reparación de equipos electrónicos, telecomunicaciones y electrodomésticos (excepto equipo informático); Técnicos en la instalación y reparación de redes, equipos y en sistemas computacionales; Técnicos en operaciones de equipos de radio. (Instituto Nacional de Estadística Geografía e Información, 2019b: 1)

Del total de personas ocupadas en México en 2018, el sector de las tecnologías de la información y la comunicación representó el 1.4%, la ciudad de México fue la entidad federativa donde los ocupados en el sector representaron un porcentaje más alto, con un total de 3.2%, lo cual se corresponde con el hecho de que de las poco más de 750

mil personas ocupadas en el sector a nivel nacional, el 18.1% de toda esa población se encontraba en la ciudad de México (Instituto Nacional de Estadística Geografía e Información, 2019b: 3)

En 2016 el ingreso mensual del sector (que incluye en esta medición "sueldos, salarios o jornal", "destajo" para trabajador subordinado y "sueldos o salarios" para trabajador independiente) era de 8 633 pesos mensuales, por 20.83 días al mes y 8 horas diarias (aunque en el informe se destaca que trabajan 43 horas a la semana), implicó un salario por hora de 51.8 pesos. Con doctorado o maestría, el promedio era de 18 235 pesos, mientras que los profesionales con el grado de licenciatura ganaban 11 597 pesos mensuales. Los que tenían preparatoria o su equivalente educativo tuvieron un ingreso de 4 658 pesos (Instituto Nacional de Estadística Geografía e Información, 2019b: 9).

Así, un programador informático en México tenía un ingreso, en promedio, de 4.03 veces más en relación con un trabajador que ganaba un salario mínimo en las zonas no fronterizas, y de 2.34 veces más con lo que ganaba un trabajador no calificado en la zona fronteriza con Estados Unidos.

Debe mencionarse que la composición entre trabajadores asalariados y trabajadores independientes (o autónomos) del sector informático y de telecomunicaciones en México, se caracteriza por el predominio de quienes trabajan por cuenta propia, con un 75%, mientras que solo el 20.2% son subordinados, es decir, asalariados. Además, el 2.5% contrata a otros programadores y el 1.4% no recibe remuneración (Instituto Nacional de Estadística Geografía e Información, 2019b: 7)

Pero esta composición, que favorece la interpretación de los teóricos del capitalismo cognitivo, también se acompaña con el hecho de que los trabajadores

independientes en el sector no disponen de prestaciones ni cobertura de salud. Del 20.2% de trabajadores subordinados, el 81% cuenta con servicio médico, el 85% con prestaciones como aguinaldo y vacaciones, y el 86% con un contrato escrito de base o tiempo indefinido (Instituto Nacional de Estadística Geografía e Información, 2019b: 10)

En el caso de Costa Rica, el decreto de salarios mínimos para el 2019 estableció que para un bachiller universitario, informático o de cualquier carrera, el salario mínimo mensual era de 18 262.88 pesos, por 20.83 días mensuales y 8 horas diarias, constituyó un salario por hora de 109.59 pesos. Para el caso de un licenciado universitario, informático o de cualquier carrera, el salario mínimo mensual era de 21 921.14 pesos, equivalente a un salario por hora de 131.55 pesos (Ministerio de Trabajo y Seguridad Social, 2019: 2).

Esto supone que un informático bachiller universitario ganaba, en promedio, 2.56 veces lo que un trabajador no calificado, y un informático con licenciatura ganaba, en promedio, 3.08 veces más que alguien con un trabajo considerado simple.

Por ser estos los salarios mínimos por ley, y ser el mercado informático muy dinámico en Costa Rica⁴³, los valores nominales del sector tienen asimetrías semejantes a México y a Estados Unidos. De hecho, en 2016 los administradores de tecnologías de la información formaron parte de los cuatro profesionistas con más altos ingresos con

⁴³ Alican (2005: 159) destaca que Costa Rica es uno de los principales productores de sistemas informáticos por habitante del mundo, así como uno de los países en los que hay más computadoras por habitante. También Mochi (2006: 145) destaca el caso de Costa Rica en el contexto latinoamericano de la industria de sistemas informáticos y sostiene que en dicho país se exportada tecnología informática muy sofisticada, como inteligencia artificial o biotecnología. Por ello no es casual que defensores del capitalismo cognitivo, como López, Mungaray y Pérez (2015: 214) analicen capacidades biotecnológicas del sector agrícola en México, Argentina y Costa Rica, a pesar que dicho país centroamericano pertenezca a una escala poblacional y geográfica diferente a la de los dos primeros.

45 222.95 pesos mensuales, solo superados por los titulados en microbiología, odontología y derecho judicial (Gutiérrez, Kikut, Corrales y Picado, 2018: 54-55)⁴⁴.

Aunado a lo anterior, la distribución entre asalariados y trabajadores independientes en Costa Rica, se distingue de la de México y se asemeja más a la de Estados Unidos, porque el 88.1% de los informáticos se encuentran trabajando con contrato a plazo indefinido y el 95% cuenta con aguinaldo y vacaciones. De estos, el 65% son subordinados, el 25% son jefes medios y solo el 2.5% forman parte en la alta gerencia o dirección. Así, casi no hay trabajadores independientes en el sector, ya que está directamente integrado a la manufactura industrial internacional o a la producción de empresas (Gutiérrez, Kikut, Corrales y Picado, 2018: 60-66).

La división salarial y por tareas en Costa Rica se compone de seis oficios, en el cual solo uno es de capatazgo: i) Computación, ii) Administración de Tecnologías de Información; iii) Informática Generalista; iv) Desarrollo de Software; v) Redes y Telemática; y vi) Informática Empresarial (Gutiérrez, Kikut, Corrales y Picado, 2018: 95)

Baste mencionar que en 2016 el desempleo de graduados universitarios en computación y en sistemas de información fue estadísticamente 0 (cero), pero en Administración en tecnologías de la información el desempleo fue del 18% (Gutiérrez, Kikut, Corrales y Picado, 2018:201), lo que revela que son los empleados de carácter técnico los que encuentran una alta demanda en Costa Rica, no así los encargados de tareas de gestión o administración en dicho sector.

⁴⁴ Adicionalmente, en el cantón considerado para esta investigación, Heredia, con apenas 100 mil habitantes de una población nacional total de 4.86 millones, concentra el 5.6% de todos los graduados universitarios del país (Gutiérrez, Kikut, Corrales y Picado, 2018: 49).

Lo anterior permite sostener que la división del trabajo en la rama informática en Berkeley, California y en Costa Rica está enfocada a labores técnicas propias del sector, mientras que en México existen, comparativamente, muchos puestos de dirección y gerencia relacionados con tareas que en el capítulo anterior se identificaron como improductivas dentro del proceso de desarrollo de los sistemas informáticos.

Este hallazgo puede contribuir a explicar resultados en investigaciones previas, Mochi (2006: 196) y Borja (2016: 118), de que la rama informática en México presenta un rezago estructural en el sector informático, pero que puede ser matizada con la caracterización general del mercado de trabajo que ofrecen Martínez, Valle y Sánchez (2019: 39): en tanto que Estados Unidos es un país de productividad alta y participación salarial alta, México es un país de productividad alta y baja participación salarial, mientras que Costa Rica es un país de baja productividad y participación salarial alta.

Si se toman en consideración los ingresos promedio en los tres territorios considerados, se tiene que un programador informático en California, Estados Unidos, gana 24.17 veces más que un programador informático en Ciudad de México, 9.51 veces más que uno con licenciatura en Heredia, Costa Rica.

Si se toma en consideración la cobertura a la seguridad social para asalariados, estimadas anteriormente, entonces un programador californiano gana 20.94 veces más que uno de la ciudad de México y 7.61 veces que uno herediano con licenciatura en informática, lo que muestra el carácter redistributivo de la seguridad social a nivel extraterritorial.

7. El precio de los factores objetivos para la programación informática

Como se identificó en el capítulo anterior, los factores objetivos que son transformados durante la jornada laboral de los programadores informáticos son la

memoria de almacenamiento y procesamiento de datos computados, que incluye los servidores. Además, el medio de trabajo lo constituyen los lenguajes de programación y otros sistemas informáticos para producir sistemas informáticos. Los materiales auxiliares son la electricidad y el internet, y el edificio, en tanto que espacio absoluto del proceso de trabajo, debe incluir al menos un escritorio y una silla adecuadas para 8 horas de trabajo.

7.1 Las materias primas

La capacidad de almacenamiento y procesamiento de datos de los dispositivos electrónicos constituyen las materias primas para el proceso de desarrollo de sistemas informáticos. A esto deben añadirse los servidores, que son gigantescas memorias de almacenamiento y procesamiento de datos en red, y que son utilizadas por los programadores para guardar el código, para realizar las pruebas de los sistemas, para almacenar la base de datos generada por el sistema informático, así como para probar, integrar y ejecutar los módulos desarrollados.

Para minimizar las divergencias que podrían surgir ante el uso de computadoras distintas, y aprovechando las facilidades que ofrece el comercio electrónico, se ha tomado como máquina de referencia la “HP EliteBook 1050 G1 - 15.6" - Core i7 8750H - 16 GB RAM - 256 GB SSD” que salió a la venta en Amazon el 29 de julio del 2019 y que tiene un precio de 30 358.97 pesos (Amazon, 2019a).

Las características técnicas de esta máquina (256 gigabites de almacenamiento y 7 procesadores de 8750 hertz) permiten un funcionamiento óptimo, al nivel de la programación informática, de entre 2 y 3 años, tal y como el mismo Moulier Boutang (2012: 184) afirma al respecto de que el ciclo de la informática dura 3 años, debido que en esta industria se duplica la capacidad de memoria en ese plazo. Más aún, este autor

espera que esto suceda en un año y medio debido al almacenamiento infra-atómico, que trae capacidades semejantes a las de las neuronas cerebrales⁴⁵.

Si se asume que dicha computadora tendrá una vida útil de 2 años, y se laboran 2 000 horas por año, entonces se consumirían productivamente 7.59 pesos por hora de trabajo en programación informática, es decir, esa cantidad de dinero reaparecerá en el valor de uso informático producido.

Como se mencionó al inicio de este apartado, además de la computadora utilizada por cada programador, se requiere de servidores con cierta capacidad de almacenamiento, ya que entre más complejo sea el sistema informático a producir, mayor deberá ser la capacidad de los servidores utilizados.

Este tipo de insumo es ofrecido como servicio en la nube y existe una enorme variedad de oferentes, algunos cobran tarifas mensuales y otros anuales, y por lo general están acompañados de otras herramientas para programadores, como controladores de bases de datos, entornos integrados de desarrollo (IDE), certificados y dominios (direcciones www).

Para esta materia prima se han tomado tres precios: i) el servicio Azure Cloud, de la Corporación Microsoft, cuyo precio por 100 gigabites anuales de almacenamiento es de 3 364.77 pesos, y que puede ser comprado mensualmente por 280.21 pesos (Windows, 2019a); ii) el servicio Amazon Web, cuyo precio por 100 gigabites anuales de almacenamiento es de 570 pesos en California (no disponible para Ciudad de México ni para Heredia, Costa Rica), y cuyo costo mensual es de 49.43 pesos; iii) el

⁴⁵ De hecho, el paradigma de la computación cuántica ya se encuentra desplegándose gracias al anuncio a inicios del 2019 de que saldrá a la venta el primer sistema de computación cuántica de uso comercial, denomina IBM Q System One (IBM, 2019)

almacenamiento en Digital Ocean, cuyo precio por 250 gigabites al año (no ofrecen menos) es de 1 140.6 pesos y mensual por 95.05 pesos (Digital Oceans, 2019)

Este servicio de almacenamiento en la nube debe mantenerse las 24 horas, los 7 días de la semana, aún cuando los programadores solo trabajen 5.21 días a la semana, 8 horas diarias, ya que de lo contrario, sería necesario borrar la información durante las horas no laborales, almacenándolas en alguna otra memoria, y volver a grabarlo al inicio de la jornada de trabajo. Esto se vuelve aún más contundente cuando el sistema informático ha sido desarrollado y se encuentra disponible en internet bajo una dirección IP y un dominio (una dirección www).

Aun cuando pareciera que estos precios deberían ser expresados por hora en general, como el tiempo abstracto de las tasas de interés y no como hora de trabajo en un sentido concreto, las 16 horas diarias de uso inactivo de los servidores por parte de los obreros parciales deben considerarse como parte del costo de las 8 horas laborales, tal y como los desperdicios indispensables en cualquier proceso de trabajo, pero que ingresan en su totalidad en el proceso de valorización, y que Marx (1984: 248) denomina “excrementos del proceso laboral”.

Por consiguiente, utilizando el servicio Microsoft Azure, cada hora de trabajo consumiría 1.68 pesos, (en 166.66 horas de trabajo al mes), mientras que con el servicio Amazon Web dicho consumo sería de 0.3 pesos y con Digital Oceans de 0.57 pesos si se compran los 250 gigabites, o 0.29 pesos estandarizándolo a los mismos 100 gigabites de los otros oferentes.

7.2 El medio de trabajo

En relación con el medio de trabajo, en el capítulo anterior se argumentó que los lenguajes de programación constituyen dicho factor objetivo dentro del proceso de

producción de sistemas informáticos, y que también se deben incorporar los sistemas integrados de desarrollo (IDE), así como los repositorios y controladores de versiones de código, como Github o Visual Studio Code, en tanto sistemas informáticos para producir sistemas informáticos, pero debe aclararse que no son estas las únicas tres herramientas utilizadas durante dicho proceso de trabajo.

Es precisamente aquí donde cobran relevancia las comunidades de programación libre y abierta, los obreros colectivos $(s, -s)_i$, ya que existen versiones libres de estos tres medios de trabajo para programadores, así como versiones privativas.

Esto, desde el punto de vista de la formación de valor y la valorización, supone una disminución en el precio de costo si se utilizan programas de uso libre y sin precio monetario, como Java, Linux, NetBeans, en comparación con utilizar C#, alguna versión del sistema operativo Windows (XP, 10) y Visual Studio como IDE, todos de la Corporación Microsoft o los programas de Apple Inc, y por los cuales es necesario pagar una licencia de uso.

Para el caso de utilizar tecnologías estrictamente privadas, e integrándolas con los servicios en la nube Azure de Microsoft, la licencia anual por el sistema operativo Windows 10 varía de país en país, además de estar segmentado entre versiones para uso personal (2 642.39 en Estados Unidos y 3 599 pesos en México o Costa Rica), para pequeñas empresas (3 801.81 en Estados Unidos y 5 199 pesos en México o Costa Rica) y para usuarios avanzados (5 874.09 en Estados Unidos y 7 899 pesos en México o Costa Rica) (Windows, 2019b).

Si se toman los 250 días laborales, con 8 horas cada uno, entonces el uso personal de dicho sistema operativo en Estados Unidos tiene un costo por hora de 1.32 pesos, mientras que en México y Costa Rica uno de 1.8 pesos la hora. El uso en

pequeñas empresas tiene un costo por hora en Estados Unidos de 1.9 pesos, mientras que en México o Costa Rica de 2.6 pesos por hora. Finalmente, el uso avanzado de dicho sistema operativo en Estados Unidos es de 2.94 pesos por hora, mientras que México o Costa Rica es 3.95 pesos por hora.

Estas diferencias son de interés, en primer lugar, porque revelan que un mismo medio de producción es vendido a precios más altos solo por la ubicación geográfica en perjuicio de los países de renta baja; y en segundo lugar porque el costo del medio de producción varía si se utiliza como industria domiciliar (Windows 10 para uso personal), como industria artesanal (Windows 10 para pequeñas empresas) o como manufactura (Windows 10 para usuarios avanzados).

Más aún, desde la comprensión de los teóricos del capitalismo cognitivo, solo en las dos primeras categorías (industria domiciliar y artesanal) cabe la noción de “autonomía”, pero revela que los programadores independientes en Estados Unidos enfrentan un costo menor en este medio de trabajo en comparación con los mexicanos o los costarricenses.

En el caso de utilizar tecnologías estrictamente privadas, es necesario mencionar que casi todos los lenguajes de programación son de distribución sin costo monetario, y están incorporados a los IDE y otras herramientas de programación informática, y es por el uso de estas que se cobra una tarifa de uso.

Así, el Visual Studio de Microsoft puede ser utilizado con el lenguaje C#, también propiedad de dicha corporación, tal que ambos medios de producción se obtienen por un mismo precio.

Al igual que con el sistema operativo, las tarifas varían de acuerdo al tipo de usuario, pero además, con un costo diferenciado entre el primer año de uso y la

renovación luego de ese periodo. Así, una suscripción estándar para profesionales cuesta 22 782.99 pesos el primer año y 15 188.99 pesos la renovación anual por usuario a partir del segundo año, mientras que una suscripción para empresas es de 114 040.99 pesos el primer año, y de 48 836.69 pesos la renovación anual a partir del segundo año (Microsoft, 2019c)

Con estos datos, y para la misma jornada anual de 250 días y 8 horas por diarias, el costo por hora para profesionales, el primer año, es de 11.4 pesos, y disminuye a 7.59 pesos la hora a partir del segundo año. En el caso del costo por hora para empresas, el primer año, es de 57.02 pesos por hora, y disminuye a 24.42 pesos por hora a partir del segundo año. Si se considera que se ha definido una vida útil de 2 años para la computadora, por tanto, para el uso de profesionales el costo es de 9.5 pesos por hora de trabajo, y el de empresas de 40.72 pesos por hora de trabajo.

Como se destacó anteriormente, en relación con el medio de trabajo es posible encontrar opciones sin costo monetario, tanto en lenguajes como Python o Java, como en sistemas operativos como Linux, y de los IDE como NetBeans o el Emacs-IDE, pero tienen un costo en términos de horas de trabajo en el desarrollo y depuración de dichos sistemas.

Como se destacó en el capítulo anterior, esto responde a prácticas comunitarias de colaboración, potencialmente asimilables con prácticas de otras comunidades, como ciertas en el sur de México, en cuanto que son más eficientes que las específicamente capitalistas, lo que constituye una evidencia a favor del argumento de Blondeau (2004: 42) de que estas comunidades contravienen las aspiraciones postmodernas de la disolución de todos los valores occidentales.

Sin embargo, el carácter voluntario de este trabajo ha sido criticado por Ross (2013: 215) en la medida que ha sido incorporado en muchos procesos de producción capitalista. El que los programas informáticos ingresen al proceso de producción como medios de trabajo, y no como materias primas, remite a los casos del artesano que produce sus propias herramientas en Braverman o Cafentzis, o a las brujas-médicas de Federici (2010: 275), porque controlan el conocimiento de su oficio, pero eso no implica que no puedan ser contratados por empresas capitalistas que utilizan programas libres y en código abierto para producir sistemas informáticos mediante trabajo asalariado.

Tal es el caso de las denominadas bifurcaciones (“forks” en la jerga anglosajona) (Caluking et. al.:2013, 1193), como Android, las cuales son sistemas informáticos elaborados a partir de sistemas libres o en código abierto, pero con otra licencia para uso comercial, es decir, modificando los derechos de propiedad a favor del capitalista que adelanta el capital variable. Esta práctica, sin embargo, está acompañada de pérdida de credibilidad y prestigio dentro de dichas comunidades, porque supone una división que debilita sus propios esfuerzos por mantener la calidad de dichos sistemas informáticos y divide a los programadores entre diferentes versiones de un mismo valor de uso informático.

7.3 Materiales auxiliares y del edificio en obreros colectivos $(s, -s)_j$

Además de estos factores objetivos, es necesario determinar los precios de los materiales auxiliares de la programación informática, es decir, el costo de la electricidad y del internet, así como del edificio donde se realicen las tareas en las que se divide la programación informática. En este sentido, entran consideraciones fiscales estatales y nacionales, (que no convienen a esta disertación), así como las diferencias de costos en

insumos como la electricidad y el internet, tal y como se muestra en el siguiente apartado donde se estiman para el caso de los trabajadores “autónomos”.

De esta manera, por las marcadas diferencias entre los precios de dichos insumos en el eje continental bajo estudio, una manera apropiada de resolverlo es considerar el alquiler de oficinas en el que estos tres factores objetivos son adquiridos por una tarifa única. Para ello se puede tomar Regus⁴⁶, quien tiene oficinas de alquiler en diversas modalidades en los tres territorios bajo consideración, lo que permite unificar el costo de los siguientes medios de producción i) medios auxiliares: electricidad e internet; ii) edificio: el escritorio y la silla en una oficina.

Tabla 3 Precios de las oficinas en Regus por jornada diaria de ocho horas por persona en pesos del julio del 2019.

	Berkeley, CA	Ciudad de México, Mx.	Heredia, CR
Oficinas privadas	Entre 338.38 y 465.75	83.64	205.31
Espacio compartido (“co-working”):	264.24	72.24	178.69

Fuente: Regus (2019a), Regus (2019b) y Regus (2019c).

Con base en esta información, se evidencia que el precio de una oficina de este tipo, y con la misma empresa, en California es 4.1 veces el precio que en la Ciudad de México, y 1.65 veces el precio que en San José, Costa Rica.

⁴⁶ Regus no es la única opción disponible en cada uno de los tres territorios, por ejemplo en California está Pacific Workplaces (2019), en ciudad de México está We-Work (2019) y en San José, Costa Rica está Impacthub (2019), pero su abordaje solo dificulta el análisis del capital constante, cuando el interés se centra en el capital variable, según los mismos teóricos del capitalismo cognitivo.

7. 4 Precio de costo de una hora de programación informática para obreros colectivos $(s, -s)_j$

Con base en estos datos ya es posible determinar el precio de costo de un hora de trabajo en programación informática en los tres territorios considerados para empresas capitalistas, así como la composición valor de la composición orgánica de capital, de una hora de trabajo en programación informática, tomando el salario medio en California, el ingreso medio en México, y el salario de un licenciado en programación informática en Costa Rica y utilizando los precios para empresas del sistema operativo y las tarifas de IDE para usuarios especializados.

Tabla 4: Precio de costo de una hora de trabajo en programación informática de un obrero colectivo $[(s, -s)]_j$. Berkeley, California, EE.UU; Ciudad de México, Mx; Heredia, CR. En pesos mexicanos de julio, 2019

Factor productivo		Berkeley, CA	Ciudad de México, Mx.	Heredia, CR
Fuerza de trabajo	Salario por hora del programador informático	1 251.81	51.80	131.53
	Seguridad Social	12.77	8.60	34.63
Materia prima	Computadora	7.59	7.59	7.59
	Servidores (Azure)	1.68	1.68	1.68
Medio de trabajo	Sistema operativo (Windows 10)	2.94	3.95	3.95
	Lenguaje de programación e IDE (Visual Studio)	40.72	40.72	40.72
Materiales auxiliares y edificio	Oficina con electricidad e internet	42.3	10.46	25.66
Costo total de una hora de trabajo		1 359.81	124.80	218.76

Composición valor de la composición orgánica de capital sin seguridad social	7.61	124.33	60.52
Composición valor de la composición orgánica de capital con seguridad social	7.53	106.63	47.91

Fuente: elaboración propia.

Como se muestra en la Tabla 4, la composición de valor de la composición orgánica de capital en California es muy baja comparada con la de México y Costa Rica, lo que se explica por las asimetrías en las remuneraciones ya comentadas anteriormente, y que ponen de manifiesto que es un sector intensivo en capital variable. Además, debe notarse que la inclusión de la seguridad social dentro de la composición valor de la composición orgánica de capital ocasiona que disminuya, ya constituye un aumento del valor apropiado por la clase trabajadora.

Esta tasa estimación de la composición de valor del capital puede ser entendida como el capital constante consumido por una unidad de capital variable, lo cual supone que la transferencia de valor de los medios de producción a la mercancía final es un proceso uniforme, es decir, que cada hora de trabajo transfiere siempre la misma cantidad de valor durante las 4 000 horas de trabajo establecidas como vida útil para los medios de producción.

Esto se origina en que los materiales auxiliares (internet, electricidad, servidores) así como el edificio, son alquilados según el uso específico, es decir, no se puede incluir todo su costo como capital constante, ya que se compran diaria o mensualmente, y se consumen productivamente conjuntamente con otros productores que adquieren estos medios de producción previa, simultánea y posteriormente hasta su total depreciación física o moral.

De ahí que para este ejercicio práctico no sea posible estimar una composición orgánica como lo hace Marx, y solo pueda efectuarse aproximaciones, como tomar las

2 120 horas de trabajo identificadas para producir el LMS de cinco módulos y 100 usuarios como capital variable y todo el capital constante, tal que la composición orgánica (sin seguridad social en este caso) para el proceso en Berkeley sea de 7.61, mientras que en México es de 124.33 y en Costa Rica es de 47.91.

Más aún, estas estimaciones deben ser matizadas por la crítica de Fuchs y Seignani (2013) sobre estos análisis sectoriales que no toman en consideración la cadena productiva de los dispositivos electrónicos (“hardware”), ni tampoco de las industrias extractivas. Adicional a este cuestionamiento, cabe añadirle los costos ecológicos que dichos dispositivos, incluyendo servidores, implican para la vida en La Tierra en general, no contemplados los por dichos precios monetarios.

De esta manera, con los datos de la tabla 4 producir el sistema informático en cuestión en Berkely, California, tendría un costo estimado para una empresa capitalista de 2 882 798.66 pesos. En ciudad de México, tiene un costo estimado de 264 568.35 pesos, mientras que en Heredia, Costa Rica, tendría un costo aproximado de 521 006.49 pesos.

Con base en esto, es evidente que para los precios de venta observados, y considerando los costos que enfrentan las empresas capitalistas en los tres territorios bajo estudio para producir dichos sistemas de gestión de aprendizaje (LMS), ninguna estaría dispuesta a producirlo si tuviera un único cliente porque no habría ganancia y por lo tanto, no aparecería plusvalor alguno.

Este resultado es coherente con la argumentación anteriormente ensayada respecto de las estrategias seguidas por los capitalistas de la rama en relación con la fijación de precios basados en licencias de uso, y permite centrar la atención en los apartados finales en los obreros colectivos $(s, -s)_i$ en relación con su integración al

aparato de acumulación capitalista en el eje territorial bajo estudio, ya que precisamente el debate con los teóricos del capitalismo cognitivo se articula con las comunidades de programación libre y abierta en tanto que caso paradigmático del trabajo autónomo en el capitalismo contemporáneo.

8. Precio de costos para los obreros colectivos $(s, -s)_i$: industria domiciliar y trabajo autónomo

Para disponer de información sobre el denominado “trabajo autónomo” en los teóricos del capitalismo cognitivo, deben considerarse las condiciones en las que se encuentran los programadores informáticos no asalariados, ya sea contratistas o colaboradores de las comunidades de programación libre y abierta, esto es, deben calcularse los costos que enfrentan en relación con los materiales auxiliares, electricidad e internet. Además, debe tenerse en cuenta el edificio como espacio absoluto del proceso de trabajo, el cual puede resumirse en una silla y un escritorio adecuados para tal trabajo, bajo el supuesto que el edificio es su vivienda personal y de los planteamientos del capítulo anterior sobre el efecto del IDE en términos de vigilancia y control.

8.1 Los materiales auxiliares

Para el caso de los precios de la electricidad, es necesario determinar cuánta electricidad consume la computadora seleccionada para este análisis. Existen algunas estimaciones estandarizadas, por ejemplo Strong (sf) y RapidTables (2019), sin embargo entre los blogueros de dispositivos electrónicos para computadora (“hardware”), como juanf (2010) en elotrolado.net o Vento (2011) en ozeros.com, se recomienda la plataforma Outervision.com.

En esta plataforma se pueden ingresar todas las especificaciones técnicas de una computadora (desde el tamaño de pantalla, las memorias y el uso de la tarjeta de video) y estimar por horas de uso, tal y como se muestra en la siguiente figura obtenida como salida en dicha plataforma:

Figura 10. Consumo de Kwh por 8 horas de uso de la computadora “HP EliteBook 1050 G1 - 15.6" - Core i7 8750H - 16 GB RAM - 256 GB SSD”

```
"OuterVision PSU Calculator part list https://outervision.com/b/DQ04jw
Motherboard: Desktop
CPU: 1 x Intel Core i7-8700
Memory: 16 x 16GB DDR4 Module
Video Card: 1 x NVIDIA GeForce GTX 1050
Computer Utilization: 8 hours per day
Gaming/Video Editing/3D Rendering Time: No Gaming / 3D Apps
Monitor: 1 x LED 15 inches

Load Wattage: 329W
Recommended Wattage: 379W
Note: Standard keyboard, mouse, and 8 hours of computer utilization per day already included in calculations.
Generated by OuterVision PSU Calculator 2019-08-09 02:58:12"
```

Fuente: Outervision (2019)

Como se muestra en la figura 10, esta computadora implica un consumo diario de 329 W por 8 horas de trabajo, durante 250 días de trabajo al año son 82.25 kWh anual, un consumo mensual de 6.85 kWh por mes, y por tanto, de 41.125 watts por hora de trabajo.

Para obtener este resultado se introdujeron las especificaciones de la computadora seleccionada durante 8 horas, con la excepción de que se registró como si fuera de escritorio. Esto se hizo para considerar el consumo de electricidad que supone el enrutador (“router”) durante las conexiones a internet.

Para calcular el precio monetario de estos watts consumidos, y retomando el eje continental bajo estudio, se hace nuevamente visible la diferencia entre Berkeley, California (EE.UU) y Coyoacán, Ciudad de México y Heredia, Costa Rica, en relación

con los proveedores existentes y el tipo de fuente de energía del que se obtiene la electricidad.

Así, para Berkeley, California, existe un amplio y diverso mercado de electricidad, así como una extensa literatura al respecto, por ejemplo Davis (2017: 1104), Borenstein y Bushnell (2018: 7) o Davis y Sallee (2019: 17)

Más aún, existen páginas donde se pueden comparar diversos costos de electricidad según el proveedor en Berkeley, California como Choose Energy (2019) y Electricity Local (2019), y con base en sus recomendaciones se ha seleccionado la proveedora comunitaria de energías limpias para ciudadanos, y negocios, denominada East Bay Community Energy.

El servicio residencial de East Bay Community Energy, que podría asimilarse con industria domiciliar, consiste en un costo mensual desde los 1 701.4 pesos (con la energía menos renovable) hasta los 1 845.11 pesos al mes, con energía totalmente renovable, es decir, con tasas de 4.57 pesos/kWh hasta 4.98 pesos/kWh con energía solar, con un consumo máximo mensual de 359 kWh (East Bay Community Energy, 2019: 3).

Esto quiere decir, que en la industria domiciliar, se tendrían que ocupar 37 computadoras en funcionamiento para superar dicha cantidad de energía. Si en cada hora de trabajo la computadora consume 41.125 watts, es decir, 0.041125 kWh entonces el costo por hora es de 0.19 pesos.

Para el caso de México, la Comisión Federal de Electricidad (CFE) tiene una amplia gama de tarifas, pero de carácter regional. La tarifa Doméstica de Alto Consumo (DAC) del centro del país incluye la ciudad de México, ahí aplica el costo de 107.33 pesos por mes fijos, como el mínimo mensual, al cual deben sumársele el equivalente de

25 kWh (hasta un máximo de 250 kWh por mes, tarifa 1), tramo en el que cada kWh cuesta 4.79 pesos/kWh, es decir, una tarifa total de 227.08, por tanto un costo de 1.42 pesos/kWh (Comisión Federal de Electricidad, 2019).

Esto quiere decir que para sobrepasar el máximo de 250 kWh, con la computadora seleccionada, se requerirían 36 computadoras durante un mes para sobrepasar dicha cantidad de kWh. Si en cada hora de trabajo la computadora consume 41.125 watts, es decir, 0.041125 kWh entonces el costo por hora es de 0.06 pesos.

Para el caso de Costa Rica, y manteniendo la relación de cercanía con las Universidades, se toman las tarifas de la Empresa de Servicios Públicos de Heredia S.A (ESPH) una empresa cuyo capital es 98% municipal y brinda servicios al distrito central de Heredia, donde se ubica la Universidad Nacional de Costa Rica.

En este caso, la tarifa residencial de los primeros 200 kWh cuesta 2.27 pesos/kWh, es decir, una tarifa máxima de 449.54 pesos y el adicional a 2.92 pesos. Si el consumo es menor a 3000 kWh mensuales, la tarifa por el kWh es de 2.77 pesos/kWh, para un máximo de 8 309.45 pesos. Este segmento tarifario se clasifica como comercio y servicio (Empresa de Servicios Públicos de Heredia, 2019).

Esto quiere decir que para sobrepasar el máximo de 200 kWh de consumo mensual, con la computadora especificada, se requerirían 29 computadoras en funcionamiento las 8 horas diarias durante los 20.83 días laborales. Si cada hora de trabajo la computadora consume 41.125 watts, es decir, 0.041125 kWh entonces el costo por hora es de 0.09 pesos.

Antes de estimar el costo del internet residencial, considérese que en estos linderos entre la electrónica y la computación es que suceden los fenómenos mal llamados “inmateriales”. Por un lado la electricidad funciona aquí como fuerza motriz, y

es la que se encarga de realizar los cálculos a una tasa que se mide como bytes entre segundo, ya que realizar un cómputo requiere una determinada cantidad de bytes, los cuales pueden venir en paquetes de 32 y de 64 unidades. La máquina computadora realiza las operaciones a través del procesador, y luego convierte el resultado en luz a través de la pantalla.

De esta forma, todo lo que muestra la computadora no es más que la misma energía eléctrica pero proyectada y operada a través de bombillas de luz. Los sistemas informáticos (sistemas operativos, lenguajes de programación) aparecen aquí como el mecanismo a través del cual esa electricidad es utilizada para impulsar la memoria de procesamiento de datos para obtener en el resultado deseado en términos de computación electrónica, es decir, un algoritmo o una solución automatizada.

El lenguaje de programación lo que permite es operar la máquina mediante la electricidad, hacerla que imite una acción del trabajador que éste no quiere o no puede hacer por sí mismo, como realizar muchos cálculos laboriosos en muy poco tiempo, o trazar un plano con la arquitectura de una ciudad simétricamente. Es decir, el lenguaje de programación es el medio a través del cual se utiliza este “autómata central” denominado máquina computadora⁴⁷ aprovechando el movimiento de los electrones a través de los conductores utilizados para distribuirla.

Por otro lado, la información se desplaza por el espacio a través de ondas codificado mediante el protocolo IP, y transformada en la apariencia lumínica de las pantallas de computadoras y teléfonos móviles mediante el lenguaje HTML aportado

⁴⁷ No conviene profundizar en la relación entre “Motor Analítico” en Babbage y “autómata central” en Marx (1984: 464) pero tampoco debe omitirse las semejanzas que supone la máquina computadora para la coordinación de los procesos de trabajo en el capitalismo contemporáneo y el autómata central para el sistema organizado de máquinas de trabajo en la gran industria durante el siglo XIX.

por Tim Berners Lee a inicios de los 90s, y que bajo el amparo de las licencias “copyleft” permitió el internet abierto que se popularizó desde entonces.

Esto hace que la transmisión y comunicación de la información se impulse a través de energía, y por consiguiente los dispositivos electrónicos (“hardware”) representan una limitación, tanto para enviar como para recibir la información, ya que son indispensables para materializar esas ondas como energía nuevamente y convertirla en datos y presentarlos en la pantalla.

Así, la aparente “inmaterialidad” del internet y la computación se desvanecen, y se revelan como la fría exterioridad de los dispositivos electrónicos a través de los cuales se produce, emite, recibe y proyecta información empaquetada como bytes y se proyecta bajo la forma de luz en las pantallas de ordenadores y otros dispositivos electrónicos: sus precios expresan esa realidad concreta en la que existen como factor objetivo de formación de valor y de valorización, y revelan asimetrías territoriales como cualquier otro medio de producción considerado en esta investigación.

Las diferencias en el mercado de internet entre los tres territorios bajo estudio son tan marcadas como en los otros precios, tanto en la disponibilidad de proveedores como en los servicios y tarifas que ofrecen. Por ejemplo, en Berkeley, California (cp. 94611) existen muchas proveedoras de internet, que se pueden consultar en Internet Advisor (2019), mientras que en la Ciudad de México, para el código postal 04360 de Coyoacán, está Axtel (2019) y TotalPlay (2019) y en Costa Rica, para el código postal 40101, se encuentran la proveedora estatal Kölbi (2019) y las privadas Claro (2019) y Tigo (2019).

Para facilitar la comparación y considerando las diferentes estructuras tarifarias, se calcula el costo promedio mensual durante 24 meses, que es el periodo utilizado

como referente en este ejemplo práctico, y se estima el costo por hora, como se muestra en la siguiente tabla:

Tabla 5: Costo de internet por una velocidad de 100mbs en pesos de julio del 2019

	Berkeley, CA	Ciudad de México, Mx.	Heredia, CR
Costo promedio en 24 meses	3366.35	5999	1126.82
Costo por hora	20.2	3.6	6.76

Fuente: (Excede, 2019), TotalPlay (2019) y Tigo (2019).

Así, tomando la velocidad de 100mbs en los tres territorios, se tiene que en Berkeley, CA, el costo de una hora de internet es 5.63 veces más cara que en Coyoacán, Ciudad de México, y 2.99 veces más cara que en Heredia, Costa Rica, aunque la primera se presenta como internet satelital, y las otras dos opciones no.

8.2 El edificio

Para estimar el precio del edificio, en vez de considerar los alquileres residenciales, se toma el costo de un escritorio y una silla adecuados para la ejecución de las tareas de programación informática, y su precio se obtiene de Amazon para unificarlo en los tres territorios y simplificar el cálculo de los costos que enfrentan los trabajadores autónomos. En este sentido, se considera que el proceso de trabajo se realiza en la residencia familiar o personal de la fuerza de trabajo, es decir, como industria domiciliar.

El caso del escritorio se consideró un escritorio metálico para hogar-oficina (“home-office”), cuyo precio en Amazon (2019c) es de 1 501.79 pesos, por lo que para un ciclo de vida de 24 meses, es decir, 500 días de trabajo 8 cada uno, tiene un costo de 0.38 pesos. En el caso de la silla se consideró una marca Furmax de oficina con respaldo medio, soporte lumbar giratoria, malla con reposabrazos de tela, a un precio de 855.45

pesos (Amazon, 2019c), por lo para la jornada de 8 horas, durante 500 días, tiene un costo de 0.21 pesos por hora.

8.3 El precio de costo de una hora de programación informática en el obrero colectivo $(s, -s)_i$ como industria domiciliar

De esta forma, y omitiendo los potenciales costos de envío (ya omitidos con la computadora), es posible determinar los costos de producción de sistemas informáticos por hora para los trabajadores autónomos, asimilable a la industria domiciliar en Marx y a la industria artesanal en Braverman, como se muestra en la tabla 6:

Tabla 6. Precio de costo de una hora de trabajo en programación informática. De un obrero colectivo $[(s, -s)]_i$. Berkeley, California, EE.UU; Ciudad de México, Mx; Heredia, CR. En pesos mexicanos de julio, 2019.

Factor productivo		Berkeley, CA (EE.UU)	Ciudad de México, Mx.	Heredia, CR
Fuerza de trabajo	Seguridad Social	30.23	2.30	4.74
Materia prima	Computadora	7.59	7.59	7.59
	Servidores (Pacific Oceans)	0.29	0.29	0.29
Medio de trabajo	Sistema operativo (Debian)	0	0	0
	Lenguaje de programación e IDE (Emacs)	0	0	0
Materiales auxiliares	Electricidad	0.19	0.06	0.09
	Internet	20.2	3.59	7.04
Edificio	Escritorio	0.38	0.38	0.38
	Silla	0.21	0.21	0.21
Costo total de una hora de trabajo con seguridad social		59.08	14.42	20.34
Costo total de una hora de trabajo sin seguridad social		28.86	12.12	15.60

Fuente: Elaboración propia.

Como se muestra en la Tabla 6, los obreros parciales concebidos como autónomos por los teóricos del capitalismo cognitivo, se enfrentan a costos parecidos en los casos de ciudad de México y de Heredia, Costa Rica, pero muy bajos en relación con Berkeley, California, debido a la seguridad social, electricidad e internet, aunque como se mencionó antes, los trabajadores en este sector en México no están dentro de la seguridad social y en los otros dos territorios tienden a ser asalariados.

Si se elimina la seguridad social, entonces esas diferencias se reducen de manera sensible entre los tres territorios considerados. Sin embargo, esto no implica que los obreros parciales no gasten en la salud de sus cuerpos, si estos son los depositarios del valor de uso de su fuerza de trabajo: crear valor.

Para el caso de utilizar tecnologías libres, y con base en los datos de la Tabla 6, producir el sistema informático LMS en cuestión en esta tesis para la fuerza de trabajo en Berkely, California, tendría un costo estimado de 125 256.07 pesos incluyendo la seguridad social y sin ella de 61 177.16 pesos. En la ciudad de México, tiene un costo estimado de 30 576.87 pesos con seguridad social y de 25 787.87 sin cobertura de salud, mientras que en Heredia, Costa Rica, tendría un costo aproximado de 43 129 pesos con seguro de salud y de 33 070 pesos sin él.

Con base en lo anterior, salta a la vista que existe un margen entre el precio de venta identificado y estos costos de producción, por lo que en términos de Baran y Sweezy (1980:13) existe un excedente económico, es decir, entre lo que se produce y sus costos, y por consiguiente es pertinente el análisis de la explotación, hallazgo que permite abrir la discusión al respecto de si se constituye en explotación en el sentido marxista o no, como sostiene Moulrier Boutang (2012: 185).

9. Estructura relacional de la explotación capitalista a partir de los precios y los salarios

Con base en la información presentada en apartados anteriores es posible estimar las relaciones fundamentales para el análisis de la explotación capitalista dentro del pensamiento marxista medido en horas de trabajo.

El procedimiento ensayado apunta en este ejercicio práctico al cálculo de una “plusvalía aparente” estimada a partir de los precios de venta bajo la noción de una “tasa de objetivación de valor en 74.64 peso/hora”, ya que el foco está en cómo se distribuye el tiempo necesario y el plustrabajo en una hora de trabajo, y no en las relaciones globales entre precios de producción y valores.

9.1 Los obreros colectivos $(s, -s)_i$ como empresa periférica en Berkeley, California

Un sistema de gestión de aprendizaje (LMS) de cinco módulos y para 100 usuarios puede ser producido con fuerza de trabajo de Berkeley, California, por contratistas o empresas capitalistas basadas en tecnologías libres y abiertas por un costo total de 125 249.6 pesos, y dado que una hora de trabajo independiente en este territorio tiene un costo de 59.08 pesos y asumiendo la tasa de objetivación de valor en 74.64 pesos/hora, da un excedente de 15.56 pesos, excedente que en este caso de trabajadores autonomizados del poder de mando del capital sería apropiado por ellos mismos para su reproducción como clase, pero que no son suficientes para mantener, además, a la clase capitalista propietaria de los factores objetivos del proceso de producción.

Si no se considera la seguridad social, entonces dicho excedente aumenta a 45.78 pesos, pero no lo suficiente para que cualquier trabajador, ni siquiera uno no calificado, aceptara dicha remuneración ya que no resultaría suficiente para comprar su

canasta de subsistencia relacionada con un salario mínimo por hora de 209.11 pesos por hora.

Sin embargo, como en las comunidades de programación libre y abierta sus colaboradores no necesariamente esperan ingreso alguno por sus aportes, podría aceptar apropiarse de ese excedente para su reproducción vital, aunque fuera incompleta o tuviera que complementarla con otro trabajo.

El que el costo de dicho sistema disminuya tanto, en comparación con la estructura de costos específicamente capitalista, puede ser comprendido bajo la noción de empresa periférica de Gordon, Edwards y Reich (1986: 246) en Estados Unidos durante el siglo XX, es decir, aquellas que son funcionales para atender una demanda inestable, asumen un riesgo al producir mercancías no del todo estandarizadas, y sirven para la subcontratación.

Esto a su vez permite explicar el hallazgo de Borja (2016: 161) para el caso de la relación positiva entre innovación y programación libre y abierta en la Ciudad de México, ya que el fracaso en una iniciativa en una empresa periférica implica una pérdida menor en comparación con el capital adelantado por parte de una corporación capitalista.

De esta forma, estas comunidades de programación libre y abierta en California sí están articuladas al aparato de explotación capitalista pero como trabajo barato potencialmente explotable a través de contratistas independientes u otras formas de asociación.

9.2 Tasa de plustrabajo en el obrero colectivo $(s, -s)_i$ en Ciudad de México y en Heredia, Costa Rica

Si el sistema informático bajo estudio fuera producido por programadores independientes autonomizados del poder de mando capitalista, es decir como contratistas independientes bajo las condiciones aquí planteadas, en la ciudad de México tendría un costo de 30 570. En Heredia, Costa Rica tendría un costo de 43 120.8 pesos respectivamente.

En este caso, el precio de venta de los tres sistemas de gestión de aprendizaje (LMS) es mayor que el precio de costo incluyendo su seguridad social, por lo que existe un nicho para que tanto contratistas y empresas capitalistas, que utilicen tecnologías libres y abiertas, estén dispuestas a producir dicho valor informático como una mercancía en los territorios del sur en el eje continental bajo estudio.

Así, dados los costos para trabajadores autónomos en la ciudad de México, y en Heredia, Costa Rica, y asumiendo la tasa de objetivación de valor en 74.64 pesos/hora, el excedente aumenta a 60.22 pesos y 54.3 pesos respectivamente, con lo que el margen de excedente aumenta lo suficiente como para que empresas capitalistas puedan contratar a esos programadores, pagarles una fracción de ese monto, y apropiarse ellos del remanente.

Por ejemplo, si se le pagara la hora como salario mínimo de trabajadores no calificados, tanto en ciudad de México como en Heredia Costa Rica, se generaría una estructura relacional que asegura la existencia de una ganancia, y por tanto, de un plusvalor aparente. Esto sucede, por ejemplo, con estudiantes universitarios no graduados o con trabajadores voluntarios, como denuncia Ross (2013: 215), y que

revela precisamente cómo se integran las comunidades de programación libre y abierta en el aparato social de acumulación de capital territorial e institucionalmente.

Marx (1984: 54) propone los conceptos de trabajo simple y trabajo complejo, siendo el primero la unidad de medida del segundo, para explicar el papel preeminente de un general o un banquero en una sociedad burguesa respecto al deslucido papel del ser humano sin más. Pero, si dos personas A y B se reconocen como iguales, ¿por qué una hora de trabajo de A debería objetivar más valor que una hora de trabajo de B si ambos son iguales ante la ley? ¿Por qué tiene un salario más alto el trabajo calificado que el simple, si ambos se miden en horas de trabajo y si el fundamento de la ley del valor es la igualdad humana como prejuicio popular?

La respuesta desde la teoría weberiana de clase, según Wright (1996: 695) y Fuchs (2010: 185), sería la diferencia en el nivel educativo y de oportunidades, pero desde la teoría de clase marxista, esta diferencia se origina en las diferencias en el poder de negociación de los trabajadores calificados y no calificados manifestado como una mayor apropiación del trabajo excedente, aunque a éstos se les presente como resultado de la tradición (Marx, 1984: 55)

Considérese que en México y Costa Rica la educación es pública y gratuita, por lo que es la sociedad misma quien incurre en los costos de formación de la fuerza de trabajo calificada, (como los programadores informáticos), y no los trabajadores mismos en términos monetarios, lo que permite cuestionar la validez del planteamiento weberiano de las diferencias educativas como fundamento de dichas diferencias y al mismo Marx (1984: 209) en el sentido de que los “costos de aprendizaje... entran pues en el monto de los valores para la producción de ésta [la mercancía fuerza de trabajo]”.

En esta segunda interpretación, estas diferencias de remuneración entre trabajo simple y trabajo calificado se pueden explicar como el resultado sistemático de la dinámica de acumulación capitalista y la ley que lo rige, postulada por Braverman (1998:58) como la "ley general de la división capitalista del trabajo" en tanto que:

“los trabajadores cuyo conocimiento especial y entrenamiento es conservado, son alejados cada vez más de las obligaciones del trabajo simple. Esto da lugar a una estructura que polariza extremadamente entre aquellos cuyo tiempo es infinitamente valioso y aquellos cuyo tiempo vale casi nada”.

Es decir, la estructura social generada por la dinámica de acumulación capitalista provoca y estimula estas diferencias en las remuneraciones salariales, lo cual profundiza la desigualdad en la distribución del ingreso sobre el que se soporta el orden social burgués, lo que a su vez se ve reforzado por el argumento aportado por “el juego de la inspección”, anteriormente expuesto, en el sentido de que las diferencias de salarios operan como mecanismo de control de la conducta individual, lo cual es continuamente justificado ante los ojos de la clase trabajadora mediante ideologías burguesas como la axiomática neoclásica presente en autores tan diferentes como Mankiw (2013: 23) y Piketty (2014: 21).

De esta manera, si se utiliza el salario mínimo para trabajadores no calificados, en vez del correspondiente sector, fundamentado en la noción de trabajo simple tras la comprensión del fetichismo de la mercancía (Marx, 1984: 87), se pueden calcular las relaciones de explotación marxista en este ejemplo práctico del sistema LMS, al menos bajo la clave de un plusvalor aparente en el que el trabajo simple y el salario mínimo corresponden con la experiencia de la clase proletaria.

En este sentido, para estimar la tasa de explotación como trabajo necesario y plus-trabajo a partir de una “plusvalía aparente” para contratistas y empresas capitalistas que utilizan tecnologías libres y abiertas, es necesario considerar en qué momento de una hora de trabajo se transfiere el valor de los factores objetivos a la mercancía cognitiva. Sin embargo, como el mismo Marx lo plantea, esta transferencia sucede de manera simultánea a la formación del valor y la valorización (1984: 241)

Esto permite plantear que, dado que la tasa de explotación es la relación entre el plusvalor y el capital variable, debe tomarse el producto de valor (Marx, 1984: 256) para estimar el tiempo necesario y el plus-trabajo y no la tasa de objetivación nominal de 74.64 pesos/valor porque esto conduciría a incluir erróneamente el capital constante en dicho cálculo, y solo se debe considerar el trabajo añadido.

Con base en lo anterior, el procedimiento propuesto para estimar la tasa de plusvalor, a partir de la tasa de plus-trabajo, es considerar que si en 60 minutos, una hora de trabajo, se objetiva 74.64 pesos/valor, y que el trabajo pretérito de los medios de producción en México y Costa Rica es de 12.12 pesos y 15.60 pesos respectivamente, entonces el producto de valor, o trabajo añadido, es de 62.52 pesos para México y de 59.04 pesos para Costa Rica.

Con base en esta argumentación, si se toma el salario mínimo para los trabajadores no calificados (12.84 pesos por hora) en México y un producto de valor aparente de 62.52 pesos, se puede sostener que un programador informático en México objetivaría el valor necesario para su reproducción como clase proletaria en 12.32 minutos, por lo que dedicaría 47.68 minutos a generarle un ingreso al capitalista, ya sea que se considere ganancia o plusvalor. La tasa de plusvalor aparente, calculada como

plustrabajo, sería de 387.92% para un obrero colectivo $(s, -s)_i$ compuesto por fuerza de trabajo de ciudad de México, dispuesta a trabajar por el salario mínimo.

De esta misma manera, si se toma el caso de Costa Rica, un programador informático objetivaría el valor necesario para su reproducción como clase proletaria (42.75 pesos por hora), en 43.45 minutos, por lo que dedicaría 16.55 minutos a generarle un ingreso al capitalista, ya sea que se considere ganancia o plusvalor aparente. La tasa de plusvalor aparente, calculada como plustrabajo, sería de 38.11% para un obrero colectivo $(s, -s)_i$ compuesto por fuerza de trabajo de Heredia, Costa Rica, dispuesta a trabajar por el salario mínimo.

Nótese que al tomar el salario mínimo, tanto de México como de Costa Rica, la composición de valor de la composición orgánica de capital estimada, bajo las limitaciones señaladas en apartados anteriores, es de 94,39% y 36.49% respectivamente, ya que el precio/valor de los medios de producción utilizados en la programación informática, con tecnologías libres y abiertas, tiene casi la misma magnitud que el precio/valor de una hora de trabajo no calificado en México, y representa un tercio del precio/valor en el caso costarricense. Estos resultados se resumen en la siguiente tabla:

Tabla 7. Tasa de explotación en las comunidades de programación libre y abierta contratadas para desarrollar un sistema de gestión de aprendizaje (LMS) en ciudad de México y en Heredia, Costa Rica.

	Ciudad de México	Heredia, Costa Rica
Tasa de objetivación de valor/hora	74.64	74.64

Salario por hora trabajador no calificado en pesos	12.84	42.75
Capital constante consumido por hora de trabajo en pesos	12.12	15.6
Trabajo añadido en 60 minutos	62.52	59.04
Trabajo necesario (minutos)	12.32	43.45
Plustrabajo (minutos)	47.68	16.55
Composición de valor de la composición orgánica de capital*	94.39%	36.49%
Tasa de explotación	3.869	0.3811

Fuente: elaboración propia.

*La composición de valor ha sido calculada con el capital constante consumido, ya que como se indicó anteriormente, los medios de producción, como servidores, electricidad e internet, son comprados por un plazo determinado y son de uso compartido con otros productores.

De acuerdo con la argumentación planteada bajo la noción de un plusvalor aparente, en 12:32 minutos en México y en 43:45 minutos en Costa Rica respectivamente, se produce la formación del valor durante una hora de trabajo, magnitud requerida para reproducir la fuerza de trabajo como clase proletaria, y los restantes minutos se dedican específicamente para la valorización, es decir, para producir el plusvalor del que usufructa la clase capitalista bajo la forma de ganancia por ser propietaria de los factores objetivos del proceso de trabajo.

La noción de plusvalor aparente implementada en este ejercicio práctico, construida a partir del precio de venta y de ganancia permite aproximar la división de la jornada laboral en trabajo necesario y plustrabajo, pero se distancia de los enfoques robustos sobre la expresión monetaria del tiempo de trabajo (MELT), estimados a partir

de los precios de producción, lo cual supone omitir las diferencias entre el plusvalor producido y el apropiado.

En relación con lo anterior, y a contrapunto con el poder de negociación de los trabajadores como explicación de las diferencias salariales, debe considerarse que algunas ramas económicas se apropian de más plusvalor del que producen, fundamentado en que los precios de producción favorecen a las ramas de composición orgánica alta por sobre las de una baja.

Por lo que al tomarse los precios de mercado, y no los precios de producción, se corre el riesgo de que esta estimación en la distribución entre trabajo necesario y plustrabajo manifieste esos excesos de plusvalor apropiado por la rama, y no el valor objetivado en una hora de trabajo, como ha sido el enfoque ensayado en esta investigación.

Pese a esto, utilizar los precios de mercado para aproximar el plustrabajo permite ofrecer una aproximación en horas de trabajo a dicha relación tan importante para el pensamiento marxista y el análisis de clase, y con ello cuestionar la argumentación de la "última hora" de Senior" (Marx, 1984: 270), lo que a su vez interpela la "economía de la polinización" de Moulier Boutang (2012 :175) basada en los "inmateriales 2" porque muestra que la relación de explotación es estructural durante la jornada laboral y no solo una prolongación de la misma, es decir, que se impone como plusvalor relativo y no solo como plusvalor absoluto, aunque el plusvalor estimado sea uno "aparente".

10. Apropiación capitalista de los recursos comunitarios: subordinación del obrero colectivo $(s, -s)_i$ al poder de mando capitalista sobre su trabajo

Los resultados obtenidos en apartados anteriores sobre los precios de costo de una hora de trabajo de un obrero colectivo $(s, -s)_i$ revela que los obreros parciales en

Berkeley, California, enfrentan costos monetarios mayores que sus iguales en la ciudad de México y en Heredia, Costa Rica.

Si se considera que es precisamente Estados Unidos, y con mayor relevancia California, uno de los países con más asiduos colaboradores en las comunidades de programación libre y abierta, ya que ahí surgió la licencia “Berkeley Software Distribution” (BSD), entonces se tiene un resultado contra-intuitivo en relación con la lógica del egoísmo racional de Lerner y Tirole (2002) porque no esperan una remuneración monetaria por sus aportes a los sistemas informáticos libres y abiertos que les prodigue un beneficio más alto que a los ubicados en los territorios mexicanos y costarricenses.

Sin embargo, como se ha argumentado en el capítulo anterior, dicho comportamiento debe comprenderse desde las prácticas comunitarias. En este sentido, Kundu, Swain, Biswas (2010: 228) destacan que una meta primaria en el proyecto del uso de código abierto es construir una comunidad colaborativa entre usuarios y desarrolladores. Y por tanto, ambos miembros juegan el rol determinante en volver robusto el sistema informático. De ahí que en dichas comunidades se busque la alfabetización informática para que los usuarios desarrollen una competencia suficiente en el lenguaje de programación en el cual el sistema informático está escrito, tal que puedan cambiar directamente el código y hacer el programa más seguro. Tal es el caso de R, muy utilizado entre biólogos y economistas.

También Bouras, Kokkinos, Tseliou (2013: 103) destacan que más que un ingreso monetario, los colaboradores en estas comunidades han señalado otras consideraciones a las económicas puramente egoístas, como compartir conocimiento, la satisfacción de realizar algo con valor, aprender y mejorar las habilidades personales.

También los aspectos legales de la propiedad intelectual desempeñan un rol significativo en la adopción y uso de la estos programas abiertos.

Todo esto favorece no solo la alfabetización informática, sino que facilita que sea traducido a los idiomas locales, como BOSS, un sistema operativo derivado de Debian que está traducido a 18 lenguas indias, lo que permite que los países de ingresos bajos puedan conectarse a las redes de conocimiento que posibilita el internet.

Adicionalmente, que muchos usuarios con una competencia media en lenguajes como R aporten a la comunidad, no implica que esto sea productivo desde el punto de vista de la clase capitalista, ya que en muchos casos, estos aportes se realizan en el campo de investigación académica, aunque recientemente estas tecnologías estén siendo utilizadas cada vez más por las corporaciones, ante la obsolescencia de sus propios programas, como SPSS e Eviews.

En este sentido, la participación de corporaciones capitalistas dentro de las comunidades de programación libre y abierta es un fenómeno en crecimiento y ampliamente documentado. Incluso el mismo Stallman (1985: 6) ya destacaba algunos modelos de negocios para cobrar licencias por el uso de sistemas libres desde los inicios de su proyecto GNU. Lo que resulta paradójico para los defensores de esas licencias que propone Stallman es que se considere "coercitivo" el que los productos derivados de sistemas libres, deban ser libres a su vez, pero no se considere coercitivo que Microsoft no permita distribuir libremente versiones modificadas del Office sin pagarle millones de dólares (Lessing, 2004:13)

Pese a esta crítica, después de la publicación de Lerner y Tirole (2002) se ha discutido con mayor intensidad por informáticos y economistas la participación de corporaciones capitalistas multinacionales dentro de las comunidades de programación

libre y abierta. Por ejemplo, Calukings, et.al. (2013: 1184) se centran en identificar el punto de inflexión de cuándo una empresa maximizadora de ganancias en código cerrado debe cambiar su estrategia de negocio a uno de código abierto. Dentro de esta lógica de apropiación de recursos comunitarios, sostienen que si la calidad del sistema inicial es baja, entonces es mejor iniciar con un modelo abierto, mientras que si es alta, es mejor iniciar con una licencia privada, y dependiendo de parámetros como el costo de realizar la investigación y desarrollo o los costos de mantenimiento, entonces ambos modelos pueden ser óptimos en algún momento del tiempo o nunca.

Desde esta misma perspectiva, Xing (2014: 1183) modela la competencia uno a uno de ambos tipos de sistema informático (código abierto y privado), y propone un modelo donde el propietario compite con la comunidad y con un sistema de código abierto comercial. En su estudio encuentra que los programas libres reducen los precios y las ganancias de los productores de programas privados, aunque no necesariamente disminuye la participación de mercado.

En esta misma orientación, Kendall, Kendall y Germoprez (2016: 235) destacan, entre las ventajas para las corporaciones, la habilidad para dispensar las tareas críticas en el consumo de tiempo como las pruebas y la identificación de errores (“debugging”), al trasladárselas a las comunidades de código abierto, es decir, un apalancamiento de los recursos comunitarios como estrategia competitiva.

De esta forma, las corporaciones capitalistas, además de liberar sus código en licencias “menos restrictivas”, han centrado sus esfuerzos en volver más atractivos sus propios proyectos para los programadores informáticos que, como notan Lerner y Tirole (2002: 222), al ser parte de un equipo comercial no pueden disfrutar igual los mismos

retos intelectuales y las mismas interacciones de equipos que aquellos involucrados en el desarrollo en código abierto.

Con esto, los recursos comunitarios de los obreros colectivos $(\mathbf{s}, -\mathbf{s})_i$ son representados como pasivos de las corporaciones capitalistas, y ante los cuales responden contratando obreros parciales para que realicen tareas s , como la integración o el mantenimiento, que ningún programador informático está dispuesto a realizar de manera voluntaria⁴⁸.

Adicionalmente a estas consideraciones económicas respecto de la apropiación capitalista de los recursos de los obreros colectivos $(\mathbf{s}, -\mathbf{s})_i$, es necesario mencionar que otros factores políticos han debilitado ideológicamente a dichas comunidades de programación libre y abierta, los cuales se han sucedido luego de algunas victorias organizativas, como la de Wikipedia al inicio de siglo, y jurídicas, como la del caso Jacobsen vs Katzer en 2006, donde la corte falló a favor de la licencia del lenguaje (DeBrie y Goeschel, 2016: 11), así como de los episodios de insubordinación liderados por Wiki-Leaks y Anonymus, cuyas actividades afectaron la imagen de Estados Unidos durante la época de la crisis internacional del 2008-2009.

Así, acompañado del desarrollo teórico-ideológico destacado anteriormente para apropiarse de los recursos comunitarios, se han sucedido una serie de derrotas técnicas,

⁴⁸ La reciente proliferación de los token no-fungibles (NFT por su siglas en inglés) ha promovido la eliminación de la capacidad de replicabilidad de muchas mercancías cognitivas, (videojuegos, arte visual, música, etc). Esta tecnología se apoya en las cadenas de bloques (“blockchain”) utilizadas en las criptomonedas (como bitcoin) para crear archivo digitales no replicables y cuyo principal objetivo es crear escasez artificial impidiendo que se pueda hacer copias, es decir, es una implementación de la programación informática semejante a los derechos de gestión digital (DRM) utilizados para limitar las copias de libros digitales, pero con una potencia técnica mucho mayor. Esto constituye un revés para la tesis de los teóricos del capitalismo cognitivo basadas en los comunes digitales, y una victoria de corporaciones capitalistas como Apple, que promueven una tecnología excluyente y centrada en la escasez artificial. Según Steinwold (2019) las primeras utilizaciones de las cadenas de bloques en la generación de ítem únicos vinculados a piezas de arte se dieron entre 2012 y 2013. Sin embargo desde finales del 2020 e inicios del 2021 onerosas subastas de NFTs han encendido debates sobre las implicaciones que tendrá para la rama informática, principalmente por sus costos ecológicos (Ethereum, 2021).

como la desaparición de Napster en 2001 y el modelo de las redes a pares (peer-to-peer)-que Kleiner (2010: 8) considera como la "economía política de la informática y la cultura libres" porque implementan la "comunidad de los iguales" en su arquitectura, y la aparición de las redes sociales privadas, como Twitter, Facebook o Instagram, las cuales se articulan como una red de espionaje y no como mecanismos para compartir conocimiento y colaborar, como lo han sido las redes de "hackers", definidas como antiautoritarias porque el intercambio de información no se encuentra bajo control de ninguno de sus participantes (Raymond, 2001: 82).

Además, se han presentado derrotas estratégicas, como la detención en 2012 de Julian Assange en la Embajada Ecuatoriana y su entrega a las autoridades londinenses en agosto del 2019 con un severo deterioro de su salud (Grenfel, 2019), así como la detención y suicidio (en condiciones no esclarecidas y con varios cargos federales) en 2013 de Aaron Schwartz, brillante programador informático y destacado activista vinculado a muchos proyectos de la cultura "hacker" como Creative Commons y Reddit (Knappenberger, 2014; 1:40:00)

De esta forma, tanto desde la perspectiva de la expropiación creciente de la riqueza comunitaria de los obreros colectivos $(s, -s)_i$, como desde la perspectiva ideológica y política, ha habido un debilitamiento de sus experiencias de insubordinación, y por consiguiente una tendencia a una subjetivación política con predominio de la subalternidad sobre las antagonistas y autonómicas.

En consecuencia, la conjetura de que la organización del proceso de trabajo en las comunidades de programación libre y abierta tiende a hegemonizarse en el capitalismo cognitivo, debe considerarse una conclusión apresurada por parte de los teóricos que defienden dicha tesis.

Finalmente, el auto-control y autogestión del proceso de trabajo de los obreros colectivos $(s, -s)_i$ que les ha permitido desplazar técnicamente al de los obreros colectivos $(s, -s)_j$ en varios ámbitos de la rama informática, queda registrado en los marcadores de los IDE utilizados por esos programadores informáticos, lo que facilitará que este proceso de trabajo pueda ser completamente automatizado mediante inteligencia artificial y que los programadores informáticos sean sustituidos por robots⁴⁹.

Esto no debería ser una sorpresa, ya que como Huerta (2017: 60-64) plantea, el origen de la inteligencia artificial, o aprendizaje de máquinas, se remonta directamente a la colaboración en la actividad laboral y su aplicación en diversos campos ha implicado sustituciones de miles de empleos en diversas empresas. Precisamente la argumentación de Turing (1950) busca que las máquinas aprendan a imitar a los seres humanos, y con ello construir robots que se encarguen de realizar todos los procesos laborales.

De ahí que Turing ponga su énfasis, precisamente, en cómo “educar” una máquina para que imite a una persona, pero su sesgo ideológico le hace considerar a la “máquina humana” como si fuera un animal condicionable mediante premios y castigos modelados como probabilidades (Turing, 1950: 457), por lo que se le pueda extender la crítica que Braverman le hace al “management” científico que busca educar al trabajador como a un caballo, es decir, con látigo y zanahoria (1998: 47).

⁴⁹ Se puede vaticinar siguiendo a Braverman (1998: 227) que así como el desarrollo del oficio de procesamiento de datos fue “abortivo”, porque la introducción de la computadora produjo una nueva división del trabajo que aceleró la destrucción del oficio recién creado, la programación informática de la inteligencia artificial con redes neuronales sustituirá en un futuro a varios de los obreros parciales que actualmente forman parte de los obreros colectivos dedicados al desarrollo de sistemas informáticos, y con ello, disminuirán sus salarios y en última instancia, también las comunidades de programación libre y abierta podrían desaparecer, y con ello, el poder de negociación asociado a su propiedad sobre el medio de trabajo y el control sobre su propia actividad laboral.

Pese a estas críticas de Braverman, y las de Marcuse (1969: 14) al respecto de la alienación, la automatización y la robotización dentro de los procesos de trabajo son procesos históricos en marcha. Por ejemplo Huang Yu (2019: 56) estima que la densidad de robots en la industria China es de 68 unidades por cada diez mil trabajadores en 2017-2018, y que cada máquina automática destruye entre 1.5 y 5.6 puestos de trabajo ocupados por miembros de comunidades locales, y reducen el salario entre 0.25% y 0.5%. Para este autor, sin embargo, no ha habido suficiente investigación sobre el impacto de la robotización sobre los trabajadores, aunque con la información que dispone de cuatro empresas que poseen datos comparativos antes y después de la automatización, la reducción de la fuerza de trabajo oscila entre 67 y 85%.

Capítulo 5. Conclusiones

En este apartado se informa sobre el cumplimiento de los objetivos de la investigación y de los resultados destacados al respecto de las tesis de los teóricos del capitalismo cognitivo sometidas a debate a lo largo del documento.

El desarrollo de esta investigación a partir de las tesis de los teóricos del capitalismo cognitivo ha permitido obtener resultados satisfactorios, aunque debatibles,

al respecto del poder de negociación de los trabajadores informáticos en la condiciones del capitalismo contemporáneo.

En relación con el análisis del proceso de trabajo en la programación informática, se identificó que las materias primas de dicho proceso son las memorias de almacenamiento y procesamiento de datos (incluidos los servidores) con base en los planteamientos de Turing, lo cual refuta la tesis de Blondeu de que son una fuerza abstracta, mal llamada “inmaterial”, que dificulta separarla del cuerpo de los trabajadores de dicha rama.

En este sentido, se precisaron los lenguajes de programación y otros sistemas informáticos para producir sistemas informáticos, como el medio de trabajo en dicho proceso en tanto es a través de ellos que los programadores informáticos operan las máquinas computadoras, y otras máquinas mecánicas, para realizar tareas que no pueden o no quieren realizar por sí mismos, al igual que lo hace un operario industrial con la trefiladora que corta y estira el metal para elaborar alambres.

Además, se identificaron siete tareas productivas dentro del proceso de desarrollo de los sistemas informáticos desde el punto de vista de la clase capitalista: i) arquitectura, ii) diseño, iii) codificación, iv) pruebas, v) integración, vi) instalación y vii) mantenimiento, y se argumentó, a partir de las diversas formas de organizar dicho proceso de trabajo en las comunidades de programación libre y abierta y su práctica de compartir el código, que las tareas relacionadas con la vigilancia y el control, así como la de comercialización, son improductivas para la clase capitalista, a pesar que sea ella mismas quien las introduce para asegurarse de obtener una ganancia.

En este sentido, se propuso el concepto de flexibilidad modular como una estrategia de la clase capitalista implementada como mecanismo de limitación del poder

de negociación de los trabajadores emanado de la amenaza de paralización del proceso de trabajo, la cual está basada en el “Principio de Babbage” de solo comprar las cantidades necesarias de cada tipo de fuerza de trabajo, posibilitado técnicamente por el control numérico que introduce la programación informática en los procesos de trabajo, partiendo de ese mismo proceso de trabajo.

Además, se precisaron acontecimientos durante los casi 40 años de existencia de las licencias libres en las que se funda parte del poder de negociación de los programadores informáticos, que aportan evidencia de que la subjetivación política de los mismos, entendida como las experiencias de clase de autonomía, antagonismo y subalternidad, ha estado marcada por la apropiación capitalista de sus recursos comunitarios y una mayor subordinación a su poder de mando, principalmente en términos de producción de contenidos o valores de uso mentales en términos de Carchedi, pero no en la imposición de los tiempos de trabajo ya que las tareas de arquitectura, diseño, codificación, pruebas e integración no han sido suficientemente estandarizadas como para automatizarlas y controlar sus tiempos de ejecución.

En este sentido, se identificaron cuatro fundamentos económicos que explican la mayor productividad de estas comunidades, en términos de horas de trabajo, en relación con los programadores informáticos asalariados por corporaciones capitalistas: i) la modularidad, ii) el reciclaje de código, iii) la ley de muchos ojos, iv) prácticas comunitarias de colaboración y apoyo mutuo.

En términos de remuneraciones, el carácter voluntario de muchos colaboradores de los proyectos libres y abiertos, ha sido aprovechado por las corporaciones capitalistas para externalizar algunas de las tareas, apalancando sus recursos comunitarios y reduciendo sus costos de producción. En este sentido, se cuestionó la división salarial

capitalista entre trabajos no calificados, o simples, y trabajo calificados, o potenciados, a partir de la igualdad humana como fundamento de la ley de valor.

En relación con la tesis de una tendencia hacia la hegemonización de las características autonómicas de las comunidades informáticas, se destacó el creciente interés académico por los intelectuales e ideólogos al servicio de los intereses capitalistas encargados de sistematizar las experiencias de las principales comunidades de programación libre y abierta, y derivar de tal conocimiento nuevas técnicas de subordinación que favorezcan su apropiación, así como otros acontecimientos de carácter político, como la detención de líderes muy influyentes entre la cultura “hacker”, que ponen en tela de duda que las características antagonistas de estas comunidades se profundicen hasta constituirse en una amenaza global para la estabilidad y continuidad de la acumulación capitalista como principal generador de riqueza y pobreza en la actualidad.

También se estudió la relación de las licencias abiertas, “copyleft”, con el poder de negociación de estos trabajadores, en tanto que constituyen una forma de propiedad comunitaria sobre su medio de trabajo, semejante a la de los artesanos de los siglos XVII y XVIII que estudió Marx y que Cafentzis utiliza para cuestionar la tesis de Vercellone de que las características de las comunidades de programación libre y abierta tienden a hegemonizarse en el capitalismo contemporáneo.

Más aún, se propuso una tipología para los usuarios de valores de uso informático basada en sus competencias de alfabetización informática para cuestionar la pertinencia del término “prosumer” y aportar evidencia a favor de que las tecnologías privadas promueven el analfabetismo informático como estrategia para espiar y secuestrar información que facilite la automatización de los procesos de trabajo en los

que se introducen sus sistemas informáticos, así como para favorecer el control burocrático a partir de la vigilancia civil.

En relación con el análisis del proceso de formación de valor y de valorización en los sistemas de gestión de aprendizaje se propuso un procedimiento intuitivo desde los precios de venta, identificados en plataformas en internet, para aproximar los valores en el marco de la discusión contemporánea dentro del pensamiento marxista sobre la transformación de los valores en precios con base en la expresión monetaria del tiempo de trabajo.

Así mismo, se propuso un abordaje de la difícil relación entre valores y precios centrando la atención en el análisis de la explotación capitalista a partir de “una hora de trabajo informático” como unidad de medición y se reconocieron las debilidades del mismo en términos de la incompletitud de considerar una rama y no todo el proceso global de producción, así como de tomar el precio de venta como si fuera el valor justificado en la centralidad del análisis de la explotación para el pensamiento marxista.

Con base en esto, se realizaron estimaciones de los costos de producción por hora de trabajo en tres territorios del bloque continental americano arraigados en la rama informática: Berkeley, California, Estados Unidos , Ciudad de México, México y Heredia, Costa Rica, con lo que se verificó las profundas asimetrías en términos de las remuneraciones a la fuerza de trabajo, y que favorecen la externalización de las tareas que más requieren horas de trabajo, como la codificación y las pruebas, desde el Norte hacia el Sur del continente. Así, se encontró que un programador informático en California, Estados Unidos, gana aproximadamente 24.17 veces más que uno en la ciudad de México, México y 9.51 veces más que uno en Heredia, Costa Rica.

Los costos estimados para dichos territorios se diferenciaron entre aquellos propios de las corporaciones capitalistas que utilizan fuerza de trabajo asalariada y tecnologías privadas, de los costos que enfrentan los programadores al participar activamente en las comunidades fundadas alrededor de lenguajes y sistemas informáticos distribuidos mediante licencias libres y abiertas, en tanto que representantes del trabajo autónomo para los teóricos del capitalismo cognitivo y tipificadas como industria domiciliar por Marx en su momento.

Se estimó que las corporaciones capitalistas del sector presentan una composición de valor de su composición orgánica de 7.53 % en Berkeley, California, 106.63 % en Ciudad de México, y de 47.91 % en Heredia, Costa Rica, lo que expresa que es una rama intensiva en capital variable y cuyas diferencias entre territorios se explican por las asimetrías en términos de las remuneraciones antes señaladas, dadas la particularidad del cálculo al respecto de utilizar solo el capital constante consumido.

Se estimó una tasa de plusvalor aparente, a partir del precio de venta, para aproximar el plustrabajo para programadores informáticos en la ciudad de México, México y en Heredia, Costa Rica, contratados en una empresa capitalista para desarrollar un sistema de gestión de aprendizaje (LMS) de cinco módulo y para 100 usuarios, utilizando programas libres y abiertos en la industria domiciliar y con una remuneración del salario mínimo nacional (y no el mínimo de la rama) para su reproducción como clase proletaria.

En el caso de los programadores informáticos en la ciudad de México, México y con las condiciones anteriores, se estimó que en 12.32 minutos objetivan el valor necesario para su reproducción, por lo que dedican 47.68 minutos a generarle un ingreso

al capitalista, considerado como un plusvalor aparente, calculado como plustrabajo, en 387.92%.

En el caso de programadores informáticos en Heredia, Costa Rica, y con las mismas condiciones antes expuestas, se estimó que en 43.45 minutos objetivan el valor necesario para reponer su fuerza de trabajo, por lo que dedican 16.55 minutos a la generación del plusvalor con el que se reproduce la clase capitalista, considerado como un plusvalor aparente, calculado como plustrabajo, en 38.11%.

Esta utilización del salario mínimo para trabajos no calificados, simples, se argumentó a partir de hecho de que muchos de los participantes en dichas comunidades de programación libre y abierta lo hacen de manera voluntaria, o desde una posición como estudiantes universitarios no graduados o aún en proceso de formación, lo que facilita que no perciban ni siquiera los salarios mínimos de la rama.

Con base en esto, se situó el desarrollo de los sistemas informáticos libres y abiertos en la función de empresa periférica de teóricos de la economía política radical en relación con el aparato de acumulación capitalista globalizado a partir del internet.

Finalmente, se argumentó que la cantidad de horas de trabajo estimadas para desarrollar un módulo de LMS en 424 no se vio cuestionado por la aparente disolución de la jornada laboral que posibilita el trabajo remoto de programadores informáticos a través del internet, ya que tanto los entornos integrados de desarrollo (IDE) privados como los libres y abiertos, constituyen monitores en tiempo real de la cantidad de horas de trabajo y de las acciones realizadas por los programadores informáticos.

De esta forma, se sostiene que la formación del precio de venta sí se realiza con base en la cantidad de horas de trabajo añadida a los medios de producción al transformarlas signándolas con líneas de código en ese tiempo, y por tanto se rechaza la

hipótesis de la crisis de la ley de valor sostenida por los teóricos del capitalismo cognitivo para la rama informática.

Con base en lo anterior, y en relación con el objetivo general de la investigación, se concluye que el control numérico que introducen los valores de uso informáticos en los procesos de trabajo tiene efectos ambiguos en el poder de negociación de los trabajadores en el caso paradigmático de la programación informática.

Por un lado, posibilitan la flexibilidad modular como estrategia de la clase capitalista para eliminar los episodios de insubordinación individual como el sabotaje, y de insubordinación colectiva, como la paralización del proceso de trabajo, mediante el reemplazo de los obreros parciales, cuyos registros en el entorno integrado de desarrollo (IDE) permiten anticipar desviaciones de la conducta de subordinación exigida por su poder de mando como clase propietaria de los factores objetivos del proceso de producción.

Y por otro lado, la forma de propiedad comunitaria expresadas en las licencias libres y abiertas, permite experiencias de autonomía en relación con la producción de valores de uso informáticos que no están motivados por el interés capitalista de producir mercancías cognitivas para apropiarse de su excedente, como Wikipedia, así como experiencias de insubordinación vinculadas con la cultura “hacker” abiertamente antagónicas con la forma de propiedad capitalista sobre la información y el conocimiento.

Pese a ese carácter ambiguo, las coordenadas de subjetivación política ensayadas, a partir de lo propuesto por Modonessi, permiten sostener que la creciente participación de corporaciones capitalistas dentro de las comunidades de programación libre y abierta, así como la alfabetización informática promovida por las primeras en

los usuarios de las tecnologías de la Web 2.0, son señales de que la configuración de experiencias autonomía-antagonismo-subalternidad se encuentra en una reconfiguración hacia el predominio de las últimas, en contraste con el predominio antagonista que las caracterizó desde sus orígenes, el cual se ha debilitado con mayor intensidad a partir del periodo comprendido entre la crisis de las dot.com en Estados Unidos en 2001 y la Gran Recesión del 2008-2009.

Bibliografía

- Accord LMS (2019) Affordable LMS Pricing. Pricing. Disponible en <https://www.accordlms.com/lms-pricing> (Consultado el 15/07/2019)
- Agamben, Giorgio (2010) Signata Rerum. Sobre el método. Barcelona: Editorial Anagrama. ISBN: 978-84-339-6305-5
- Ahmad, Muhamad Ovais; Markkula, Jouni; Oivo, Marrku (2013). Kanban in software development: A systematic literature review. In Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO. Conference on (pp. 9-16).
- Alameda Alliance for Health (2019) Alliance Group Care. Visitors. Disponible en <https://www.alamedaalliance.org/visitors/health-plans/group-care> (Consultado 26/07/2019)
- Alican, Fucac (2005) Software para el desarrollo. Cartago, Costa Rica: Editorial Tecnológica de Costa Rica. ISBN 9977-66-173
- Althusser, Louis (2004) Crítica previa a la lectura de El Capital. México: Ediciones Paradigmas y Utopías.
- Amazon (2019a) HP EliteBook 1050 G1 - 15.6" - Core i7 8750H - 16GB RAM - 256 GB SSD. Disponible en https://www.amazon.com/dp/B07VSHK6XQ/ref=sr_1_2?fst=as%3Aoff&qid=1565373848&refinements=p_89%3AHP&rnid=2528832011&s=computers-intl-ship&sr=1-2 (Consultado el 10/08/2019)

Amazon (2019b) Escritorio esquinero para el hogar o la oficina con forma de L de LeCrozz. Disponible en <https://www.amazon.com/Escritorio-esquinero-hogar-oficina-LeCrozz/> (Consultado el 10/8/2019)

Amazon (2019c) Furmax Silla de oficina con respaldo medio, silla de escritorio con soporte lumbar giratoria, silla ergonómica de malla con reposabrazos, Tela. Disponible en <https://www.amazon.com/respaldo-escritorio-giratoria-ergon%C3%B3mica-reposabrazos/dp/B07B7K7N3P/> (consultado el 10/08/2019)

Amazon Web Services (2019) Precios de Amazon S3. Almacenamiento. Productos. Disponible en <https://aws.amazon.com/es/s3/pricing/> (consultado el 8 de julio del 2019)

Álvarez Newman, Diego (2012) El toyotismo como sistema de flexibilización de la fuerza de trabajo. Una mirada desde la construcción de productividad en los sujetos trabajadores de la fábrica japonesa (1994-2005) Revista de estudios transfronterizos. Volumen XII No.2 pp181-201

Aneesh, A (2009) Global Labor: Algocratic Modes of Organization. Sociological Theory Vol 27 No 4 pp.347-369

Avancini, Henry; Amandí, Analía (2000) A Java Framework for Multi-Agent Systems. SADIO Electronic Journal of Informatics and Operations Research. Vol. 3, no. 1, pp. 1-12

Axtel (2019) Axtel Xtremo. Disponible en <https://landing.axtel.mx/nuevos/res/search/xtremo/> (Consultado el 20/07/2019)

- Babbage, Charles (2009) On the economy of manchineer and manufactures. Londres: Charles knight Pall Mall East. ISBN 978-1-108-00910-2
- Banco Central de Costa Rica (2019) Tipos de cambio de venta del dólar estadounidense. Indicadores económicos. Disponible en <https://gee.bccr.fi.cr/indicadoreseconomicos/Cuadros/frmVerCatCuadro.aspx?idoma=1&CodCuadro=%20370> (Consultado el 26/07/2019)
- Banco de México (2019) Tipos de cambio peso/dólar. Sistema de información económica. Disponible en <https://www.banxico.org.mx/tipcamb/main.do?page=tip&idioma=sp> (Consultado el 26/07/2019)
- Bacaro, Lucio (2008) Labour, Globalization and Inequality: Are Trade Unions Still Redistributive? Suiza: Organización Internacional del Trabajo. ISBN 978-92-9014-882-1
- Baran, Paul; Sweezy, Paul (1980) El Capital Monopolista. México: Siglo XXI Editores. ISBN:968-23-0440-7
- Barthes, Roland (1993) La Aventura Semiológica. Barcelona: Paidós. ISBN: 84-7509-581-X
- Benbya, Hind; Belbaly, Nassim (2002) The “new” new economy: lessons learned from the burst of dot-com’s bubble, dispelling the myths of the new economy. Journal of E-Business, Vol. 2, No. 2
- Bernhardt, Annete; Thomason, Sarah (2017) What Do We Know About Gig Work in California? An Analysis of Independent Contracting. UC Berkeley Labor

Center. Disponible en <http://laborcenter.berkeley.edu/pdf/2017/What-Do-We-Know-About-Gig-Work-in-California.pdf> (Consultado el 17/10/2019)

Blondeau, Olivier (2004) Génesis y subversión del capitalismo informacional. En Capitalismo cognitivo, propiedad intelectual y creación colectiva. Ed. Emmanuel Rodríguez López y Raúl Sánchez Cedillo. Madrid: Traficante de Sueños. ISBN: 84-933555-0-X

Brañas Garza, Pablo; Barreda Ivan (2011) Economía Experimental y del comportamiento. Pablo Brañas Garza, coordinador. Barcelona: Antonio Bosch Editor.

Braverman, Harry (1998) Labour and Monopoly Capital. The degradation of work in Twentieth Century. New York: Monthly Press Review. ISBN 0-85345-940-1

Borestein, Severin; Bushnell, James (2018) Do Two Electricity Pricing Wrongs Make a Right? Cost Recovery, Externalities, and Efficiency. Energy Institute WP 294. Energy Institute at Haas. Disponible en <https://haas.berkeley.edu/wp-content/uploads/2019/05/WP294.pdf>. (Consultado el 11/08/2019)

Borja Rodríguez, Blanca Araceli (2016) Innovación y Software Libre en México. Modelos de innovación privativo y colectivo en empresas de la Industria de Software en el Distrito Federal. 2013-2015. Tesis para optar por el grado de maestro en economía. México: Facultad de Economía, UNAM.

Bouras, Christos; Kokkinos, Vasileios; Tseliou, Georgia (2013) Methodology for public administrators for selecting between open source and proprietary software. Telematics and informatics. No. 30. pp.100-110

- Boyer, Robert (2007) Regímenes de acumulación y dinámica histórica. En Crisis y regímenes de crecimiento. Introducción a la teoría de la regulación. Argentina: Miño y Dávila Editores ISBN: 8496571467
- Bucella, Dominico; Fanti, Luciano (2018) A Game theoretic Approach to the Choige of Union Olygopoly Bargaining Agenda. Investigación Económica, Vol. 77, No. pp. 97-120
- Bureau of Economic Analysis (2019) Gross Domestic Product by Industry: First Quarter 2019, US. Disponible en <https://www.bea.gov/news/2019/gross-domestic-product-industry-first-quarter-2019> (Consultado el 20/08/2019)
- Bureau of Labor Statics (2019a) Unemployment rates for States. Local Area Unemployment Statitics, US. Disponible en <https://www.bls.gov/web/laus/laumstrk.htm> (Consultado 26/07/2019)
- Bureau of Labor Statics (2019b) Employment by Mayor Industry Sector, US. Disponible en <https://www.bls.gov/emp/tables/employment-by-major-industry-sector.htm> (Consultado el 20/08/2019)
- Bureau of Labor Statics (2019c) Software Develeoment, System Software. Occupational Employment and Wages, US. Disponible en [https://www.bls.gov/oes/current/oes151133.htm#\(3\)](https://www.bls.gov/oes/current/oes151133.htm#(3)) (Consultado el 20/08/2019)
- Bureau of Labor Statics (2019d) Computer and Information Technology Occupations. Occupational Outlook Handbook, US. Disponible en <https://www.bls.gov/ooh/computer-and-information-technology/home.htm> (Consultado el 20/08/2019)

- Cafentzis, George (2016) Una crítica del capitalismo cognitivo. Hipertextos, Vol. 4, N° 6. pp-13-49
- Caja Costarricense de Seguridad Social (2019a) Calculadora Patronal. Distribución de los porcentajes de cotización. Disponible en <https://www.ccss.sa.cr/calculadora> (Consultado el 11/08/2019)
- Caja Costarricense de Seguridad Social (2019b) Cuál es el porcentaje de cotización para el seguro voluntario o trabajador independiente y cómo se determina el ingreso de referencia? . FAQ. Disponible en <https://www.ccss.sa.cr/faqs?cat=96> (Consultado el 11/08/2019)
- Calo, Ryan; Rosenblat, Alex (2017) The taking economy: Uber, information and power. Columbia Law Review, Vol. 117, No. 6. pp. 1623-1690
- Calukins, Jonathan; Feichtinger, Gustav; Grass, Dieter; Hartl, Richard; Kort, Peter; Seidl, Andrea (2013) When to make proprietary software open source. Journal of Economic Dynamics & Control. No 37. pp. 1182–1194
- Caple, Chris (1992) 'The detection and definition of an industry : the English medieval and post medieval pin industry.', Archaeological journal., Vol. 148 No. 1. pp. 241-255.
- Cardoso, Pablo; Vercellone, Carlo (2016) Nueva división internacional del trabajo, capitalismo cognitivo y desarrollo en América Latina. Chasqui. Revista Latinoamericana de Comunicación. No 133. pp. 37-59. e-ISSN 1390-924X
- Carchedi, Guglielmo (2014) Old wine, new bottles and the Internet. Work Organisation, Labour & Globalisation. Vol. 8, No. 1. pp. 69-87

Carpenter, Jeffrey; Matthews, Peter Hans (2003) Behavioral Marxism I: Collective Action. Middlebury College. Vermont, EUA: Department of Economics. Pp. 1-24

Comisión Federal de Electricidad (2019) Tarifa DAC. Servicio doméstico de alto consumo Conoce tu tarifa. Consulta tu tarifa. Disponible en <https://app.cfe.mx/Aplicaciones/CCFE/Tarifas/TarifasCRECasa/Tarifas/TarifaDAC.aspx> (Consultado el 14/08/2019)

Choose Energy (2019) Electricity cost by state. Electricity. Disponible en <https://www.chooseenergy.com/shop/plan-details/pge/pge-ca-electricity/time-of-use-rate-e6-2570-6476/> (Consultado el 14/08/2019)

Claro (2019) Internet Inalámbrico Residencial. Servicios. Disponible en <https://www.claro.cr/personas/servicios/servicios-hogar/internet/> (Consultado el 20/08/2019)

Crowley, Martha; Tope, Daniel; Joyce Chamberlain, Lindsey; Hodson, Randy (2010) Neo-Taylorism at Work: Occupational Change in the Post-Fordist Era. Social Problems, Vol. 57, No. 3. pp. 421-447

Cuesta Santos, Armando (2008) La productividad del trabajo del trabajador del conocimiento. Ingeniería Industrial, vol. 29, No. 3, pp. 1-5

Davis, Flora (2016) La comunicación no verbal. Madrid, España: Alianza Editorial. ISBN: 978-84-206-6424-8

David, Lucas; Sallee, James (2019) Should Electric Vehicle Drivers Pay A Mileage Tax? Energy Institute WP 301. Energy Institute at Haas. Disponible en

<https://haas.berkeley.edu/wp-content/uploads/2019/05/WP301.pdf>. (Consultado el 15/08/2019)

Davis, Lucas (2017) Evidence of a Decline in Electricity Use by U.S. Households. Economics Bulletin. Vol 37. No.2. pp 1098-1105.

Davis, R. W (1989) Stakhanovism and the Soviet System: A Review Article. Soviet Studies, Vol. 41, No. 3. pp. 484-487

Debie, Elsie; Goeschel, David (2016) Open Source Software Licences: legal implications and Practical Guidance. The Nebraska Lawyer Magazine. Vol. 19 No. 2 pp 7-13

De la Garza Toledo, Enrique (1994) Las teorías de la elección racional y el marxismo analítico. Estudios Sociológicos. Vol 12. No 35. Pp 357-379

De la Garza Toledo, Enrique (1993) Reestructuración productiva y respuesta sindical en México. México: UNAM/UAM ISBN: 970-620-357-5

Department of Labor (2019) State minimum wage laws. Wage and hour division, US. Disponible en <https://www.dol.gov/agencies/whd/minimum-wage/state> (Consultado el 26/07/2019)

Departamento del Trabajo y Previsión Social (2019) Salarios mínimos vigentes a partir del 1 de enero del 2019. Comisión Nacional de Salarios Mínimos, Estados Unidos Mexicanos. Disponible en https://www.gob.mx/cms/uploads/attachment/file/426395/2019_Salarios_Minimos.pdf. (Consultado el 11/08/2019)

- Díaz Polanco, Héctor (2011) La rebelión zapatista y la autonomía. México: Siglo XXI Editores
- Digital Ocean (2019) Spaces Object Storage. Storage. Disponible en <https://www.digitalocean.com/pricing/#Storage> (Consultado el 8/07/2019)
- Dikson, Davis (1988) Soviet computer lag. Science, New Series, Vol. 241, No. 4869 p. 1034
- DiNardo, John; Fortin, Nicole; Lemieux, Thomas (1996) Labor Market Institutions and the Distribution of Wages, 1973-1992: A Semiparametric Approach. Econometrica Vol 65, No. 5. pp- 1001-1044
- Downey, Allen; Eleker, Jeffrey; Meyers, Chris (2002) Aprenda a pensar como un programador. Con Python. Wellesley, Massachusetts: Green Tea Press. ISBN 0-9716775-0-6
- Dussel, Enrique (2006) 20 tesis de política. México: Siglo XXI. ISBN: 968-23-2626-5
- Dyers Whiteford, Nick (2004) Sobre la contestación al capitalismo cognitivo. Composición de clase de la industria de los videojuegos y de los juegos de ordenador. En Capitalismo cognitivo, propiedad intelectual y creación colectiva. Eds Olivier Blondeau, Nick Dyers Whiteford, Carlo Vercellone, Aruel Kyrou Antonella Corsani, Enzo Rullani, Yann Moulier Boutan y Mauricio Lazzarato. Madrid: Traficantes de Sueños pp. 49-61
- East Bay Community Energy (2019) PG&E – EBCE. Joint Rate Comparisons. Disponible en https://ebce.org/wp-content/uploads/EBCE-Web-Comparison-July2019_FINAL090619.pdf (Consultado el 13/08/2019)

Eco, Umberto (2000) Tratado de Semiótica General. Barcelona: Lumen. ISBN: 84-264-01-05-8

Engels, Friedrich (1975) El papel del trabajo en la transformación del mono en hombre. En Obras Escogidas de Marx y Engels. Tomo II. Caracas/Madrid: Editorial Fundamentos. ISBN: 84-245-0159-4

eHealth (2019) Small Business Health Insurance Plan. Disponible en <https://www.ehealthinsurance.com/small-business-health-insurance?zipCode=94611> (Consultado el 26/07/2019)

e-Learning Force (2019) Office 365 LMS. Subscription Plans. Pricing. Disponible en <https://www.elearningforce.com/office-365-lms/pricing> (Consultado el 15/07/2019)

Electricity Local (2019) Berkeley, CA, Electricity Statistics. Home. States. California Berkeley Disponible en <https://www.electricitylocal.com/states/california/berkeley/> (Consultado el 14/08/2019)

Empresa de Servicios Públicos de Heredia (2019) Tarifas Eléctricas a partir del 1 de enero del 2019. Disponible en <https://www.esphsa.com/sites/default/files/inline-files/Tarifas%20Energ%C3%ADa%202019.pdf> (Consultado el 15/07/2019)

Erneroglu, Berch (2011) The dynamics of the labor process in the age of neoliberal capitalist globalization: a class analysis. International Review of Modern Sociologist. Vol 37 No 1. pp 31-52

- Escamilla Paredes, Mauricio (2014) Histéresis laboral: un estudio empírico para México 1994-2013. Ensayo para obtener el grado de especialista en econometría aplicada. Programa único de especialización en economía, división de estudios de posgrado. México: Facultad de Economía, UNAM.
- Ethereum (2021) The environmental impact of NFTs. Disponible en <https://ethereum.org/en/nft/#environmental-impact-nfts> (consultado 20/03/2021)
- Excede (2019) For Home, código postal 94611. Disponible en <https://www.excede.com/plan-results/plat100/> (Consultado 20/07/2019)
- Federici, Silvia (2010) Calibán y la Bruja. Mujeres, cuerpo y acumulación originaria. Madrid: Traficante de sueños. ISBN: 978-84-96453-51-7
- Fernández, Flory (2002) El Análisis De Contenido Como Ayuda Metodológica Para La Investigación. Ciencias Sociales Vol II No. 96: pp.35-53
- Fernández Ruiz, Jorge (2013) Teoría de juegos: su aplicación en economía. México: Colmex. ISBN: 978-607-462-091-7
- Fielding, Roy; Hann, Il-Horn; Roberts, Jef; Slughter, Sandra (sf) Why do developers contribute to open source proyectos? First evidence of economic incentives. Graduate School of Industrial Administration. Carnegie Mellon University. Pittsburgh. pp. 1-4 Disponible en <https://pdfs.semanticscholar.org/953d/9faa84cca23a9f3bc07d23eaba83acdb25fa.pdf>. (Consultado 9/06/2019)
- Freud, Sigmund (1992) El Malestar de la Cultura. En Obras Completas. Volumen 21. Paraguay: Amorrutu Editores. ISBN: 950-518-597-9

- Fuchs, Christian (2017) The Information Economy and the Labor Theory of Value. International Journal of Political Economy. Vol 46. No. 1. Pp. 65-89.
- Fuch, Christian (2015) The Digital Labour Theory of Value and Karl Marx in the Age of Facebook, YouTube, Twitter, and Weibo. En Reconsidering Value and Labour in the Digital Age. Eds. Eran Fisher y Christian Fuchs. Londres: Palgrave Macmillan. ISBN 978-1-137-47857-3 pp.26-41
- Fuchs, Christian (2014) Digital Labour and Karl Marx. New York: Routledge. ISBN 978-0-415-71615-4.
- Fuchs, Christian (2011) Cognitive capitalism or informational capitalism? The role of class in the information economy. En Cognitive capitalism, education and digital labor, ed. Michael Peters and Ergin Bulut, New York: Peter Lang. pp.75-119.
- Fuchs, Christian (2010) Labor in Informational Capitalism and on the Internet. The Information Society, 26. Pp.179–196. ISSN: 0197-2243
- Fuchs, Christian; Seignani, Sebastian (2013) What is Digital Labour? What is Digital Work? What's their Difference? And why do these Questions Matter for Understanding Social Media. tripleC. Vol 11 No.2. pp 237-293
- Fumagalli, Andrea (2010) Bioeconomía y capitalismo cognitivo. Madrid: Traficantes de Sueños. ISBN 13: 978-84-96453-54-8
- Gajst, Natalia (2010) La escuela francesa de la regulación: una revisión crítica. Revista Científica "Visión de Futuro", vol. 13, núm. 1, pp. 2-19

- García Calavia, Miguel Angel (1999) Trabajo y capital monopolista, veinticinco años después, un texto clásico todavía vigente. Cuadernos de relaciones Laborales. No 14. UCM, Madrid. pp.193-215
- García Covarrubias, Luis Fernando (2015) Flexibilidad Laboral en América Latina: Evidencia Empírica Reciente. Tesis que para optar por el grado de Maestro en Economía. México: Instituto de Investigaciones Económicas, UNAM.
- García Gallego Aurora; Miller, Luis (2011) Negociación. En Economía experimental y del comportamiento. Pablo Brañas Garza. España: Antoni Bosch editor S.A.
- Gautier, Axel; Lamesch, Joe (2020) Mergers in the Digital Economy. CESifo Working Papers ISSN 2364-1428
- Girón, Alicia (2008) Fondo Monetario Internacional: de la estabilidad a la inestabilidad. El consenso de Washinton y las reformas estructurales en América Latina. En La Globalización y el Consenso de Washinton. Comp. Gladys Lechini. Buenos Aires, Argentina: CLACSO. pp 45-60 ISBN 978-987-1183-96-8
- Gramsci, Antonio (2000) Americanismo y Fordismo. Cuaderno 22 (V) (1934) En Cuadernos de la Cárcel. México: Ediciones Era. ISBN: 968-411-461-3
- Greenspan, Alan (1998) Is there a new economy? Haas Annual Business Faculty Research Dialogue, University of California. Documento digital disponible en <https://www.federalreserve.gov/boarddocs/speeches/1998/19980904.htm>
(Consultado el 19/11/2018)
- Godbout, Todd (1993) Employment change and sectoral distribution in 10 countries, 1970-1990. Montly Labor Review

- Gordon, David; Edwards, Richards; Reich, Michael (1986) Trabajadores segmentados, trabajadores divididos. La transformación histórica en el trabajo en Estados Unidos. Ministerio de trabajo y seguridad Social, Madrid. España.
- Grenfell, Óscar (2019) Julian Assange's father warns WikiLeaks' publisher's health is "declining rapidly" in Belmarsh Prison. Word Socialist Web Site. Disponible en <https://www.wsws.org/en/articles/2019/08/20/ship-a20.html> (Consultado 9/11/2019)
- Guerrero, Diego (2011) The dependences of prices on labour-values. Revista Atlántica de Economía . Vol. 1 pp. 1-33
- Guerrero, Diego (2000) Insumo-producto y teoría del valor-trabajo. Política y Cultura, no. 13, pp. 139-168
- Guillén Romo, Héctor (2013) Las crisis. De la gran depresión a la primera gran crisis mundial del siglo XXI. México: Ediciones Era. ISBN: 978-607-445-308-9
- Gutiérrez Coto, Ilse; Kikut Valverde, Lorena; Corrales Bolívar, Karen; Picado Madrigal, Cinthya. (2018) Seguimiento de la condición laboral de las personas graduadas 2011-2013 de las universidades costarricenses. Observatorio Laboral de profesiones. San José, Costa Rica: CONARE. ISBN 978-9977-77-251-6
- Harvey, David (2006) Spaces of Global Capitalism. Towards a theory of uneven geographical development. Verso: New York and London.
- Harvey, David (1986) Making Sense of Marx by Jon Elster? Political Theory, Vol. 14, No. 4 pp. 686-690

Herrera Rodríguez, Fred (2017) La sobreproducción de capital constante en los esquemas de reproducción de Marx: el método de bifurcaciones y singularidades en el Manuscrito Económico 1861-63. Tesis para optar por el grado de licenciado en economía. Heredia, Costa Rica: Universidad Nacional de Costa Rica.

Helmslev, Louis (1971) Prolegómenos a una teoría del lenguaje. Madrid: Gredós.

Hinkelammert, Franz; Mora Jiménez, Henry (2013) Hacia una economía para la vida. Preludio a una segunda crítica de la economía política. Michoacán, México/Heredia, Costa Rica: Universidad Michoacana de San Nicolás de Hidalgo/ EUNA. ISBN: 978-607-424-388-8

Huerta Lugo, Sandra Alejandra (2017) Revoluciones del conocimiento. El trabajo y su transformación. Tesis para optar por el grado de maestra en Administración. México: Facultad de Contaduría y Administración, UNAM.

Husson, Michel (2003) Sommes-nous entrés dans le « capitalisme cognitif » ? Critique communiste n°169-170, été-automne 2003. Pp-1-8 Disponible en hussonet.free.fr/cogniti.pdf (Consultado el 4/02/2019)

IBM (2019) IBM Unveils World's First Integrated Quantum Computing System for Commercial Use. Disponible <https://newsroom.ibm.com/2019-01-08-IBM-Unveils-Worlds-First-Integrated-Quantum-Computing-System-for-Commercial-Use> (Consultado el 15/10/2019)

ImpactHub (2019) Salas de reuniones. Espacios. Disponible en <https://sanjose.impacthub.net/> (Consultado el 25/08/2019)

Internet Advisor (2019) Internet Service options in Berkeley, CA. Home. Search. Disponible en <https://www.internetadvisor.com/berkeley-california> (Consultado el 20/07/2019)

Instituto Mexicano del Seguro Social (2019a) Prestaciones y fuentes de financiamiento de los Regímenes de Aseguramiento del IMMS. Anexo A. Disponible en <http://www.imss.gob.mx/sites/all/statics/pdf/informes/20172018/22-Anexos.pdf> (Consultado el 11/08/2019)

Instituto Mexicano del Seguro Social (2019b) Seguro de Salud para la Familia. Disponible en (<http://www.imss.gob.mx/derechoH/segurosalud-familia>) (Consultado 11/08/2019)

Instituto Nacional de Estadísticas y Censos (2019) Encuesta Continua de Empleo al II Trimestre del 2019. Resultados Generales. San José, Costa Rica: INEC. ISSN: 2215-3136

Instituto Nacional de Estadística, Geografía e Información (2019a) Indicadores de ocupación y empleo. Cifras oportunas durante junio del 2019. Comunicado de prensa No. 351/19. Disponible en https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2019/iooe/iooe2019_07.pdf (Consultado el 11/08/2019)

Instituto Nacional de Estadísticas y Censos (2019b) Estadísticas a propósito de las ocupaciones relacionadas con las tecnologías de la información y de la comunicación. Datos Nacionales. Comunicado de prensa 310/19.

- iSpring (2019) Incredible Value at a Reasonable Price. Pricing. Disponible en <https://www.ispringsolutions.com/ispring-learn/pricing> (Consultado el 15/07/2019)
- Jacob, Carlos; Míguez, Pablo (2016) El trabajo cognitivo en el capitalismo contemporáneo: el surgimiento y la evolución del sector de software en Argentina y Brasil. Cuadernos del CENDES, vol. 33, núm. 93, pp. 67-89
- James, Buddy; Lalonde, Lori (2015) Pro XAML with C# Application Development Strategies. New York: Apress/Springer. ISBN 978-1-4302-6775-1
- Jing Xie & Chen-Ching Liu (2017) Multi-agent systems and their applications, Journal of International Council on Electrical Engineering, 7:1, 188-197, DOI: 10.1080/22348972.2017.1348890
- Jola Sánchez, Andrés (2013) Desarrollo del sector servicios y su papel en la consolidación del crecimiento económico mundial. Ecos de Economía, vol. 17, núm. 36. pp. 43-68
- Juanf (2010) Cálculo exacto de una fuente de alimentación. EOL. Foros. PC. Hardware. Disponible en https://www.elotrolado.net/hilo_calculo-exacto-de-una-fuente-de-alimentacion_1499626 (Consultado el 9/08/2019)
- Kasim, N. N. M.; Khalid, F (2016) Choosing the Right Learning Management System (LMS) for the Higher Education Institution Context: A Systematic Review. iJET – Volume 11, Issue 6, <http://dx.doi.org/10.3991/ijet.v11i06.5644> pp. 55- 61
- Kelly, Diana (2016). Perceptions of Taylorism and a Marxist scientific manager. Journal of Management History, Vo 22. No. 3 , pp298-319.

- Kendall, Julie; Kendall, Kenneth; Germonprez, Matt (2016) Game theory and open source contribution: Rationale behind corporate participation in open source software development. Journal of Organizational Computing and Electronic Commerce. Vol 26. No 4. pp 323-343 ISSN: 1091-9392
- Kim, Changkeun (2010) The Recent Controversy on Marx's Value Theory. A Critical Assessment. Institute for Social Science, Gyeongsang National University. Pp 282-320 Documento digital disponible en <http://gesd.free.fr/kim2010.pdf> (Consultado 6/10/2019)
- Kleiner, Dymitri (2010) The Telecommunist Manifesto. Amsterdam, Netherlands: Network Netbook Series. ISBN 978-90-816021-2-9
- Knappenberger, Brian (2014) The Internet's Own Boy: The Story of Aaron Swartz. Disponible en <https://youtu.be/9vz06QO3UkQ> (Consultado el 26/08/2019)
- Kölbi (2019) Internet Hogar sobre Fibra óptica. Disponible en https://www.kolbi.cr/wps/portal/kolbi_dev/hogares/internet/internet-individual-fibra (Consultado el 20/08/2019)
- Kott, Alexander (2018) Intelligent Autonomous Agents are Key to Cyber Defense of the Future Army Networks. The Cyber Defense Review . Vol. 3, No. 3. pp. 57-70
- Krueger, Anne (1999) Journal of Economic Perspectives. Vol 13, No. 3. Pp 153–156
- Kundu, Ambar; Swain, Debashish; Biswas, Sitanath (2010) Application of free and Open source software and its Impact on society. International Journal of Computer Science and Information Technologies, Vol. 1 No.4 , pp.226-229. ISSN: 0975-9646

- Kuster, Christian; Kûster, Tobias; Lützenberger, Marco; Albayrak, Sahin (2014)
Model-Driven Development and Validation of Multi-Agent Systems in JIAC V
with the Agent World Editor. Procedia Computer Science. Vol 32 p.p. 920 –
927
- Kyrou, Ariel (2004) Elogio del plagio. El sampling como juego o acto artístico. En
Capitalismo cognitivo, propiedad intelectual y creación colectiva. Ed. Emmanuel
Rodríguez López y Raúl Sánchez Cedillo. Madrid: Traficante de Sueños. ISBN:
84-933555-0-X
- Lara Sánchez, Miguel Angel (2012) Del fordismo a la revolución actual del proceso de
trabajo y de sus respectivas condiciones generales de la reproducción. Tesis para
obtener el grado de Doctor en Economía. México: Facultad de Economía,
UNAM.
- Lee, Sangheon; McCann, Deirdre; Messenger, Jon (2007) Working Time Around the
World. Trends in working hours, laws and policies in a global comparative
perspective. New York: Routledge.
- Lerner, Josh; Tirole, Jean (2002) The simple economics of open source. The journal of
industrial economics. Vol. 50, no.2 pp 197-234
- Lessing, Lawrence (2004) Prólogo. En Software libre para una sociedad libre. Richard
Stallman. Madrid: Traficantes de sueños. ISBN: 84-933555-1-8 pp. 11-15
- Liao, Shannon (2019) Chinese developers use GitHub to protest long work hours. The
Verge. Disponible en <https://www.theverge.com/2019/4/2/18291035/chinese-developers-github-protest-long-work-hours> (Consultado el 9/04/2019)

- Lipietz, Alain (1997) El mundo del Postfordismo. Ensayos de Economía n°12, vol.7.
Pp- 11-53
- Liviu, Mihai (2014) Comparative study on software development methodologies.
Database Systems Journal. Vol 3. pp.37-56
- López Leyva, Santos; Mungaray Montezuna, Ana; Pérez, Silvia (2015) Knowledge-based Economy in Argentina, Costa Rica and Mexico: A comparative analysis from the Bio-Economy Perspective. Vol.3 no.2, pp.213-236. ISSN 2392-8042
- Lukacs, George (1978) El marxismo ortodoxo y el materialismo histórico. México: Editorial Grijalbo. ISBN 968-419-046-8
- Lula, Pawel; Madej, Jan; Stal, Janusz; Tadeusewicz, Ryszard; Tuchowski, Janusz (2014) Programming Methods and Tools. Cracovia, Polonia: Cracow University of Economics . ISBN 978-83-64509-07-0
- MacCormack, Alan; Rusnak, John ; Baldwin, Carliss (2006) Exploring the Structure of complex software Designs: an empirical study of open source and proprietary code. Harvard Business School Working Paper No. 05-016. Pp.1-39
- McConnell, Steve (1998) The Art, Science, and Engineering of Software Development. IEEE Software. Vol. 18, No. 02, Pp.9-11
- Mandel, Ernest (1986) Las Ondas largas del desarrollo capitalista. La interpretación marxista. Madrid: Siglo XXI.
- Mankiw, Gregory (2014) Macroeconomía. España: Antoni Bosch Editor. ISBN: 978-84-95348-94-4

- Mankiw, Gregory (2013) Defending the one percent. Journal of Economic Perspectives. Volume 27, Number 3, pp 21–34
- Mao, Rui; Xu, Jianwei ; Zou, Jingxian (2018) The labor force age structure and employment structure of the modern sector. China Economic Review 52 pp. 1–15
- Marazzi, Christian (2014) Capital y lenguaje. Hacia el gobierno de las finanzas. Buenos Aires: Tinta Limón. . ISBN 978-987-27390-9-6
- Marcuse, Heberth (1969) Marx y el trabajo alienado. Buenos Aires: Carlos Pérez Editor
- Martínez González, Gloria; Valle Baeza, Alejandro; Sánchez, Cesar (2019) Productividad y tasa de plusvalor a nivel internacional: evaluación empírica. Revista Problemas del Desarrollo. Vol 196. No.50
- Maruping, Likoebe; Venkatesh, Viswanath; Agarwal, Ritu (2009) A control theory Perspective on Agile Methodology Use and Changing User Requirements. Information Systems Research Vol 20 no 3 pp 377-399 ISSN 1047-7047 Doi 10.1287
- Marx, Karl (1984) El Capital. Crítica de la economía política. Tomo I. El proceso de producción. México: Siglo XXI Editores.
- Melucci, Alberto (1999) Acción colectiva, vida cotidiana y democracia. México: COLMEX. ISBN: 978-9681-208-691

- Microsoft (2019a) Almacenamiento Azure. Productos. Disponible en <https://azure.microsoft.com/es-es/pricing/calculator/>?(Consultado el 8 de julio del 2019)
- Microsoft (2019b) Comprar Windows 10. Store Disponible en <https://www.microsoft.com/es-mx/store/b/windows> (Consultado el 08 /07/ 2019)
- Microsoft (2019c) Precios de Visual Studio. Suscripción Estándar. Disponible en <https://visualstudio.microsoft.com/es/vs/pricing-details/> (Consultado el 08/07/ 2019)
- Ministerio de Trabajo y Seguridad Social (2019) Salarios mínimos para el sector privado. Departamento de Salarios, Costa Rica. Decreto No. 41434-MTSS. Disponible en http://www.mtss.go.cr/temas-laborales/salarios/Documentos-Salarios/Lista_Salarios_2019.pdf (Consultado el 13/08/2019)
- Mochi Alemán, Óscar (2006) La industria del software en México en el contexto internacional y latinoamericano. Morelos, México: UNAM. ISBN: 970-32-3095-4
- Modonesi, Máximo; García Vela, Alfonso Galileo; Vignau Loría, María (2017) El concepto de clase social en la teoría marxista contemporánea. México: UNAM. ISBN: 978-607-02-9370-2
- Modonessi, Máximo (2010) Subalternidad, antagonismo, autonomía: marxismos y subjetivación política. Buenos Aires, Argentina: CLACSO/ Prometeo Libros. ISBN 978-987-1543-46-5
- Montusci, Luisa (1999) Cambios en la estructura del empleo: de la industria a los servicios. CEMA. Documento de Trabajo No. 131 p.p 1-22 Disponible en

<https://ucema.edu.ar/publicaciones/download/documentos/127.pdf> (consultado el 19/02/2019)

Montuschi, Luisa (1998). Cambios en la estructura del empleo: de la industria a los servicios. CEMA Documento de Trabajo No. 127 pp.1-22 Disponible en <https://ucema.edu.ar/publicaciones/download/documentos/127.pdf>. (Consultado el 14/01/2019)

Moodle (2019) Moodle en Azure. Página Principal, Comunidades, Moodle en Español. Tema 1. Instalación y Actualización Disponible <https://moodle.org/mod/forum/discuss.php?d=171373> (consultado 14/07/2019)

Morandé, Felipe (2016) A casi tres décadas del Consenso de Washington ¿Cuál es su legado en América Latina? Estudios Internacionales. No 185. pp-31-58 - ISSN 0716-0240

Morini, Cristina (2007) The Feminization of Labour in Cognitive Capitalism. Feminist Review, No. 87, Italian Feminisms pp. 40-59

Moseley, Fred (2016). La MELT y el Razonamiento circular en la Nueva Interpretación y en la Interpretación Temporal y de Sistema Único. Massachussets : Mount Holyoke College. Pp. 1-19 Documento digital disponible en: <https://marxismocritico.com/2016/07/13/la-melt-y-el-razonamiento-circular-en-la-nueva-interpretacion-y-en-la-interpretacion-temporal-y-de-sistema-unico/> (consultado 9/11/2019)

Moulier Boutang, Yann (2012) La abeja y el economista. España: Traficantes de Sueños. ISBN 13: 978-84-96453-69-2

- Moulier Boutang, Yann (2004) Riqueza, propiedad, libertad y renta en el capitalismo cognitivo. En Capitalismo cognitivo, propiedad intelectual y creación colectiva. Ed. Emmanuel Rodríguez López y Raúl Sánchez Cedillo. Madrid: Traficante de Sueños. ISBN: 84-933555-0-X
- Muffato, Moreno (2006) Open Source: A multidisciplinary Approach. Series on technology management Vol 10. Londres: Imperial College Press ISBN 1-86094-665-8
- Murillo Tsijli, Manuel (2007) Introducción a la matemática discreta. Cartago, Costa Rica: Editorial Tecnológico de Costa Rica. ISBN: 978-9977-66-193-3
- Negri, Antonio (2004) Los libros de la autonomía obrera. Antagonismo, organización, comunismo: hipótesis para la nueva política del sujeto hiperproletario global. Madrid: Akal Ediciones.
- Negri, Antonio; Lazzarato, Mauricio (2001) Trabajo inmaterial y producción de subjetividad. Río de Janeiro: DP&A Editora
- Northover, Mandy; Kourie, Derrick; Boake, Andrew; Gruner, Stefan; Northover, Alan (2008) Toward a philosophy of Software Development: 40 years after de birth of Software Engineering. Journal for General Philosophy of Science. Vol 30. No. 1 pp.85-113
- OECD (2018) Trade union density in OECD countries. Visualizador de Datos disponible en (<https://stats.oecd.org/index.aspx/Index.aspx?DataSetCode=TUD>) (Consultado 1/11/2018)
- Olson, Mancu (1971) The logic of collective action: Public goods and theory of groups. Cambridge, Massachusetts: Harvard University Press.

- Outervision (2019) Power Supply Calculator. Basic. Disponible en <https://outervision.com/power-supply-calculator> (Consultado el 9/08/2019)
- Ouadoud , Mohammed; Yassin Chkouri, Mohamed; Nejjari, Amel (2018) Learning Management System and the Underlying Learning Theories: Towards a new Modeling of an LMS. International Journal of Information Science & Technology –iJIST, ISSN: 2550-5114. Vol.2 No. 1. pp-25-33
- Pacific Workplaces (2019) NEXTSPACE COWORKING BERKELEY Netspace coworking Berkeley. Locations. Disponible en <https://pacificworkplaces.com/office-space/> (Consultado el 11/08/2019)
- Pearl, Judea; Mackenzie, Dana (2018) The book of why: the new science of cause and effect. New York: Basic Books. ISBN: 978-0-465-09761-6
- Pereyra, Carlos (1984) El sujeto de la historia. Madrid: Alianza Editorial.
- Piketty, Thomas (2014) El Capital en el siglo XXI. Cambridge, Massachusetts: Harvard University Press. ISBN 978-0-674-43000-6
- Pratten, Clifford (1980) Journal of Economic Literature, Vol. 18, No. 1. pp. 93-96
- Poulova, Petra; Simonova, Ivana; Manenova, Martina (2015) Which One, or Another? Comparative Analysis of Selected LMS. Social and Behavioral Sciences No. 186 pp. 1302 – 1308 doi: 10.1016/j.sbspro.2015.04.052
- Quiñones Orjuela, Edwin (2011) Moodle en Azure. En Actualización e instalación. Tema 1. Comunidades. Blog web disponible en <https://moodle.org/mod/forum/discuss.php?d=171373> (Consultado el 13/07/2019)

Ramos Martinez, Alejandro (1995) The Monetary Expression of Labour: Marx's twofold Measure of Value. Submitted to the 1995 mini-conference of the IWGVT, New York.

RapidTables (2019) Energy Consumption Calculator. Home. Calculators. Electrical Calculators. Disponible en <https://www.rapidtables.com/calc/electric/energy-consumption-calculator.html> (Consultado el 9/08/2019)

Raymond, Erick Steven (2001) The Cathedral and The Bazaar: musings on Linux and open source by an accidental revolutionary. California: O'Reilly & Associates, Inc.

Regus (2019a) Downtown Berkeley. 2001 Adisson Street, Suite 300, Berkeley, California, 94704. Home. United States. Disponible en <https://www.regus.com/offices/united-states/california/berkeley/office-space/california-berkeley-downtown-berkeley> (Consultado el 15/08/2019)

Regus (2019b) Fuente Bella 3299, Piso 14, Conjunto Residencial del Pedregal, Ciudad de México, Ciudad de México, 14130. Disponible en <https://www.regus.com.mx/office-space> (Consultado el 15/08/2019)

Regus (2019c) Centro Corporativo Plaza Roble, Edificio 5, San José, San José Province, 10203. Disponible en <https://www.regus.co.cr/office-space/costa-rica/san-jose-province/san-jose/san-jose-plaza-roble-las-terrazas> (Consultado el 15/08/2019)

Rengifo Ariza, Luis Eduardo (2017) La teoría del valor y sus manifestaciones contemporáneas. Contexto. No 6 pp 83-92 ISSN: 2346-0784

- Riemer, Kai; Johnston, Robert (2011) Artifact or Equipment? Rethinking the Core of IS using Heidegger's ways of being. Conference paper. Shanghai, China:Thirty Second International Conference on Information Systems.
- Rieznik, Pablo (2014) ¿Qué es la teoría del derrumbe del capitalismo? (Y cómo son las cosas). HICR HODUS. C risis capitalista, polémica y controversias. No. 6. pp. 1-16
- Robbins, Lionnel (1932) Essay on the nature and significance of economic science. London: McMillan& and Co. Limited.
- Robinson, Joan (1969) The economics of imperfect competition. Manhathan, New York: St. Martin Press.
- Robinson, Thomas (1970) Game Theory and Politics: Recent Soviet Views. Studies in Soviet Thought, Vol. 10, No. 4 pp. 291-315
- Rodríguez Herrera, Adolfo (2008) Trabajo y dinero. Un siglo de controversia. Heredia, Costa Rica: EUNA. ISBN: 978-9977-65-304-4.
- Roemer, John (1982) Methodological individualism and deductive Marxism. Theory and Society, Vol 11, No 4. Pp 513-520
- Romero Amado, Jorge; López Toache, Vania; Martínez De Ita, María (2018) El capital humano en México en el marco de la economía basada en el conocimiento: una revisión a los indicadores de ciencia y tecnología. Economía coyuntural. Vol.3, núm. 1, pp. 1 - 36.
- Ross, Daniel (2013) The place of Free and Open Source Software in the Social Apparatus of Accumulation. Science and Societe. Vol 77. No 22 pp.202-226

- Sánchez Rodríguez, José (2009) Plataformas de enseñanza virtual para entornos educativos. Pixel-Bit. Revista de Medios y Educación, núm. 34. pp. 217-233
- Sandoval Manzano, José Guadalupe (2007) Notas sobre crecimiento económico. Tesis para optar por el grado de maestro en economía. México: Facultad de Economía, UNAM.
- Schmidt, Tobias Sebastian; Chahin, Abdo; Kößler, Johannes; Paetzold, Kristin (2017) Agile development and the constraints of physicality: A network theory base caused and effect analysis. In: Proceedings of the 21 st International Conference on Engineering Design (ICED17), Vol. 4: Design Methods and Tools, Vancouver, Canada.
- Simon, Rebecca; Klowden, Kevin; Karo Schulman, Carolyn (2018) Future profiting the video game industry in California. Santa Mónica, California/ Washintong DC/ Londres/ Singapore: Milken Institute. Documento digital disponible en <https://milkeninstitute.org/sites/default/files/reports-pdf/Entertainment-Software-Association-Report.pdf> (Consultado el 11/09/2019)
- Smith, Adam (2016) La teoría de los sentimientos morales. Madrid: Alianza Editorial. ISBN: 978-84-206-7666-1
- Solvo (2019) Desarrollo web y software a la medida. Contáctenos. Disponible en www.solvosoft.com (Consultado el 10/12/2019)
- Sommerville, Ian (2011) Software Engineering. Massachusets, EE.UU: Pearson. ISBN 13: 978-0-13-703515-1

- Steinwold, Andrew (2019) The History of Non-Fungible Tokens (NFTs) Disponible en <https://medium.com/@Andrew.Steinwold/the-history-of-non-fungible-tokens-nfts-f362ca57ae10> (Consultado el 20/03/2021)
- Stallman, Richard (2004) Software libre para una sociedad libre. Madrid: Traficante de Sueños. ISBN: 84-933555-1-8
- Stallman, Richard (1985) El Manifiesto GNU. Fundación GNU. Archivo digital disponible en <https://www.gnu.org/gnu/manifesto.html> (Consultado el 14/09/2019)
- Starosta, Guido (2012) Cognitive commodities and the Value-Form. Science and Soceite Vol 76 no. 3. pp 365-392
- Stiroh, Kevin (1999) Is There a New Economy? Challenge, Vol. 42, No. 4 pp. 82-101
- Strong, E.D. (sf) How Many Watts Does a Computer Use? Home. Around The Home. Productivity. Disponible en <https://www.techwalla.com/articles/how-many-watts-does-a-computer-use> (Consultado el 9/08/2019)
- Tabarés Gutiérrez, Raúl (2012) El inicio de la Web: historia y cronología del hipertexto hasta HTML 4.0 (1990-99) Artefactos, vol. 5, n.o 1, pp. 57-82
- Talent LMS (2019) Standard plans. Prices. Disponible en <https://www.talentlms.com/prices> (Consultado el 15/07/2019)
- Thatcher, Margareth (1993) The years of Downing Street. United Kingdon: Aguilar. Disponible en https://www.larazon.es/historico/5483-asi-derroto-thatcher-a-los-sindicatos-FLLA_RAZON_439562 (Consultado el 31/01/2019)

- Thompson, Paul; Smiht, Chris (2017) Capital and the Labour Process. En Reading Capital Today: Marx After 150 Years. Forthcoming Ingo Schmidt & Carlo Fanelli (eds.) (2017) London: Pluto Press pp. 116-137 ISBN: 9781786800855
- Tigo (2019) Internet de otro nivel. Internet. Disponible en <https://www.tigo.cr/tigo-home/internet> (Consultado el 20/08/2019)
- TotalPlay (2019) Planes DoblePlay, CDMX. Disponible en <https://www.totalplay.com.mx/landings/cdmx> (Consultado el 20/08/2019)
- Trigo Aranda, Vicente (2004) Historia y evolución de internet. Manual formativo de ACTA No.33 ISSN: 1888-6051 pp 22-32
- Tse-Tung, Mao (1968) Sobre la Contradicción. En Obras Escogidas. Tomo I. Pekin: Ediciones en Lenguas Extranjeras. Pp-333-370
- Turing, Alan (1950) Computing Machinery and Intelligence. Mind, New Series, Vol. 59, No. 236 pp. 433-460
- Urcioli, Bonnie; Ladousa, Chaise (2013) Language Management/Labor. Annual Review fo Anthropology. Vol 42. pp-175-190
- Valle Baeza, Alejandro (1996) Productividad y salarios reales. En Los salarios de la crisis. Eds. Alejandro Valle Baeza y Gloria Martínez González. México: Desarrollo de Medios. ISBN: 968-6719-51-2
- Vásquez, Jose Luis; Garcaí, María Purificación (2017) From taylorism to neo-taylorism: a 100 year journey in human resource management. En Menedzsment innovációk az üzleti és a nonbusiness szférákban. Ed. Vilmányi Márton y Kazár Klára. Hungría: Universidad de Szeged. ISBN:9789633065440

- Vendome, Christopher; Bavota, Gabriele; Di Penta, Massimiliano; Linares-Vásquez, Mario; German, Daniel; Poshyvanyk, Denys (2016) Licence usage and changes: a large-scale study on gitHub. New York: Springer Science+Business Media. pp. 1537-1575
- Vento (2011) Calculadora De Watts. Índice del foro. Hardware. Fuentes de poder. Disponible en <https://www.ozeros.com/foro/topic/238-calculadora-de-watts/> Consultado el (Consultado el 9/08/2019)
- Veraza, Jorge (2007) Leer El Capital hoy. Pasajes selectos y problema decisivos. México: Editorial Ítaca y Ediciones Paradigmas y Utopías. ISBN: 968-7943-79-3
- Vercellone, Carlo (2004) Las políticas del desarrollo cognitivo del capitalismo cognitivo. En Capitalismo cognitivo, propiedad intelectual y creación colectiva. Ed. Emmanuel Rodríguez López y Raúl Sánchez Cedillo. Madrid: Traficante de Sueños. ISBN: 84-933555-0-X
- Voloshinov, Valentin (2009) El marxismo y la filosofía del lenguaje. Buenos Aires, Argentina: Ediciones Godot. ISBN: 978-1489-11-4
- Vygotsky, L. (1995, primera [1934]). Pensamiento y Lenguaje. Buenos Aires, Argentina: Fausto.
- We-Work (2019) Ciudad de México. Todas las oficinas. Disponible en <https://www.wework.com/es-LA/l/mexico-city--DIF> (Consultado el 15/08/2019)
- Wren, Daniel (1980) Scientific Management in the U.S.S.R., with Particular Reference to the Contribution of Walter N. Polakov. The Academy of Management Review, Vol. 5, No. 1, pp. 1-11

- Wright, Erik Olin (1996) The Continuing Relevance of Class Analysis – Comments. Theory and Society. Vol. 25, No. 5. pp. 693-716
- Xing, Minqing (2014) The impact of commercial open source software on proprietary software producers and social welfare. Journal of Industrial Engineering and Management. Vol 7. No. 5. pp.1183-1196 – Online ISSN: 2013-0953
- Ye, Yunwen; Nakakoji, Kumiyo; Yamamoto, Yasuhiro (2008) Measuring and Monitoring Task Couplings of Developers and Development Sites in Global Software Development . En Software Engineering Approaches for Offshore and Outsourced Development. K. Berkling et al. (Eds.). Alemania: Springer pp. 181–195 ISSN 1865-1348
- Yoav, Shoham; Leyton Brown, Kevin (2008) Essentials of Game Theory. EE.UU: Morgan & Claypool. ISBN: 9781598295948
- Yu, Huang (2019) Robot Threat or Robot Dividend? A Struggle between Two Lines. En Dog Days. Made in China Yearbook 2018. Eds. Ivan Franceschini, Nicholas Loubere, Kevin Lin, Elisa Nesossi, Andrea E. Pia, Christian Sorace. Australia :ANU Press. pp.54-59. ISBN 978-1760-462-93-2
- Zapata Lillo, P (2013) Economía, política y otros juegos. Una introducción a los juegos no cooperativos. México: UNAM. ISBN: 978-970-32-4706-6
- Zayout, Iyad; Hajjdiab, Hassan (2013) How much integrated development environments (IDEs) improve productivity? Journal of software, vol 8, no 10 pp2425-2432
- Zmud, Robert (1980) Management of large software development efforts. MIS Quarterly, Vol. 4, No. 2. pp. 45-55

Zorzoli, Luciana (2018). ¿Una teoría marxista sobre los sindicatos? Balance con inventario para pensar las organizaciones obreras. *Sociohistorica*, 41, e046.
<https://doi.org/10.24215/18521606e046> pp.1-20

Anexo 1: **Evidencia documental de una práctica colaborativa en una comunidad de programación libre y abierta. Secuencia de tres capturas, 18 marzo -13 diciembre 2011**

 **Moodle en Azure**
Edwin Jair Quiñones Orjuela
viernes, 18 de marzo de 2011, 23:43

Buenos Días.

Estoy integrando Moodle en la nube (Azure) con un paquete que creo codeplex para subirlo a la nube, este genera un mensaje (The service is unavailable.) al comenzar la instalación, pero de ahí no pasa y dice que genera un error en la BD pero despues de un tiempo de que el genere procesos en segundo plano funciona, esto no es una solución muy buena para alguien que necesita instalar moodle y ponerlo a punto en 3 días, mi pregunta es la siguiente, por casualidad ustedes no han hecho pruebas de subir moodle en azure, para ver si me dan una luz o como generar la instalación que no sea con el paquete de codeplex.

de antemano,

Muchas Gracias

 **Re: Moodle en Azure**
Edwin Jair Quiñones Orjuela
sábado, 19 de marzo de 2011, 01:22

Listo, la logre, si necesitan alguna ayuda sobre esto, no duden en preguntarlo.

Vemos

Promedio de calificaciones: -

[Enlace permanente](#) [Mostrar mensaje anterior](#) [Responder](#)

 **Re: Moodle en Azure**
David Hernández
sábado, 19 de marzo de 2011, 01:24

Hola Edwin,

¿Por qué no narras tu experiencia por aqui?

Nunca está de más dejar un registro de las soluciones encontradas individual o colectivamente.

Después de todo, para eso somos una comunidad ¿no? 😊

Saludos

 **Re: Moodle en Azure**
Pablo Sousa
martes, 13 de diciembre de 2011, 16:37

Hola,

Estoy intentando desplegar Moodle en Azure siguiendo los pasos que vienen en codeplex pero al desplegarlo me aparece el siguiente error:

```
One or more configuration settings are specified for this deployment configuration but are not defined in the service definition file: Moodle2_WebRoleInstances, Moodle2_WebRole:AzureCloudStorageAccountName, Moodle2_WebRole:AzureCloudStorageAccountKey, Moodle2_WebRole:driveconnection, Moodle2_WebRole:MountDrives, Moodle2_WebRole:MoodleDBHost, Moodle2_WebRole:MoodleDBName, Moodle2_WebRole:MoodleDBUser, Moodle2_WebRole:MoodleDBPass, Moodle2_WebRole:MoodleDBTablePrefix, Moodle2_WebRole:MoodleWWWRoot, Moodle2_WebRole:MoodlePasswdSaltMain, Moodle2_WebRole:MoodleTimezone, Moodle2_WebRole:MoodleDebugLevel, Moodle2_WebRole:DisplayErrors.
```

Alguna idea??

Muchas gracias

Promedio de calificaciones: -

[Enlace permanente](#) [Mostrar mensaje anterior](#) [Responder](#)

Fuente: Moodle (2019)

Anexo 2: Tecnología en la producción de alfileres en Francia 1760 y en Inglaterra en 1832 según Charles Babbage

La idea de que la intensidad del proceso de trabajo se puede medir como un ratio proviene de Charles Babbage (2009) quien realiza dos esfuerzos en este sentido. En primer lugar, registra minuciosamente la información del ilustre ejemplo de la fábrica de alfileres que menciona Adam Smith, pero afirma que en realidad fue presentada M Perronet en Francia en 1760 en una tabla que resumen las tareas en las que era dividía el proceso de trabajo en ese momento, así como el tiempo que se tardaba en cada una de ellas, con el objetivo de producir doce mil alfileres número 6 en 24.3 horas de trabajo en total y lo compara con los datos que puede obtener para la industria inglesa de 1832 para la producción de una onza de alfileres “elevens” en 7.64 horas de trabajo (2009:145)

Esta información se resume en la tabla siguiente, donde se muestra que para 1760 la división del trabajo incluía once tareas, mientras que para 1830 se dividía en siete tareas.

Tabla 8: División del trabajo en la industria de alfileres, Francia 1760 e Inglaterra 1830

Número de tarea	Francia, 1760		Inglaterra, 1832	
	12000 alfileres No. 6	Horas	12000 alfileres “elevens” (2.164 libras)*	Horas
T1	Alambre	-	Trefilado	0,787
T2	Estirado y corte	1,2	Enderezar el cable.	1,298
T3(a)	Apuntalado grueso	1,2	Apuntalar	0,649
T3(b)	Rueda giratoria	1,2		
T3(c)	Apuntalado fino	0,8		
T3(d)	Rueda giratoria	1,2		
T3(c)	Cortar los extremos puntiagudos	0,6		
T4(i)	Girado en espiral	0,5	Torcer y cortar las	0,173

T4(ii)	Cortar las cabezas	0,8	cabezas	
T5	Poner las cabezas	12	Poner las cabezas	8,656
T6	-	-	Estaño blanqueo ^o	0,464
T7	Empapelado	4,8	Empapelado	4,612
	Total	24,3	Total	16,639

Fuente: Babbage(2009: 148)

*Editado para que la cantidad de alfileres sea idéntica e intentar algunas comparaciones.

En la tabla ocho se presenta en la primera columna una numeración de la tareas con base en las siete que identifica Babbage para Inglaterra en 1832, y a la derecha de cada tarea, la cantidad de horas de trabajo que registra el autor. Cabe aclarar que en Francia, 1760, los datos se refieren a producir doce mil alfileres número 6 mientras que los datos originales son para una libra de alfileres “elevens”.

Para volver comparativos estos datos, se siguen las pistas de Caple (1992: 247), quien encuentra que los alfileres número 6 en Francia y los “elevens” en Inglaterra tenían las mismas dimensiones, en tamaño y masa, por lo que para poder compararlos se ha extrapolado los datos de 1832 en Inglaterra, que se refieren 5546 alfileres "elevens" que pesan una libra, de tal manera que, y con las limitaciones del caso, entre 1760 y 1832 la cantidad de horas de trabajo requeridas para producir doce mil alfileres se redujo de 26.3 a 16.639, medición que podría interpretarse como un aumento de la productividad debido cambios en la base tecnológica⁵⁰.

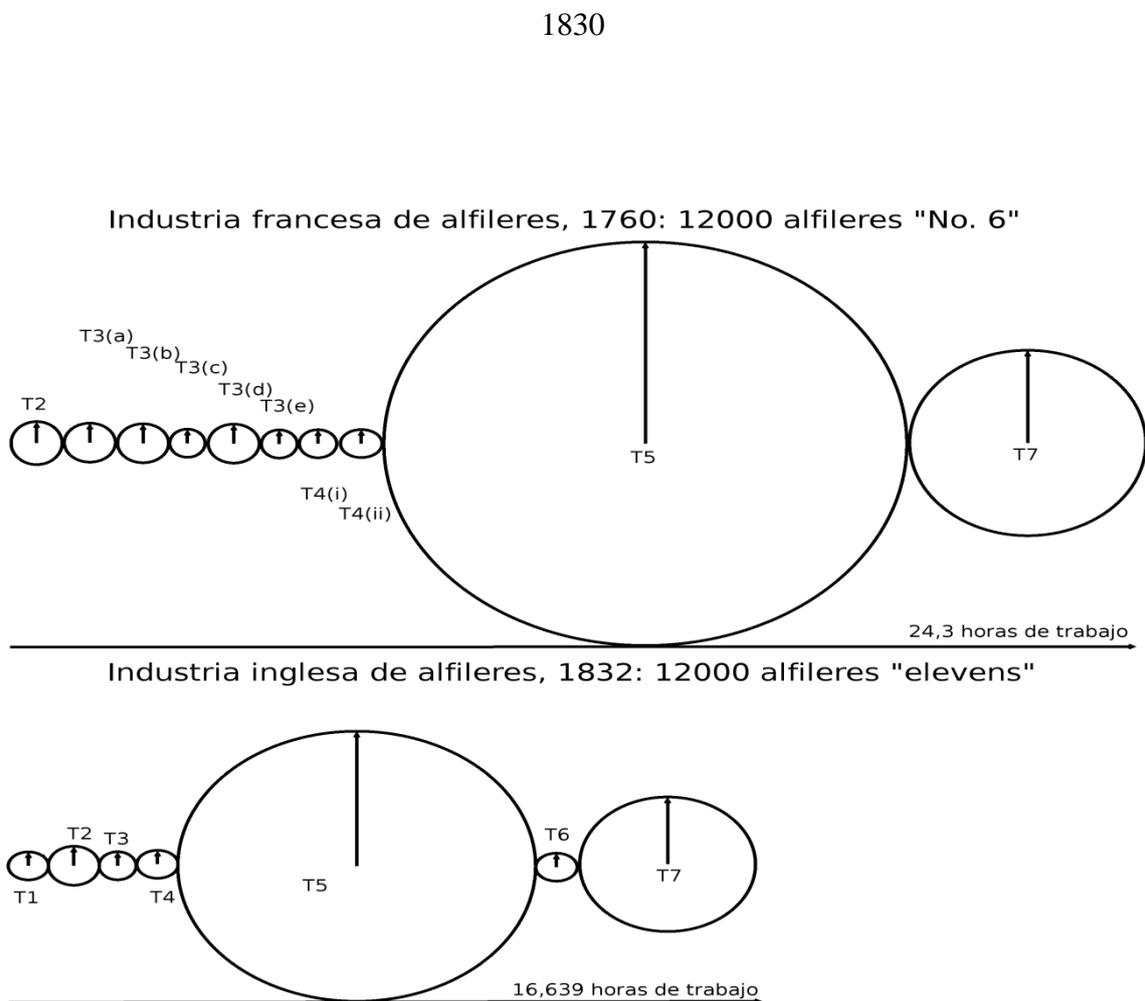
En segundo lugar, Babbage propone que la mayor parte de las ocupaciones son una series de repetición del mismo hecho o tarea, y refiere varios mecanismos para

⁵⁰ Según Pratten (1980: 95) la utilización de ciertas maquinarias permitió que la industria inglesa de alfileres produjera en 1830 un total de 45 alfileres por minuto, mientras que para 1900 se producían 180 por minuto y ya para 1980 se producían 500 por minuto, es decir, doce mil en veinticuatro minutos.

registrar tareas repetitivas que podrían servir, por ejemplo, para contar el número de vueltas de una rueda para saber la distancia recorrida (2009: 39), semejante al mecanismo que poco más de cien años describe Turing (1950: 439).

Así, la intensidad del proceso de trabajo se puede definir como un radio, siguiendo la comprensión y los datos aportados por Babbage, si se toma la información de la tabla ocho y se representa de manera angular como en la ilustración 1.

Ilustración 1: Intensidad del trabajo en la industria de alfileres, Francia 1760 e Inglaterra



Fuente: Elaboración propia con datos de Babbage (2009: 145)

En la ilustración 1 se representan cada una de las tareas en los que era dividido el proceso de trabajo para producir alfileres. El diámetro de cada círculo guarda una

proporción con el resto de los círculos acuerdo con los datos aportados por Babbage, de tal forma que si suman los diámetros de todos los círculos, se obtiene el total de hora de trabajo requeridas.

De esta manera, siguiendo la intuición de Babbage, cada vez que el radio indicado dentro de cada círculo completa una revolución, es decir, 2π radianes, en dirección de las manecillas del reloj la tarea habría sido completada, o al menos, el tiempo destinado para realizar dicha tarea se habrá agotado. Esto quiere decir que entre más mayor es el diámetro, más horas de trabajo se requieren para realizar la tarea, y por tanto la cantidad de veces que se realiza dicha tarea por hora de trabajo es menor que aquella cuyo diámetro sea menor.

Por consiguiente, y considerado que el radio es la mitad de la diámetro, se puede afirmar que, dada una base tecnológica, la intensidad del proceso de trabajo se puede medir por la magnitud del radio que caracteriza la división del trabajo de dicho proceso: acortar los radios de cada tarea implica aumentar la cantidad de veces que se ejecuta una tarea por hora de trabajo y viceversa.