



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES ACATLÁN

**Diseño e implementación de algoritmos para
manipular frecuencias e ingreso de elementos
en una aplicación para aprender LSM
(Lengua de Señas Mexicana).**

TESINA

QUE PARA OBTENER EL TÍTULO DE
LICENCIADO EN MATEMÁTICAS APLICADAS Y COMPUTACIÓN

PRESENTA:
Charbel Solís Beristáin

DIRECTORA DE TESINA:
Dra. Marícarmen González Videgaray



Santa Cruz Acatlán, Naucalpan, Estado de México, 2021



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.

Agradecimientos

A mi familia, a mi padre por su respaldo y ejemplo, a mi madre por su insistencia y amor, a mi hermano, por su bondad y comprensión. Gracias.

También les quiero agradecer a todas esas personas que aparecieron en el camino de mi formación; profesores, amigos, familiares, en especial a mi tutora por su siempre dispuesta y amable atención.

Contenido

Introducción.....	5
Capítulo I: Lengua de señas.....	8
1.1 ¿Qué es la lengua de señas?.....	8
1.2 ¿Por qué elegí la lengua de señas?.....	8
1.3 Alcances de la aplicación en el campo de la LS.....	11
Capítulo II: Game-Based Learning.....	14
2.1 Juegos y aprendizaje.....	14
2.2 Tipos de video juegos.....	16
2.3 Aportación:.....	18
2.4 Aplicaciones existentes.....	20
Capítulo III: Prototipos.....	31
3.1 Prototipo 1: Tlahtolli.....	31
3.2 Prototipo 2: Maimachiotl.....	35
3.3 Prototipo 3: Mainmachiome.....	40
Capitulo IV: Algoritmos.....	48
4.1 Simulación.....	51
4.2 Algoritmo 1: definición de probabilidades.....	58
4.3 Algoritmo 2: Modificación de frecuencias instantáneas de una selección de ítem.....	61
4.4 Algoritmo 3: Selector de nuevos ítems.....	77
4.5 Algoritmo 4: Modificación de frecuencias con base a la probabilidad de cada ítem.....	91
Conclusiones.....	96
Referencias.....	99
Bibliografía.....	99

Ilustraciones

Ilustración 1. Lugar de residencia por tipo de lengua; lengua de señas (Holcomb, 2017, p. 24).....	14
Ilustración 2. Icono de la aplicación.....	22
Ilustración 3. Menú principal (Morales, 2018).....	23
Ilustración 4. Sección de glosario (Morales, 2018).....	23
Ilustración 5. Sección de juego (Morales, 2018).....	24
Ilustración 6. Sección de examen (Morales, 2018).....	24
Ilustración 7. Sección Examen, desbloqueada (Morales, 2018).....	25
Ilustración 8. Sección examen, trofeo (Morales, 2018).....	25
Ilustración 9. Recurso gráfico de Dilo en señas.....	26
Ilustración 10. Pantalla principal de Dilo en señas (Garza, 2015).....	26
Ilustración 11. Selector de secciones (Garza, 2015).....	27
Ilustración 12. Sección de aprendizaje de Dilo en señas (Garza, 2015).....	27
Ilustración 13. Sección de juego de Dilo en señas (Garza, 2015).....	27
Ilustración 14. Score de Dilo en señas (Garza, 2015).....	28
Ilustración 15. Juego completado (Garza, 2015).....	28
<i>Ilustración 16. Recurso gráfico de Dilo en señas, la expedición extraído de https://play.google.com/store/apps/details?id=com.diloensenas.lsm2 Fecha de revisión:27/09/2018.....</i>	28
Ilustración 17. Error al ejecutar la aplicación (Garza, 2018).....	29
Ilustración 18 y 19. Imágenes de la presentación de las categorías (Garza, 2018).....	30
Ilustración 18 y 19. Imágenes de la presentación de las categorías (Garza, 2018).....	30
Ilustración 20. Primera sección de la aplicación Dilo en señas (Garza, 2018).....	30
Ilustración 21. La recompensa por jugar (Garza, 2018).....	30
Ilustración 22. Botón Desafío habilitado (Garza, 2018).....	31
Ilustración 23 y 24. Pantallas de la sección desafío (Garza, 2018).....	31
Ilustración 23 y 24. Pantallas de la sección desafío (Garza, 2018).....	31
Ilustración 25. Barra de avance del juego (Garza, 2018).....	31
Ilustración 26. Prototipo I versión final.....	33

Ilustración 27. Prototipo I primera versión.....	33
Ilustración 28 Videos del abecedario en LSM.....	34
Ilustración 29. Score cada 20 reactivos en la versión final.....	35
Ilustración 30. Score cada 100 reactivos en la versión final.....	35
Ilustración 31. Pantalla de error en la versión final.....	36
Ilustración 32. Pantalla de SQL server de la tabla Sintli.....	37
Ilustración 33. Pantalla del prototipo Maimachiotl.....	38
Ilustración 34. Ejecución de aplicación en consola de RandomArrange.....	40
Ilustración 35. Muestra de un video introducido en la aplicación maimachiotl (Alegre).....	41
Ilustración 36. Pantalla de la aplicación MainMachiome.....	43
Ilustración 37. Parte de la tabla Sintli de MainMachiome.....	44
Ilustración 38. Tablas de la base de datos Tlahtolli de la aplicación MainMachiome.....	48
Ilustración 39. Contenido de una de las tablas Tlanextli.....	48
Ilustración 40 Muestra de video introducido en la aplicación MainMachiome (Investigar).....	49
Ilustración 41. Pantalla de la aplicación.....	50
Ilustración 42 y 43. Representación del vector y del registro de cada elemento.....	51
Ilustración 42 y 43. Representación del vector y del registro de cada elemento.....	51
Ilustración 44. Representación del problema que quiero solucionar.....	52
Ilustración 45. Números aleatorios.....	53
Ilustración 46. Probabilidad de cada elemento.....	54
Ilustración 47. Intervalos de probabilidad.....	54
Ilustración 48. Secuencia simulada.....	55
Ilustración 49. Números aleatorios.....	56
Ilustración 50. Respuesta alternativa.....	57
Ilustración 51. Números aleatorios.....	58
Ilustración 52. Probabilidad de acierto.....	58
Ilustración 53. Respuesta de usuario (simulado).....	59
Ilustración 54. Similitudes entre propuesta generada por la computadora y la respuesta del usuario simulado.....	59

Ilustración 55. Implementación actualizada del vector de probabilidades de letras.....	60
Ilustración 56. Corrección de probabilidades de las letras con límites.....	60
Ilustración 57. Calibración, vector (.2 .5 .8 .5 .2).....	61
Ilustración 58. Calibración, vector (.8 .8 .6 .4 .4).....	61
Ilustración 59. Calibración, vector (.8 .8 .8 .8 .8).....	61
Ilustración 60. Calibración, vector (.2 .2 .2 .2 .2).....	62
Ilustración 61. Calibración, vector (.5 .5 .5 .5 .5).....	62
Ilustración 62. Bosquejo de selección de 5 elementos.....	62
Ilustración 63. Bosquejo de selección de 10 elementos.....	63
Ilustración 64. Frecuencia ajustada por refuerzo de varianza.....	63
Ilustración 65. Frecuencia de Varianza.....	64
Ilustración 66. Tabla de DB Tlanextli, vector de probabilidades.....	66
Ilustración 67. Vector de Error obtenido de la respuesta del usuario.....	67
Ilustración 68. Vector normalizado.....	67
Ilustración 69. Vector normalizado centrado, vector de modificaciones.....	68
Ilustración 70. Vector de modificaciones ponderado.....	69
Ilustración 71. Vector de probabilidades.....	69
Ilustración 72. Columna de elementos seleccionados.....	70
Ilustración 73. Columna de respuesta de usuario.....	71
Ilustración 74. Elementos seleccionados.....	71
Ilustración 75. Probabilidad de los elementos seleccionados.....	71
Ilustración 76. Primer gráfica de probabilidades de los elementos seleccionados.....	71
Ilustración 77. Vector procesado.....	72
Ilustración 78. Probabilidad obtenida.....	72
Ilustración 79. Probabilidad generada.....	72
Ilustración 80. Registro de correcciones en el log.....	73
Ilustración 81. Registro de corrección de errores.....	75
Ilustración 82. Corrida 2, re-implementación del algoritmo.....	77
Ilustración 83. Asistente para importar archivos de texto en Microsoft Excel.....	79

Ilustración 84. Columna de elementos.....	79
Ilustración 85. Columna de número de apariciones.....	80
Ilustración 86. Columna de número de aciertos.....	80
Ilustración 87. Columna de razón por la que fue expulsada la palabra.....	80
Ilustración 88. Columna de la posición del vector que es sustituida.....	81
Ilustración 89. Columna de estado del elemento de salida.....	81
Ilustración 90. Etiquetado de secuencia de aparición.....	82
Ilustración 91. Log reordenado.....	84
Ilustración 92. Etiquetado por elemento.....	84
Ilustración 93. Gráfica del log Tlahtolme.....	85
Ilustración 94. Comparación de logs.....	85
Ilustración 95. Tabla de DB Sintli.....	88
Ilustración 96. Log posterior a la implementación del dosificador.....	89
Ilustración 97. Gráfica del log con variable de contrapeso.....	90
Ilustración 98. Gráfica del log con la variable de contrapeso neutralizada.....	92
Ilustración 99. Rediseño de la tabla Sintli.....	93
Ilustración 100. Gráfica del log al aplicar el algoritmo de modificación de frecuencias.....	95
Ilustración 101. Número de elementos distribuidos en 3 categorías.....	96
Ilustración 102. Log de Tlahtolme con probabilidad de acierto sobre los elementos.....	96

Tablas

Tabla 1 Agrupación de palabras y número de estrellas para desbloquear el grupo.....	23
Tabla 2 Agrupación de palabras de la aplicación Dilo en señas.....	26
Tabla 3 Agrupación de palabras de la aplicación dilo en señas, la expedición.....	29
Tabla 4 Set 0 de señas de LSM.....	40
Tabla 5 Conjunto I.....	45
Tabla 6 Conjunto II.....	45
Tabla 7 Conjunto III.....	46
Tabla 8 Conjunto IV.....	47

Tabla 9 Conjunto V.....	47
Tabla 10. Operaciones aplicadas a un vector con el proceso del simulador.....	64
Tabla 11. Operaciones aplicadas a un vector con el nuevo algoritmo.....	65

Introducción.

Hoy en día estamos viviendo una ola de cambios sin precedentes. Los cambios son vertiginosos y al alcance de la mano. Recuerdo hace varios años que vi en la televisión en un comercial estaban promocionando video llamadas, junto con unos teléfonos con pantallas y una tarifa. No tardó mucho para que las video-llamadas por Internet se popularizaran y lo mejor han sido 'gratuitas'.

Actualmente no es extraño ver niños y adolescentes con aparatos celulares de gama alta, diestros en digitalizar mensajes con los pulgares y en un sinfín de interacciones con aplicaciones. En cualquier oportunidad la mayoría de personas miran algún contenido a través de su celular, oyen música revisan el Facebook®, se toman selfies, ven películas, registran información a través de fotografías o simplemente matan el tiempo con un juego en el celular. Y es aquí donde yo encuentro una oportunidad.

Este escenario no es reciente, ha sido así desde hace varios años. La importancia no estriba únicamente en las herramientas de creación, que se han simplificado enormemente, sino, por los medios de distribución. Todo esto ponen al alcance posibilidades antes inimaginables tanto como para los usuarios como para los desarrolladores.

Comenzaré con la inquietud de este trabajo. No fue sino hace 12 años aproximadamente que cursaba alguno de los niveles de japonés. En aquel tiempo pensé, ¿Cómo puedo hacer para aprender mejor?, ya que sabía lo básico de programación. Así que desarrollé un pequeño programa en Visual Basic. No recuerdo cómo era, ni la mecánica, solo recuerdo que utilizaba imágenes y que las llamaba desde un switch.

El programa no lo acabé. Estoy seguro que la aplicación hacía algo, no bien, ni mucho menos de la mejor manera. Además, el proceso de aprendizaje fue muy vertiginoso que cuando por fin tuve la aplicación, en clase ya íbamos 3 temas adelante. Lo dejé en mi lista de cosas pendientes, porque estaba seguro que un programa con esas características me sería útil.

Mucho tiempo después escuchas sobre las Tecnologías de la Información y Comunicación (TIC's) y del software educativo. Entonces descargué las novedades y le picas por ahí y por allá. Tres clics en cada aplicación para desistir e ir por la próxima aplicación, o no tenía el contenido que yo deseaba, o yo mismo tenía que poner el contenido. En muchos de los casos la experiencia era insatisfactoria. Entonces llegas a la conclusión de no descargar más aplicaciones para no ocupar

espacio. No estoy criticando alguna de estas aplicaciones, mas ninguna generó el deseo de ejecutarla nuevamente.

Posteriormente llegaron los celulares con pantalla. Y la posibilidad de instalar una gama muy amplia de videojuegos. Siempre elegí los juegos mentales, pulztes, ajedrez o scrabble. Pero me robaron el celular y con él las aplicaciones. Dos de ellos fueron bastante buenos, uno para sumar mentalmente y un scrabble multidiomas, sin embargo, me dejaban con la sensación de que algo faltaba.

El de las sumas me servía cuando hacia cálculos mentales, era más rápido con las cuentas. Debo de aceptar que fue atractivo, pero ¿qué me aporta el que supiera hacer esas operaciones? Viniendo de alguien que se hace llamar un matemático podría sonar incongruente. Solamente para aclarar este punto, las matemáticas son mucho más que sumas, restas y multiplicaciones, aunque son su fundamento, estás operaciones pueden ser realizadas por ciertos aparatos electrónicos que las harán mucho más rápido y sin errores. Las ventajas de aprender a operar mentalmente tenían muy poco que aportarme (en algo muy concreto, porque pueden resultar muy benéficas a la hora de calcular el cambio).

Fui fan del Scrabble en otros idiomas, sin embargo, me di cuenta que no aprendía las palabras. No digo que el Scrabble sea un mal juego, sino que para aprender palabras de otro idioma no era muy eficaz, porque en detalle no estaba realizando asociaciones con conceptos. Sirve muy bien para afianzar palabras que uno ya conoce. Su estructura es compleja, que analizarlo está fuera de los objetivos de este trabajo. En resumen, si lo más importante era aprender otra lengua, este no me era útil.

¿No podía haber un juego que me enseñara algo de principio a fin, sin llegar a ser aburrido? Con aburrido estoy señalando a los juegos de aprendizaje y con enseñar, por los juegos comerciales. **El principal objetivo de este trabajo es ajustar la mayor cantidad de contenido de un conocimiento específico (elementos) en este caso el LSM a una estructura de suministro a través de un juego.** Con este trabajo intento satisfacer la primera necesidad, 'el aprender algo'. Se argumentará que muchos otros programas tienen el mismo objetivo. Y así es, solo que en mi particular punto de vista no afianzan el contenido a la estructura del juego, convertir al elemento en ficha de juego, sino que adecuan el contenido a una estructura sencilla en un juego para el caso de los juegos educativos. Por otro lado, los juegos comerciales que enseñan, relegan el contenido, y la actividad central es de otra índole, la de divertirse. En núcleo del trabajo es: **Controlar las frecuencias de**

cada uno de los elementos de un grupo grande, puede permitir generar secuencias de suministro con propiedades deseadas.

Este trabajo es un paso en esa dirección, ya que explicaré cómo ajusté el contenido a la estructura. **En este trabajo no voy a demostrar si mi suposición es cierta. Aquí solo me restringiré a generar la infraestructura que me permita variar las condiciones y controlar probabilidades totales. La contrastación con respuestas reales de usuarios quedará para otra investigación porque implica otra arquitectura.**

El objetivo final es determinar características de suministro de secuencias de elementos que se adecuen a la capacidad de respuesta del usuario.

La presente obra está compuesta por 4 capítulos con las siguientes consideraciones:

En el capítulo I, explicaré brevemente el escenario de la lengua de señas de acuerdo con mi experiencia y las motivaciones que me llevaron a seleccionarlo como el contenido de la aplicación.

En el capítulo II, me volcaré hacia otra área totalmente diferente, pero dónde se encuentra incrustado buena parte de éste trabajo, Game-Based Learning. Incluiré algunas clasificaciones para poder ubicar la aplicación en todo aquel universo. Analizaré la función de la aplicación dentro de diversas perspectivas relacionadas y mostraré cuales aspectos a mi parecer se están menospreciando. Finalmente haré una comparativa de las aplicaciones actuales con los mismos objetivos.

Los capítulos de aplicación y que conciernen a este trabajo son el capítulo III y el IV. En el capítulo III mostraré los diferentes prototipos desarrollados para alcanzar los objetivos de este trabajo y explicaré brevemente sus mecánicas y su interacción con la totalidad del trabajo.

El capítulo IV, Algoritmos, es el capítulo más importante, desarrollaré a detalle los cuatro algoritmos, los pilares de la aplicación, implementados en la misma.

- Algoritmo 1: Definición de probabilidades.
- Algoritmo 2: Modificación de frecuencias instantáneas de un conjunto de ítems seleccionados.
- Algoritmo 3: Selector de nuevos ítems.
- Algoritmo 4: Modificación de frecuencias con base a las probabilidades de todos los ítems.

Por último, el apartado de conclusiones comentaré los objetivos alcanzados y esbozaré la estructura del proyecto en su totalidad y el trabajo pendiente.

Capítulo I: Lengua de señas.

1.1 ¿Qué es la lengua de señas?

La lengua de señas consiste en una serie de signos gestuales articulados con las manos y acompañados de expresiones faciales, mirada intencional y movimiento corporal, dotados de función lingüística, forma parte del patrimonio lingüístico de dicha comunidad y es tan rica y compleja en gramática y vocabulario como cualquier lengua oral (Consejo Nacional para el Desarrollo y la Inclusión de las Personas con Discapacidad [CNDIPD], 2017). Elegí incluir esta definición, si es que se le puede llamar de esa manera, porque se encontraba pegada en los muros de la sala de lengua de señas de la Biblioteca Vasconcelos, donde tomé algunos cursos de lengua de señas y de alguna manera resulta ser adecuada a los ideales de la comunidad sorda, que en ciertos aspectos que les compete es delicada.

La lengua de señas es el medio de comunicación que utilizan las comunidades de sordos del mundo; sin embargo, cada una tiene un sistema diferente (López García, Rodríguez Cervantes, Zamora Martínez, & San Esteban Sosa, 2006). Este texto aparece en las primeras hojas del manual Mis manos que hablan, lo recupero, porque, a pesar de parecer un enunciado bastante modesto refiere a un concepto de gran magnitud, la comunicación, se dicen que la diferencia entre los animales y los seres humanos es la capacidad de pensar, otros precisan que es la capacidad de comunicarse, francamente no estoy seguro de cual sea la diferencia, pero si estoy consciente del gran valor y alcance de poseer un “medio de comunicación.”

En el libro continúa: La cultura del sordo está centrada en la percepción visual, su mundo está basado en la vista y su idioma es la lengua de señas (López García et al., 2006). Estas líneas las incluyo porque, aunque, parece trivial, el hecho de que los elementos sean visuales permiten la implementación de la aplicación, en otras palabras, las señas y las imágenes están al mismo nivel perceptual.

1.2 ¿Por qué elegí la lengua de señas?

Podría decir que se fueron dando las cosas, el juego en un principio lo pensé para aprender nawatl, sin embargo, requería de mucho más trabajo, e incluso probar hipótesis al respecto de la constitución de la lengua nawatl, varias investigaciones que no podría darme el lujo de llevar a

cabo. Coincidió que por aquel entonces estaba aprendiendo lengua de señas, y tenía registrados unos videos del abecedario manual que nos habían compartido los profesores de lengua de señas, por WhatsApp®, así que diseñe un prototipo del juego y de paso integre una base de datos. En ese momento ya estaban hechas las propuestas con el simulador en Excel®, pero no lo integré. Los videos quedaron perfectos para la actividad que estaba desarrollando en ese momento.

Conforme iba tomando los cursos, los profesores me compartían experiencias y necesidades que me cambiaron mi visión y me orientaron por hacerlo con la lengua de señas.

Algunas de las necesidades de los sordos que pude apreciar son:

Lengua.

La menciono, porque representa nuestro medio con el que interactuamos con la sociedad. Parecería trivial, sin embargo, miles de personas, no están en las mismas condiciones, más del 90% de la población es oyente, entonces tenemos la posibilidad de comunicarnos con una multitud de personas, aunque nosotros terminemos no haciéndolo con todos, las posibilidades son muy grandes. En la estadística (“La discapacidad en México , datos al 2014”, 2014) el INEGI registra un total de 7.1 millones de personas con alguna discapacidad, de los cuales 33.5% indicó tener discapacidad al ‘Escuchar (aunque use aparato auditivo)’, esto nos da un aproximado de 2.3 millones de personas con discapacidad auditiva que representa un poco menos del 2 por ciento de la población total de la República Mexicana, los cuales están dispersos en todo el territorio nacional y me atrevería a sugerir que una cierta parte de ellos se encuentra marginado. Con las nuevas tecnologías se ha recorrido una brecha muy grande que ha abierto puentes antes inimaginables. Las video-llamadas, las redes sociales por mencionar los más importantes, porque son canales de comunicación donde actualmente se puede transmitir su lengua.

Conocimiento.

Desde mi perspectiva es la segunda en importancia, ya que nos encontramos en un mundo cada vez más competitivo, donde el conocimiento puede llegar a generar una diferencia notable. Esto lo digo como matemático y programador. Muchos podrían llegar a argüir que son capaces de entender cualquier concepto, cito: Se ha comprobado, a través de muchas investigaciones, que el lenguaje de señas es tan rico y dinámico como el oral. Aparentemente tiene menos elementos que el código verbal, pero no porque los sordos no puedan utilizarlos o desconozcan de su uso, sino porque no es necesario (López García et al., 2006). Es posible expresar las ideas o conceptos de

una manera descriptiva, pero no juega el mismo papel el aprendizaje describiendo cosas, que nombrando las cosas. Desde mi experiencia el conocimiento especializado como por ejemplo el matemático necesita de un conjunto de palabras que relacione los conceptos, para poder ser recuperados.

Educación.

Podría parecer lo mismo que la anterior, sin embargo, se enfoca a nuestra interacción como seres humanos, lejos de las oportunidades que nos proporciona el internet, es necesaria una educación ajustada a sus capacidades y medios, es un derecho. Que el contenido de lo que se está transmitiendo sea a través de un medio de comunicación distinto al suyo (lenguaje oral), para mí representa un obstáculo considerable. La solución que proponen las autoridades es mediar las clases a través de intérpretes, claramente esto mejora la situación, pero dista mucho de solucionar el problema.

Intérpretes.

Para poder interactuar con una sociedad de oyentes en ciertos contextos específicos es necesario la mediación de intérpretes. La desventaja es que son escasos.

No son todos los problemas, ni los principales. Son los que pude identificar.

En este punto me gustaría aclarar que no es el objetivo solucionar alguna de estas problemáticas, simplemente en propagar las señas de la LSM. Aunque mi propuesta únicamente consiste en un simple juego de asociación, este podría tener un gran impacto. También hay que tener en cuenta que la lengua tiene otros elementos que no son señas; gestos, posición del cuerpo y movimiento en el espacio. Sin embargo, teniendo un bagaje considerable es posible entender qué se está comunicando.

La meta del proyecto parcialmente es la de ofrecer una buena herramienta, **para fijar en los oyentes un vocabulario extenso en LSM**. De esta manera servir como apoyo a cualquiera que desee aprender la LSM. Hay que aclarar que no sustituye a los cursos, ya que una lengua es más que elementos léxicos y en los cursos pueden generarse diferentes ambientes, donde los profesores juegan un papel muy activo en la administración de eso contenidos, sintaxis y semántica, no se diga los sociables.

1.3 Alcances de la aplicación en el campo de la LS

La población que esta aplicación busca beneficiar directamente es a la comunidad de estudiantes de lengua de señas para proveer de una herramienta que:

- Contenga un corpus considerable de palabras.
- Relacione directamente a la seña con el objeto (imagen).
- Proveer de un número considerable de exposiciones.
- Evitar patrones en la secuencia.
- Aumentar la frecuencia de aparición de los elementos donde el usuario se equivoque con mayor frecuencia.

Estas características de la aplicación no resultan arbitrarias, de hecho, obedecen a lo que yo consideré deseable en mi proceso de aprendizaje.

La aplicación también busca beneficiar indirectamente a la comunidad sorda, ya que podría ayudar en ciertos contextos. Por ejemplo:

Es una propuesta para los sordos que no tienen acceso al contenido de la LSM. Existen sordos que no tienen acceso a materiales en LSM, ellos generan su propio código que limita sus posibilidades de comunicación. Gracias al Internet hay muchos materiales disponibles, invariablemente es necesario contar con la tecnología, sin embargo, la opción está disponible y los costos se han vuelto más accesibles.

Fomentar el conocimiento de la LSM en la comunidad oyente. En primera porque puede convertirse en un puente de comunicación entre las personas sordas y oyentes, en segunda, porque su medio de comunicación es viso-gestual y esto implica un universo que como oyentes no necesitamos, pero que el explorarlo puede significar un desarrollo cognitivo sin precedentes.

El desarrollo de intérpretes es importante, ya que pueden ayudar a las personas que conforman la comunidad sorda en varios contextos, sin embargo, esta capacidad requiere de una preparación específica y especializada, que escapa a los objetivos de esta aplicación. Sin embargo, es una aportación en el camino para los que deseen especializarse.

Menciono a los intérpretes, ya que en México nos encontramos rezagados en este tema y los recursos humanos capacitados son escasos. Por lo mismo, existen bastantes áreas de oportunidad.

Porcentaje de concentración de intérpretes de lenguas de señas, por entidad de residencia

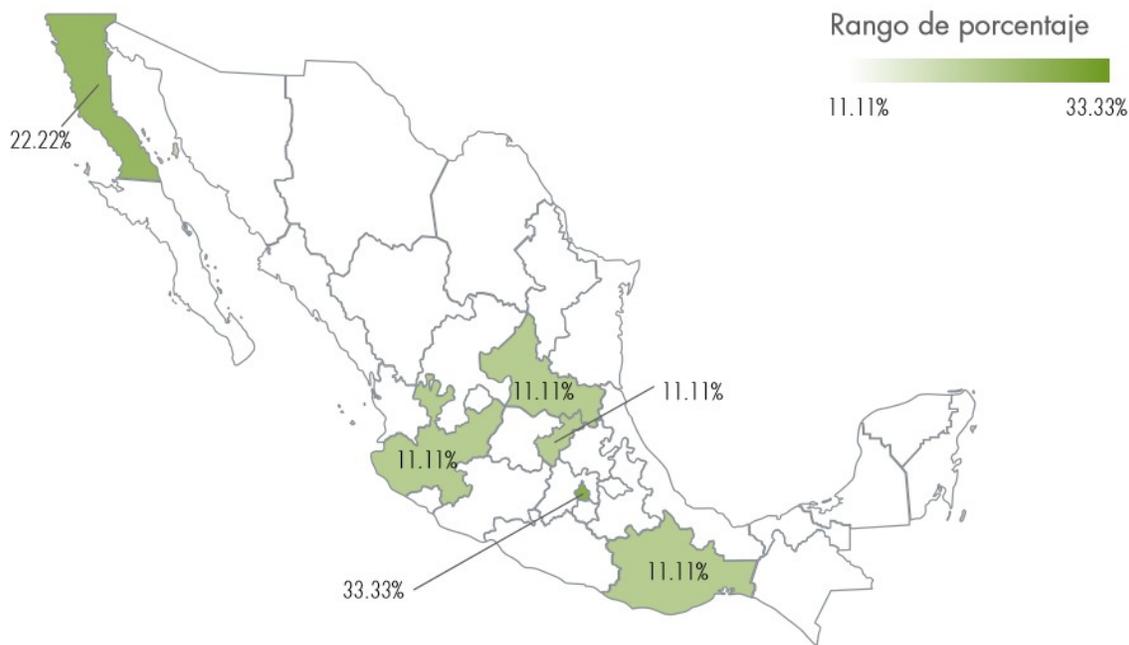


Ilustración 1. Lugar de residencia por tipo de lengua; lengua de señas (Holcomb, 2017, p. 24)

A partir de la Encuesta sobre la traducción y la interpretación en México 2017 podemos darnos cuenta que tenemos un déficit de intérpretes en la mayoría de los estados. Esto no quiere decir que no se ejerza la práctica, es posible que se lleve a cabo, pero no por personal capacitado. Solo 6 estados cuentan con intérpretes dedicados y si vemos el número de intérpretes totales nos daremos cuenta que son pocos en comparación con la población que precisa de ellos.

No resulta exagerado el texto: En países del Tercer Mundo se impide que los sordos desarrollen sus capacidades. Por esto, muy pocos logran convertirse en miembros productivos y plenos dentro de la sociedad (López García et al., 2006). La primera vez que leí este texto me llamó la atención sin llegar a ver la dimensión del problema.

Un día tuve la oportunidad de asistir a una fiesta de sordos, y yo era el inadapado en ese contexto. Ellos me invitaban a participar de sus charlas, pero yo era incapaz de expresar mis ideas o mis pensamientos en su lengua, rehuía discretamente, pero me mantenía atento a sus conversaciones. Esta experiencia, aunque traumática, me pareció excitante. Era otro mundo en una pequeña casa en la colonia San Fernando de Tlalpan. No necesitaba recorrer grandes

distancias para presenciar los procesos de una sociedad con una lengua extranjera, en el sentido de desconocida. Simplemente tenía que tomar un transporte público de 6 pesos, que me llevaría a explorar otra dimensión del lenguaje.

Puede parecer extravagante el término “otro mundo”, yo estoy convencido que así es. Las lenguas orales y de señas tienen muchas características en común, como por ejemplo la estructura, pero por la naturaleza de las mismas se exploran de muy diferentes formas. Las orales son acústico-vocales, las signantes, viso-gestuales. Cosa que las coloca en 2 categorías muy diferentes, pero lo más importante cumplen la misma función.

Por experiencia propia, otra manera de configurar nuestro universo. Los invito a estudiar una lengua de señas o cualquier lengua extranjera u originaria, por los beneficios que esto conlleva, tanto fisiológicos como de experiencias. Sobran estudios que así lo pregonan.

Capítulo II: Game-Based Learning.

2.1 Juegos y aprendizaje.

Existen varias áreas de estudio alrededor de los juegos digitales como Gamification, Computer-Assisted Learning, Educational Gaming o incluso Technology-Assisted Learning, varían de acuerdo a los objetivos que se persiguen, ya que las posibilidades con los juegos son infinitas. Estamos en la intersección de dos campos, el de los juegos y el de las herramientas digitales, el primero una manera de interactuar en el mundo, el segundo, un abanico infinito de posibilidades para materializarlos. No es de mi interés teorizar en esas áreas del conocimiento. Lo que expongo en este capítulo son opiniones accesibles de quienes se dedican a investigar con sus intereses particulares de los cuales tomaré lo que me resulte útil.

No vale la pena reparar en estos temas. No porque no sea importante sino, porque la aplicación (programa) real es una simple interacción con el usuario; lo que aquí desarrollé es la construcción de una estructura que me permita controlar directamente o indirectamente unas características de las secuencias generadas. Controlar el suministro de nuevos ítems, probabilidades totales, palabras cargadas. Generar incremento gradual y la sensación de aleatoriedad.

Definir qué es un juego no es una tarea trivial, de hecho, la mayoría de autores se limita a enunciar rasgos característicos: Si hacemos a un lado las diferencias de genero y las complejidades tecnológicas, todos los juegos comparten 4 rasgos definitorios: una meta, reglas, un sistema de retroalimentación y la participación voluntaria (McGonigal, 2011).

Elegí este marco de referencia por lo sencillo de la exposición. Hay una infinidad de definiciones que delimitan con mayor precisión el tema en cuestión, en realidad es un tema complejo que incluso podría ser analizado con una perspectiva filosófica, que resultaría infructuosa en este trabajo.

Procedemos a definir el concepto de video juego: Un video juego es, generalmente hablando, cualquier juego efectuado usando el poder computacional y una pantalla de video (Torrente et al., 2009).

A pesar de no ser la definición más explícita, la seleccioné porque tiene un par de conceptos clave, medulares en este trabajo “poder computacional” y “pantalla de video.” El primer concepto “Poder computacional” que refiere a nuestra herramienta por excelencia, la computadora. En

general los video juegos son demandantes de muchos recursos, sin embargo, estos recursos se destinan a otro tipo de necesidades (en su mayoría graficas). El presente trabajo es una pequeña aplicación que intenta explotar una capacidad característica de las computadoras, que es la capacidad de cálculo. ¿Esto suena absurdo? ¿Los video juegos no aprovechan muchas de las características de la computadora? La respuesta es afirmativa, aprovecha muchos recursos, la diferencia consiste en las necesidades que se satisfacen. Y desde mi punto de vista se está desaprovechando una pequeña área de oportunidad bastante prometedora.

Los videojuegos tuvieron su nacimiento en los laboratorios computacionales no como un producto, sino como una inquietud, ya que la tecnología estaba a su alcance, fue posterior y naturalmente que desearon capitalizarlo, la industria. Incluso se logra escuchar; la industria de los videojuegos que ocasionalmente se engloba dentro de la industria del entretenimiento. La historia ha dejado ejemplos muy creativos, creado herramientas poderosas e incluso ambientes de creación. Limitándose un poco, por ejemplo, el Aprendizaje Asistido por Computadora (CAL, por sus siglas en inglés) se ha nutrido de los avances sin prestar atención a la gama de posibilidades que ofrece una computadora, la ve como una hoja en blanco donde plasmar su contenido y no como lo que es, una máquina de cálculo muy poderosa. Las 2 proposiciones también son válidas, pero la principal y más importante desde mi particular punto de vista es relegada. Sin duda hay casos donde se implementan algoritmos más complejos y poderosos que los desarrollados en este trabajo.

Esta es una simple aplicación a un fenómeno en específico. El fenómeno es la respuesta del usuario a una secuencia de elementos. El mediador, un instrumento que se encarga de registrar la respuesta y dosificar los elementos.

Otra herramienta en la que me apoyé, aunque no hablaré al respecto porque no concierne a mi objeto de estudio, incluso como medio que ocupé, la multimedia. Si bien en este trabajo no profundizaré en esta área, me valdré de todas sus bondades, ya que, incluí audio, imágenes y video. Puede parecer muy obvio, pero bien utilizadas estas propiedades, pueden resultar una herramienta muy atractiva. Incluso implica satisfacer exigencias estéticas. Actividad que realicé simplemente por el placer de llevarla a cabo.

2.2 Tipos de video juegos

En esta pequeña sección me encargaré de recoger clasificaciones, tomé las categorías que se ajustan a las características de mi juego.

Ejercicios y prácticas

Son los worksheets (hojas de trabajo) o flash cards (tarjetas de vocabulario) de la escuela. El autor explica; Proveen de exposición repetida a hechos e información (Sharma, 2017).

Él alega que: Los programas de Ejercicios y Practicas Drill fueron el tipo de aplicación de computadora más prevaleciente por mucho años, ya que los profesores no estuvieron seguros de qué otras maneras se podían usar (Sharma, 2017).

Más adelante especifica: Las aplicaciones de Ejercicios y prácticas no utilizan toda la capacidad de las computadoras (Sharma, 2017).

Su perspectiva es desde el área de conocimiento de CAL y donde su opinión no es tan favorable a esta clasificación de juegos. Incluso su crítica es muy similar a la mía, yo hacia los Juegos de Aprendizaje y él hacia los de juegos de Ejercicios y prácticas. Y es plausible que las 2 sean igual de válidas. En el contexto del CAL el profesor produce el contenido y lo lleva a la computadora, al decir 'Toda la capacidad' supongo que se refiere a la multimedia, y a las simulaciones que son efectivas en su ramo. Para mí 'toda la capacidad' es la capacidad de cálculo, aunque he de aceptar que en este trabajo se aborda simplemente como potencialidad, ya que la capacidad del computador está más allá de lo necesario para resolver este problema.

Continúa: Los programas de Ejercicios y prácticas trabajan en primera instancia con habilidades de pensamiento de orden bajo (Sharma, 2017).

El autor tiene razón, porque se ejecuta una acción 'memorística.' No es el objetivo de este trabajo crear una aplicación que se encargue de todo el proceso de aprendizaje, sino como ya se mencionó es simplemente una herramienta que asista el proceso de aprendizaje y que explote al máximo sus características. La meta de la aplicación supone que el usuario desea poseer un número considerable de elementos. El demostrar que se cumple este supuesto está fuera de los alcances de este trabajo, pero se tuvo en cuenta para el diseño de la estructura del mismo. Con esto me refiero que el programa está diseñado para albergar n número de elementos, solo basta introducirlos a la DB, por un lado. Por lo mismo no he explorado secuencias cortas.

El mismo autor reconoce: Sin embargo, muchos programas de Ejercicios y práctica son muy Buenos en lo que ellos hacen. La computadora no se cansa de proveer a los estudiantes con práctica y retroalimentación que necesitan (Sharma, 2017).

Sinceramente espero que la presente aplicación sea una de ellas. Y agrega: La mayor parte de los programas de Ejercicios y prácticas también poseen mecanismos de seguimientos, de esta manera los estudiantes y maestros están conscientes de su progreso (Sharma, 2017). Este trabajo se concentrará en este punto, el registro del comportamiento que produce el usuario.

El autor Miguel Escalona propone una clasificación del uso de los ordenadores para facilitar el proceso de enseñanza-aprendizaje de las ciencias (Escalona, 2000). Los clasifica de la siguiente manera:

- Recurso didáctico
- Medio de información
- Herramienta de trabajo
- Elemento innovador

En Elemento innovador explica; En esta forma se debe tener conocimientos de programación por lo que además de utilizar las diferentes aplicaciones es necesario el uso de algún lenguaje de programación (Escalona, 2000). Recupero esta clasificación porque muestra una percepción que reconozco muy generalizada en la mayoría de campos que engloba la asistencia de un computador, ya sea en los juegos, multimedia o asistiendo la enseñanza o aprendizaje, para ellos es una caja negra.

Tanto los juegos como las herramientas educativas tienden a usar herramientas con objetivos específicos y los producen bajo ciertas reglas especializadas de cada campo. Por eso para el autor llegar a controlar los recursos de una computadora merece una categoría aparte.

En general, identifico un problema tanto para los juegos comerciales como para los juegos educativos desde su composición. En primer lugar, las investigaciones que se realizan con los juegos comerciales señalan los beneficios de los mismos, es indudable que se obtienen ciertos comportamientos deseables o ciertos contenidos. Aquí yo cuestiono, independientemente que estos juegos son diseñados específicamente para entretenerse ¿Es realmente equiparable el costo-beneficio? El tiempo y el dinero invertidos en esos juegos, aunque la última está perdiendo

validez por modificaciones en el modelo de negocio, actualmente, los videojuegos se están volviendo gratuitos.

Aquí podrá esgrimirse un argumento en defensa. El que los juegos tienen éxito porque satisfacen necesidades básicas, en lo que concuerdo, con todo esto la pregunta sigue siendo vigente a mi parecer.

Por otro lado, los juegos educativos virtuales, desde mi perspectiva, tiene un defecto, al limitarse a usar estructuras básicas es posible distinguir las mecánicas que están detrás y que finalmente el juego se vuelva aburrido. Yo creo que esta es una desventaja, ya que para bien o para mal el juego es algo lúdico y el poner a pensar a los usuarios para una gran mayoría, no resulta algo atractivo.

Mi propuesta consiste en ajustar el contenido a la estructura y definir las metas a partir de la estructura, así el contenido estará ligado a las metas y no será un obstáculo sino parte de la mecánica, adicionalmente los cambios serán graduales y tentativamente el reto adaptable.

Para concluir este apartado es difícil describir un videojuego. La definición depende bastante de la disciplina y los antecedentes de los individuos que lo crean; por ejemplo, futbolistas, matemáticos, programadores y psicólogos podrían tener muy diferentes puntos de vista de cómo describir un juego (Whitton, 2010).

Utilizo esta definición a propósito, ya que todo este trabajo es el producto de lo que para la intersección entre matemático y programador sería un videojuego.

2.3 Aportación:

La aplicación administra la selección de elementos(señas) procurando no generar una tendencia, la aplicación tiene definida una distribución equiprobable y posee un algoritmo que se encarga de ordenar de manera que no se perciba alguna tendencia. Esto con el fin producir una sensación de aleatoriedad, al momento de aparecer los elementos.

En este trabajó sólo se fijarán algunas de las variables que determinan el flujo. El que sean ajustadas a las necesidades del usuario se deja como trabajo posterior, ya que es necesario otro tipo de análisis e información.

Esta aplicación modificará las frecuencias de aparición de los elementos (señas) para promover el aprendizaje de los elementos donde el desempeño del usuario sea bajo.

Indudablemente tengo deficiencias desde otras áreas del conocimiento, por ejemplo, le mostré el trabajo a 2 cineastas que son entusiastas de los videojuegos, y su respuesta fue idéntica, no está listo y me ofrecieron retroalimentación. Imagino que también habrá necesidades que satisfacer en el campo de la didáctica, pedagogía, y un largo etc. Incluso en el de los juegos mismos. Como lo he mencionado anteriormente, el objetivo es desmembrar un simple juego para poder controlar y generar características deseables.

Alguna vez leí en un artículo en internet donde se menciona que “Uno de los requisitos para que un juego tenga éxito es que un niño de 5 años pueda jugarlo”, no creo que el éxito se limite únicamente en los niños. Sino que por un lado el juego sea intuitivo y por otro que no represente una carga, sino por el contrario, una manera de liberación.

El objetivo del diseño (estructura) es poder tener la capacidad de adaptar las características para poder reconocer, si es posible, patrones y finalmente producir secuencias adaptadas al usuario, tentativamente, además de no representarles esfuerzo. Para poder adaptar el proceso tendría que conocer la respuesta de los usuarios suponiendo que converge lo suficientemente lento como para ser categorizada, este será el tema de investigación posterior a esta tesina.

2.4 Aplicaciones existentes.

Actualmente existen 2 aplicaciones con el mismo objetivo

Aprende LSM y Dilo en señas.

Las 2 muy similares en estructura.



Ilustración 2. Icono de la aplicación.

Extraído de <https://play.google.com/store/apps/details?id=rodolfo.com.systemsolutionanddevelopment.aprendiendolsm>
Fecha de revisión:30/09/2018

Aprende LSM contiene 183 palabras en 12 categorías.

Categoría	Palabras	Estrellas
Abecedario	29	0
Números	20	0
Colores	13	10
Saludos	16	20
Familia	13	20
Lugares	14	20
Profesiones	11	25
Animales	14	35
Frutas	14	35
Fechas	19	50
Ropa	10	60
Deportes	10	60

Tabla 1 Agrupación de palabras y número de estrellas para desbloquear el grupo

La aplicación contiene 3 secciones.



Ilustración 3. Menú principal (Morales, 2018)

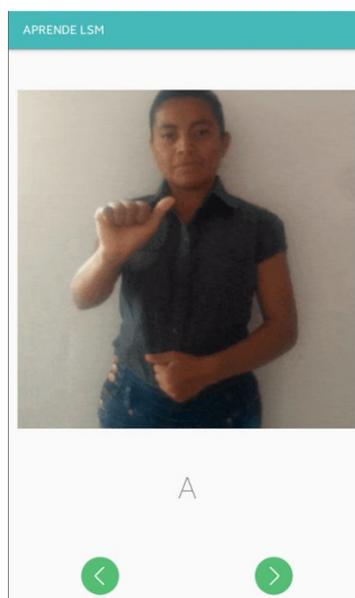


Ilustración 4. Sección de glosario (Morales, 2018)

Glosario: Contiene las 12 categorías con sus respectivas secuencias de palabras. Cada elemento muestra la correspondiente palabra con su respectivo video. El objetivo es mostrar las señas que posteriormente serán evaluadas en otra

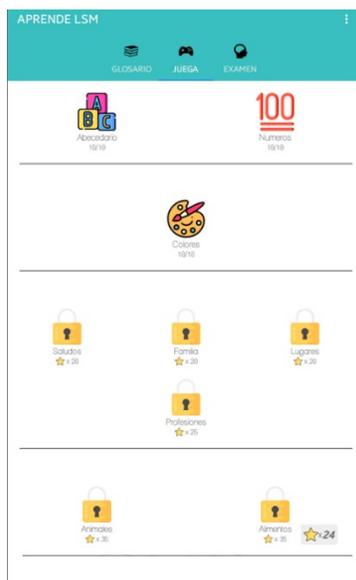


Ilustración 5. Sección de juego (Morales, 2018)

Juega: Aquí aparecen las mismas 12 categorías. En este caso cada categoría contiene un bloque de 10 preguntas. Como se puede observar fácilmente muchas de las categorías están bloqueadas.



Ilustración 6. Sección de examen (Morales, 2018)

La sección de examen se encuentra bloqueada hasta conseguir 100 estrellas, al parecer no es un requerimiento haber revisado todas las demás categorías, pero creo que es recomendable para tener un mejor desempeño.



Ilustración 7. Sección Examen, desbloqueada (Morales, 2018)

Una vez desbloqueada la sección Examen, la aplicación muestra otra mecánica de juego, asumo, para reforzar lo aprendido (Ilustración 7). Los 4 videos son repetidos hasta que se selecciona el botón calificar.

Para obtener el trofeo es necesario acertar a todos los reactivos.



Ilustración 8. Sección examen, trofeo (Morales, 2018)

La otra aplicación es Dilo en señas.

Esta aplicación tiene 2 versiones.

En la primera versión: Dilo en señas



Ilustración 9. Recurso gráfico de Dilo en señas.

Extraído de <https://play.google.com/store/apps/details?id=com.jaguarlabs.lsm> Fecha de revisión: 09/10/2018

Esta aplicación contiene: 86 palabras

Categoría	Palabras
Familia	10
Alimentos	10
Juguetes	10
Animales	10
Colores	10
Números	10
Abecedario	26

Tabla 2 Agrupación de palabras de la aplicación Dilo en señas



Ilustración 10. Pantalla principal de Dilo en señas (Garza, 2015)

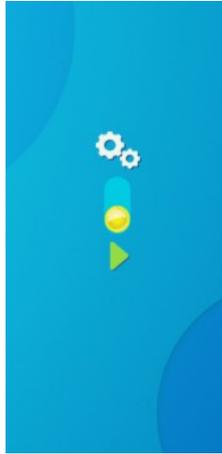


Ilustración 11. Selector de secciones (Garza, 2015)

El juego contiene 2 secciones. El icono de engrane es para el aprendizaje de las señas y el triángulo para poner a prueba los conocimientos.



Ilustración 12. Sección de aprendizaje de Dilo en señas (Garza, 2015)

La primera es de aprendizaje, donde pasan las señas con su equivalente en español.



Ilustración 13. Sección de juego de Dilo en señas (Garza, 2015)

Y otra de práctica donde se pide seleccionar la asociación correcta de la imagen.



Ilustración 14. Score de Dilo en señas (Garza, 2015)

Al final de cada categoría presenta una pantalla con la proporción de aciertos.



Ilustración 15. Juego completado (Garza, 2015)

Así luce el juego al completar todas las categorías.

La segunda versión: Dilo en señas, la expedición



Ilustración 16. Recurso gráfico de Dilo en señas, la expedición extraído de <https://play.google.com/store/apps/details?id=com.diloensenas.lsm2> Fecha de revisión:27/09/2018

Esta aplicación contiene: 150 palabras

Las palabras están organizadas por campos semánticos, promoviendo la repetición de palabras al aparecer en otros campos.

Sección	Categoría	Palabras
Casa	Familia	10
	Alimentos	10
	Juguetes	10
	Animales	10
	Verbos	10
Zoológico	Verbos	10
	Animales	10
	Saludos	10
Escuela	Números	10
	Colores	10
	Kinder	10
	Verbos	10
	Números	10
Centro Comercial	Frutas	10
	Amor y amistad	10

Tabla 3 Agrupación de palabras de la aplicación dilo en señas, la expedición



Ilustración 17. Error al ejecutar la aplicación (Garza, 2018)

Un problema grave se presenta en el 4to nivel donde marca que la conexión está muy lenta, trabándose. Al parecer es un bug de la aplicación ya que los usuarios reportan el mismo incidente en la página de descarga. La última vez me dejó jugar hasta el 6to y ahora sólo me permite jugar el primero. Solo me permitió ver 13 categorías.

Ejemplos de la primera y la última categoría



Ilustración 18 y 19. Imágenes de la presentación de las categorías (Garza, 2018)

La aplicación posee 2 secciones.



Ilustración 20. Primera sección de la aplicación Dilo en señas (Garza, 2018)

La primera sección es para aprender el vocabulario y se despliega con el botón Practicar. La segunda es el ejercicio, donde se presentan 10 reactivos. Si aciertas a todos los reactivos obtienes 3 estrellas. Para entrar a una de las dos secciones, primero se selecciona la categoría y después se elige entre el botón de practicar, para la primera o el botón con el nombre de la categoría para la segunda.



Ilustración 21. La recompensa por jugar (Garza, 2018)



Ilustración 22. Botón Desafío habilitado (Garza, 2018)

Al obtener 9 estrellas se habilita otra sección que se llama desafío. El desafío consiste en mostrar cuatro imágenes, después ocultarlas y asociar la imagen correspondiente con el video.



Ilustración 23 y 24. Pantallas de la sección desafío (Garza, 2018)

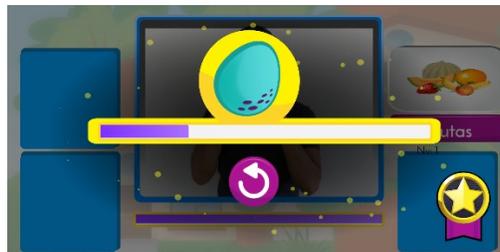


Ilustración 25. Barra de avance del juego (Garza, 2018)

Solo con los aciertos de los reactivos del desafío se avanza a lo largo del juego. La condecoración de estrella aparece cuando se llega a acertar en todos los reactivos del desafío.

De las características que me gustaría señalar. Ambas aplicaciones son buenas, sin embargo, dejan un posible hueco al agrupar los elementos en categorías. Esto puede ser trivial, pero si se analiza detenidamente, esto limita de manera tajante el número de veces que fue presentado el elemento en total. Creo sinceramente que estos diseños obedecen a no sobrecargar al usuario, el mismo objetivo que el mío, pero con diferente método.

¿Qué sucedería si existe un método para suministrar elementos, que no los agrupe por categoría o campo semántico y que sea gradual?

Actualmente el objetivo es simplificar la mecánica del juego, aunque queda la posibilidad de agregar otras mecánicas para hacer más rica la experiencia.

En general una estructura como la de la aplicación Lenguaje de Señas es agradable, sin embargo, en los últimos niveles, los requerimientos se vuelven altos y por lo tanto agotadores. Difícilmente podría decir si habría terminado los juegos si no fuera por objetivos académicos.

Con la aplicación Dilo en señas, es muy amigable, pero la complejidad del juego es ligeramente mayor, y principalmente restringe las repeticiones de los elementos de las categorías.

Esto es a grandes rasgos lo que deseo mejorar con esta propuesta de desarrollo e implementación.

Capítulo III: Prototipos.

3.1 Prototipo 1: Tlahtolli

Última versión del prototipo.



Ilustración 26. Prototipo I versión final

Ésta fue la primera versión funcional. La diferencia es una pequeña adaptación para participar en una actividad con sordos en INDEPEDI en el 2017.

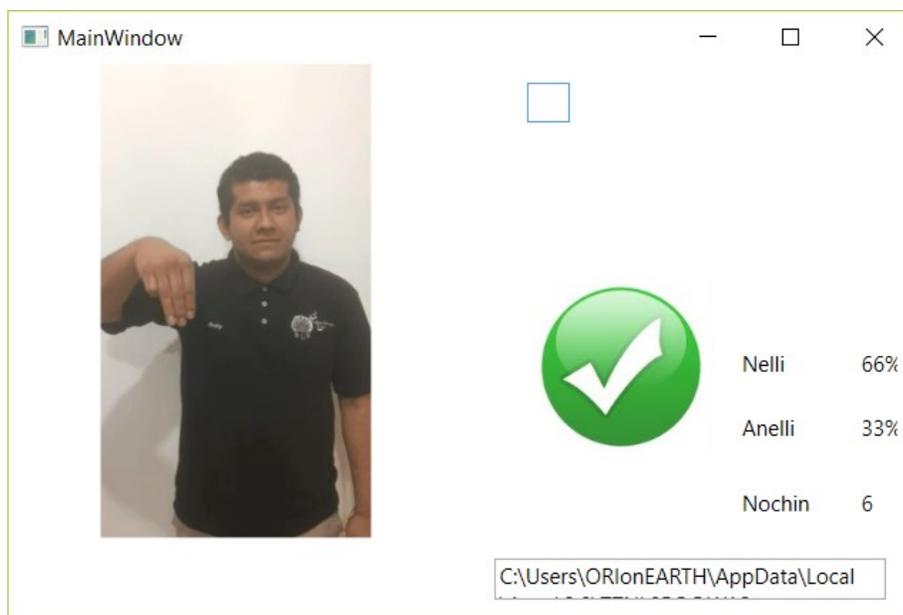


Ilustración 27. Prototipo I primera versión

Características de Tlahtolli:

- Aplicación para aprender el abecedario con señas.
- 30 elementos
- Videos (Colaboradores de Señas)
- Selección pseudo-aleatoria
- Se llama desde Inicio
- Porcentaje de acierto

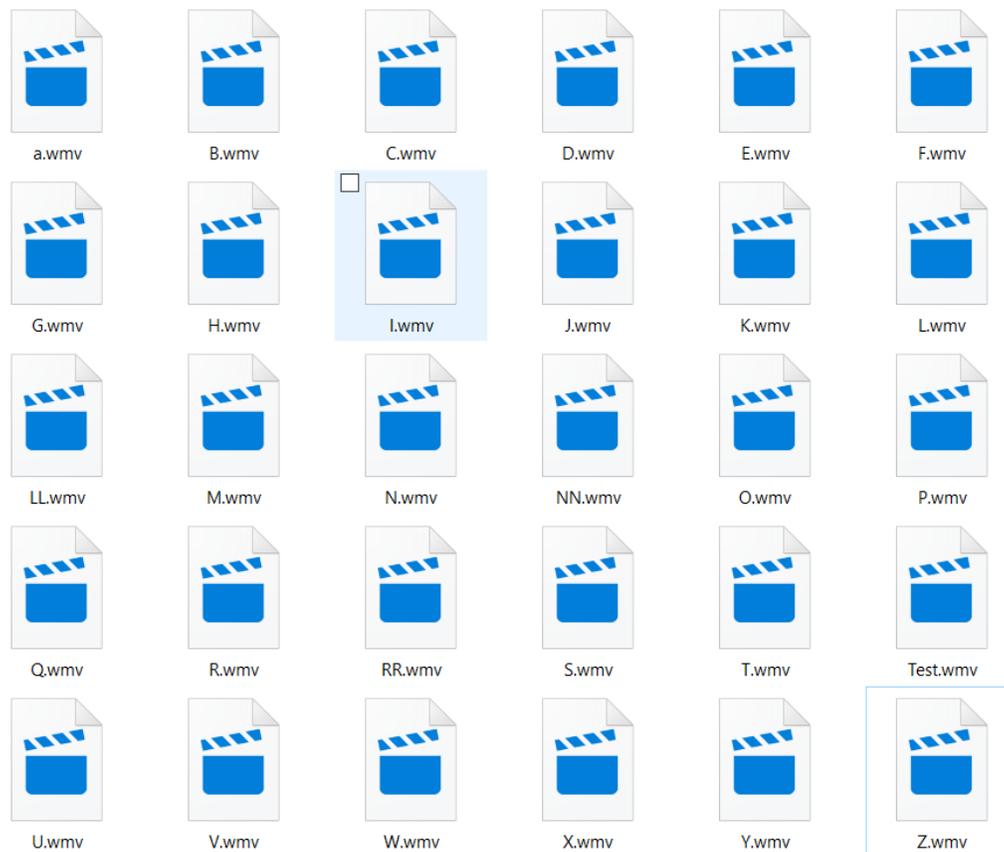


Ilustración 28 Videos del abecedario en LSM

La adaptación consistía en:

- Repetir la seña hasta acertar
- Audio al cometer errores o acertar
- Unos check-points de aciertos a lo largo de 100 exposiciones
- Un resumen de aciertos.

Mecánica:

1. Aparece un video.
2. Ingresar letra.
3. Si la letra corresponde a la seña del video se obtiene un punto, caso contrario 0.
4. Volver a paso 1.

Puntuación:

- Contiene contador de reactivos totales.
- Cada 20 reactivos una ventana muestra el desempeño correspondiente a los últimos 20 reactivos.

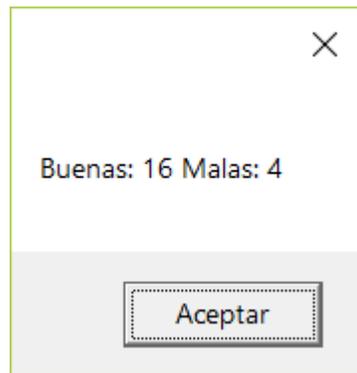


Ilustración 29. Score cada 20 reactivos en la versión final

- Cada 100 reactivos una ventana muestra el porcentaje de acierto y error.

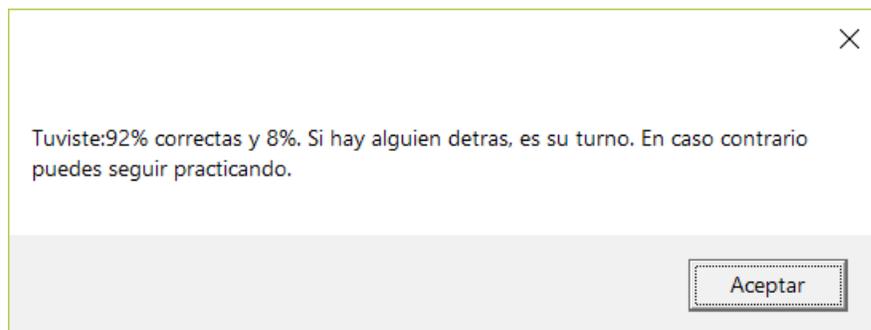


Ilustración 30. Score cada 100 reactivos en la versión final

Retroalimentación.

- Si el usuario se equivoca muestra la selección del usuario y la respuesta correcta.

Adicionales

- Cuando el usuario se equivoca se muestra la pantalla roja y muestra un icono junto a la respuesta errónea.
- Botón para repetir el video.

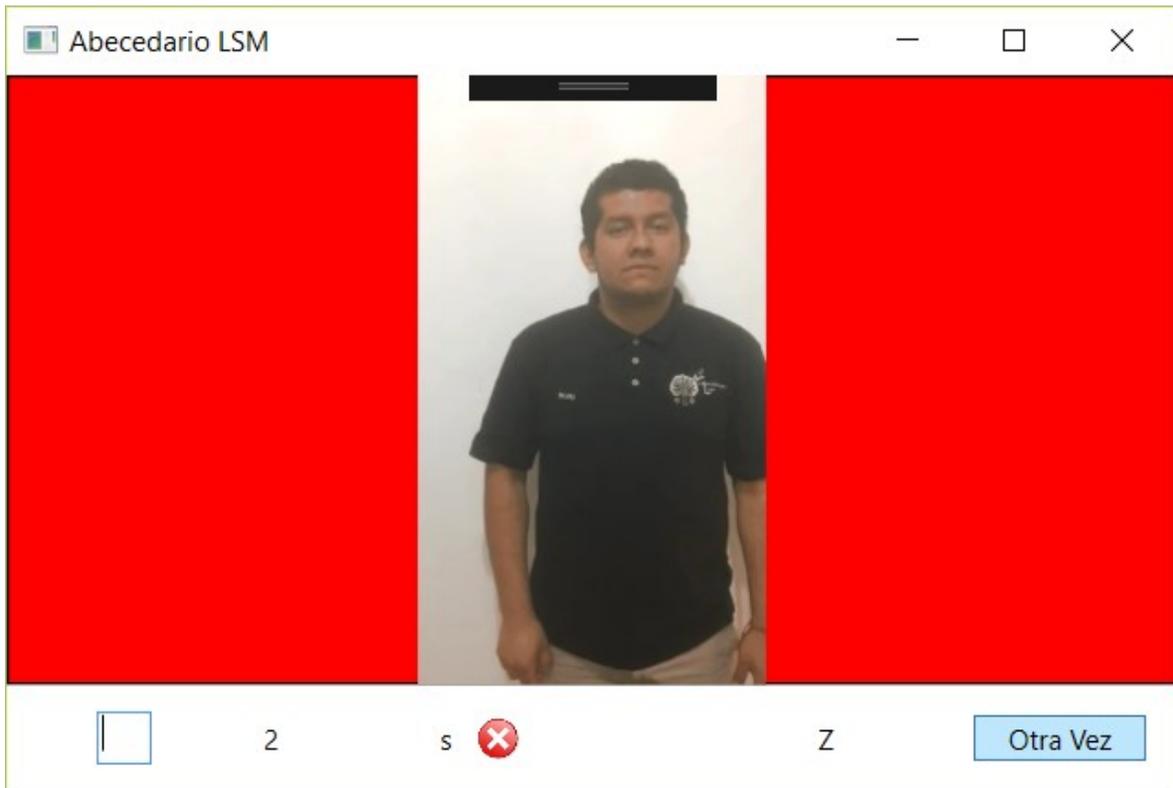


Ilustración 31. Pantalla de error en la versión final

El objetivo principal de este prototipo era definir la mecánica del juego. Además, sirvió para establecer la conexión a la base de datos.

La composición de la aplicación:

MainWindow: Ventana principal, contiene el flujo principal de la aplicación que se ejecuta al modificar el cuadro de texto tbMachiotl.

Machiotl: Clase para almacenar los datos de la tabla DB Sintli.

Tlapowalli: Clase para transformar las letras (mayúsculas y minúsculas) en números y viceversa.

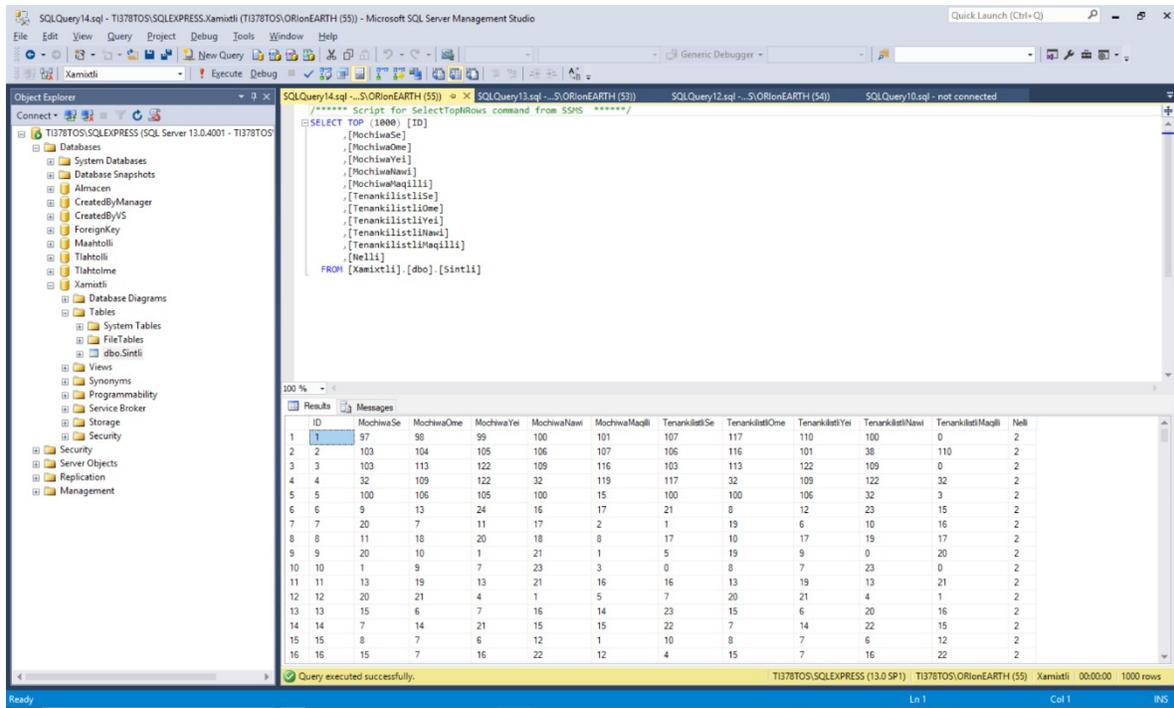


Ilustración 32. Pantalla de SQL server de la tabla Sintli

Como se puede observar la base de datos solo contenía una tabla Sintli. 5 columnas para las señas y 5 columnas para las respuestas.

3.2 Prototipo 2: Maimachiotl

Originalmente diseñé esta aplicación para aprender lengua de señas, inesperadamente se volvió una aplicación para mejorar la habilidad de mecanografiar.

Mecánica:

1. Aparece un video.
2. Ingresar palabra.
3. Click en el botón Kalaki (entrar) o presionar tecla enter.
4. Si la letra corresponde a la seña del video se obtiene punto caso contrario 0.
5. Volver a paso 1.

Puntuación:

- Keski (Cuantos): Reactivos totales
- Qalli (Bueno): Aciertos

Adicionales:

- Tlahtolmekatl (tira de palabras): selecciona el set de palabras correspondiente.
- Nawati (equilibrado): muestra cuantos elementos han sido seleccionados cuando no corresponde a 10 elementos*.
- QalTlahtol (Palabra bien): muestra el porcentaje de parecido entre las palabras del usuario y la almacenada.
- Kisa (salir): muestra el índice del vector de la palabra que sale*.
- Kalakowa (entrar): muestra el índice del vector de la palabra que entra*.
- KeskiQalli (Cuánto bien): muestra el número de valores del array que están por encima de un valor definido, en este caso* (.9).
- Motlahtol (Tu respuesta): cuando las palabras no coinciden muestra la palabra que se ingresó.
- Nelhtlahtol (Respuesta verdadera): cuando las palabras no coinciden muestra la palabra que corresponde a la imagen.
- Oksepa (Otra vez): botón para repetir video.

* Estos valores se agregaron para monitorear el algoritmo de modificación de frecuencias instantáneas.

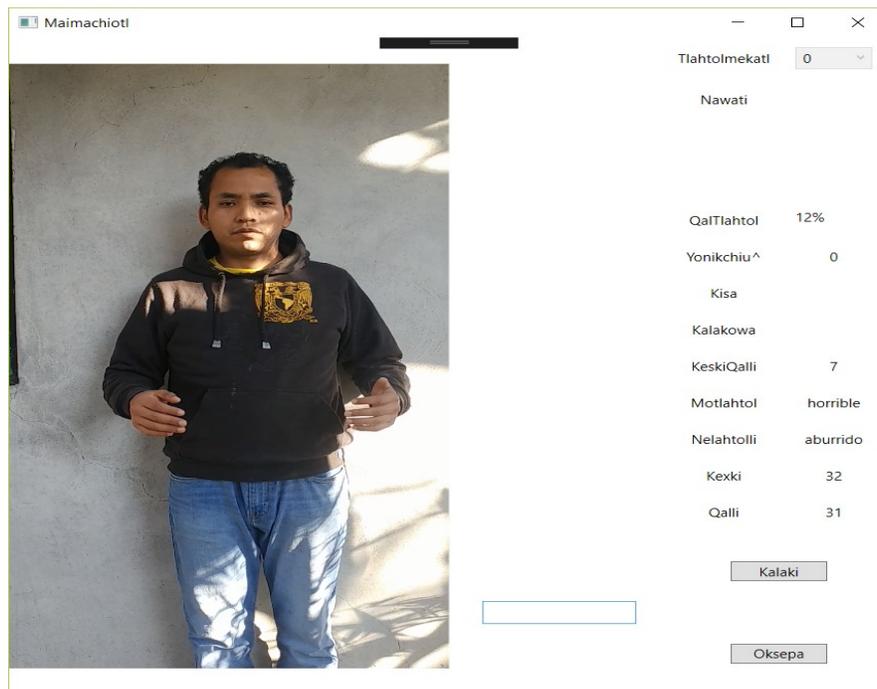


Ilustración 33. Pantalla del prototipo Maimachiotl

La composición de la aplicación:

MainWindow: Ventana principal, contiene el flujo principal de la aplicación que se ejecuta al dar click en el botón Kalaki o presionar la tecla Enter.

Tlahtolli (Relativo a lo que se expresa): Esta clase se encarga de comparar la palabra ingresada con la palabra almacenada y de acuerdo a unas reglas regresa un porcentaje de igualdad donde 1 son idénticas y 0 totalmente diferentes.

Pialistli (Tenedor): Clase para almacenar, recuperar y actualizar datos del vector de proporciones almacenado en la tabla Tlanextli.

Neltenankilistli (Repuesta verdadera): Almacenaba las selecciones de palabras y sus respectivos aciertos (actualmente en desuso).

Wehkak (Lejano): Clase para el intercambio de datos entre la tabla Sintli (fundamento) con Tlanextli(Selección).

El primer set de palabras consiste en:

primo	increible	hombre
mamá	caliente	importante
lleno	sucio	sabroso
pesado	flojo	oscuro
cerca	gris	hermano
elegante	abajo	malo
rosa	tío	largo
loco	difícil	cansado
colores	duro	bonito
próximo	diferente	vicioso
lejos	amarillo	familia
anaranjado	junto	estúpido
blanco	pobre	amigo

mujer	rojo	sobre
rápido	interesante	hijo
triste	independiente	negro
fuerte	alegre	azul
claro	mal	limpio
arriba	café	enojado
extraño	seguro	fácil
amante	lento	adentro
nieto	absurdo	nuevo
verde	responsable	terrible
profundo	morado	puntual
libre	chistoso	frio
travieso	más	aburrido
novio	inteligente	abuelo
papá	ligero	

Tabla 4 Set 0 de señas de LSM

El ordenamiento lo hice a través de una aplicación RandomArrange que yo mismo diseñé.

```

Type an integer:
40
0,12
1,23
2,9
3,33
4,27
5,4
6,2
7,17
8,5
9,26
10,14
11,37
12,32
13,6
14,15
15,24
16,10
17,7
18,39
19,29
20,34
21,38
22,22

```

Ilustración 34. Ejecución de aplicación en consola de RandomArrange

La función de este pequeño programa es generar una distribución “aleatoria” de una secuencia consecutiva menor que n. Esto para sustituir el orden alfabético con el que están dispuestos las señas en el diccionario de donde fueron extraídas. Esto porque al momento de extraer los ítems de la DB se hace ordenadamente y lo que quiero es eliminar tendencias.

Como se puede constatar fácilmente las palabras son colores, familia y adjetivos.



Ilustración 35. Muestra de un video introducido en la aplicación maimachiotl (Alegre)

3.3 Prototipo 3: Mainmachiome

Esta es la versión más importante, ya que es una pequeña alteración de la anterior, por un lado, para soportar sets de prueba, y por otro un cambio en la mecánica, se deja la evaluación de la ejecución al usuario, esto con la intención de aprender a través del hacer. La aplicación fue efectiva con la premisa, sin embargo, menospreciaba otro aspecto, el de la visión. Después de ser entrenado no era capaz de reconocer la seña incluso supiese cuál era, no tenía una asociación imagen-concepto, sino únicamente concepto-seña. Es posible que de forma abstracta se relacionen, pero de manera práctica es diferente.

Mecánica:

1. Aparece una palabra.
2. Realizas la seña de la palabra.
3. De acuerdo al criterio del usuario seleccionar el botón btNelli (paloma) si es correcto, btMauhmahtia (signo de admiración) si no está seguro y btAnelli (tache) cuando se hizo incorrectamente.
4. Volver a paso 1.

La puntuación es idéntica al del prototipo anterior.

El cuadro de texto tbTlahtolli y los botones btKalaki (entrar) y btOksepa (otra vez) se deshabilitaron.

En adicionales es idéntico a la aplicación anterior, a excepción de btOksepa que fue sustituido por el botón btOkse con el ícono de la flecha circular. Además, agregué el botón btNextia (mostrar) que muestra el video correspondiente a la palabra seleccionada.

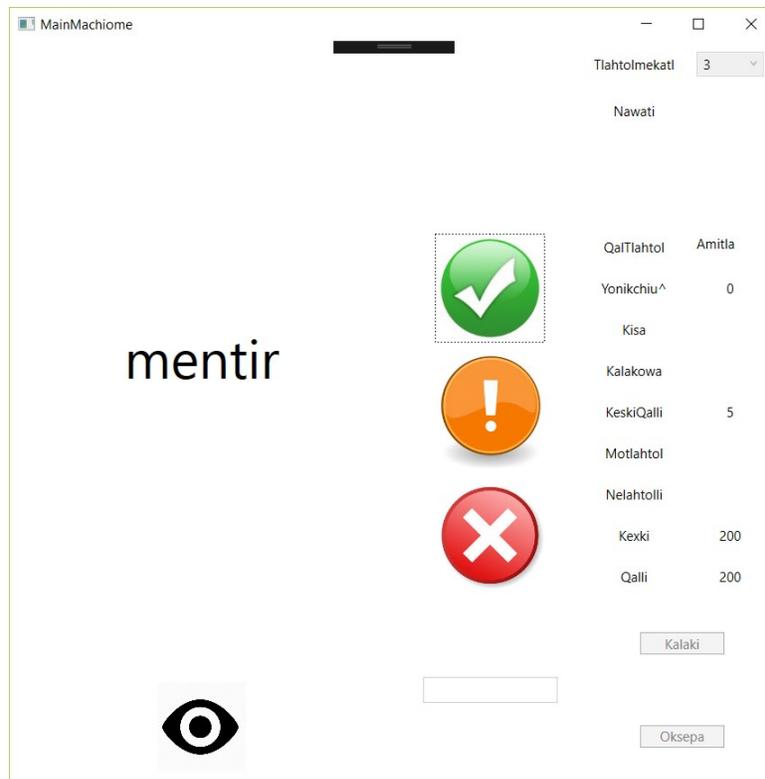


Ilustración 36. Pantalla de la aplicación MainMachiome

La composición de la aplicación:

MainWindow: ventana principal, contiene el flujo principal de la aplicación que se ejecuta al dar click en cualquiera de los botones btAnelli, btNelli o btMauhmutia.

Pialistli (Tenedor): clase para almacenar, recuperar y actualizar datos del vector de proporciones almacenado en la tabla Tlanextli.

Wehkak (Lejano): Clase para el intercambio de datos entre la tabla Sintli (fundamento) con Tlanextli(Selección).

Las palabras están agrupadas conjuntos sin un orden en específico en la BD que se identifican a través del campo **Pantli**. Están desordenadas porque los algoritmos siguen la secuencia de la BD y es deseable eliminar cualquier tipo de tendencia.

	TlahtolID	Tlamantli	Pantli	Kisa	Qalli	Pehpena	KeskiPehpena
76	91	fácil	0	60	60	yopanok	20
77	92	adentro	0	59	58	yopanok	18
78	93	nuevo	0	58	58	kema	18
79	94	temble	0	59	57	yopanok	18
80	95	puntual	0	58	56	kema	16
81	96	frio	0	59	58	yopanok	18
82	97	abumido	0	60	58	yopanok	12
83	98	abuelo	0	59	57	yopanok	17
84	100	investigar	1	44	42	yopanok	13
85	102	representar	1	44	40	yopanok	10
86	103	pegar	1	42	41	yopanok	12
87	104	tocar	1	42	41	yopanok	12
88	105	confesar	1	44	37	yopanok	7
89	106	avisar	1	41	34	kema	7
90	107	sorprender	1	42	40	yopanok	12
91	108	fue	1	42	38	yopanok	9

Ilustración 37. Parte de la tabla Sintli de MainMachiome

En esta aplicación se generaron 4 sets, todos sin excepción son verbos. Solo se usaron verbos porque los videos con los que contaba eran sobre verbos.

investigar	enamorar	dormir
representar	estudiar	vestir
pegar	volar	ir
tocar	olvidar	apoyar
confesar	encontrar	preparar
avisar	arreglar	admirar
sorprender	venir	vender
fue	llenar	nohaber
buscar	pagar	preguntar
hablar	examinar	vaciar
calmar	contestar	esconder
ganar	regresar	esperar
convertirse	desear	tener
bañarse	robar	necesitar

respetar	llover	quedar
responder	salir	cantar
faltar	controlar	

Tabla 5 Conjunto I

sentir	casar	recibir
nogustar	regalar	dibujar
amar	separar	probar
nadar	asustar	pensar
descansar	fracasar	odiar
comprender	liberar	mostrar
arrojar	engañar	va
estar	guiar	enfermar
funcionar	bailar	contagiar
creer	comunicar	sufrir
saber	dejar	declarar
comenzar	defender	comer
cuidar	escoger	leer
acabar	brillar	escuchar
aconsejar	confiar	memorizar
hundir	terminar	crecer
caer	equivocar	

Tabla 6 Conjunto II

acompañar	apurar	poner
ensayar	pedir	matar
traducir	jugar	invitar
reír	copiar	levantar
oler	gritar	detener
pintar	caminar	despedir
cocinar	educar	noquerer
llegar	subir	aprender
quitar	cancelar	contar
vivir	cerrar	parar
viajar	perder	desaparecer
usar	morir	decidir
restar	decir	correr
servir	poder	escribir
mentir	proteger	imaginar
enviar	aceptar	alcanzar
pasar	doler	

Tabla 7 Conjunto III

dar	conquistar	prometer
dudar	verbos	aprovechar
acostrumbrar	despertar	permitir
inventar	aguantar	noentender
romper	corregir	abandonar
confundir	componer	comprometer
aumentar	recordar	distraer
trabajar	no poder	nacer
construir	sumar	comprar

enseñar	coser	abrir
noservir	mirar	visitar
cumplir	idear	interpretar
adivinar	cooperar	conseguir
soñar	intentar	descubrir
gustar	ahorrar	mover
participar	hacer	tratar
escapar	deletrear	

Tabla 8 Conjunto IV

guardar	sembrar	elegir
explicar	confirmar	adorar
nosaber	organizar	continuar
aparecer	herir	conocer
discutir	competir	besar
pesar	noconocer	convencer
llevar	llorar	castigar
avanzar	deber	coger
significar	bajar	beber
apagar	decepcionar	prestar
limpiar	prender	arrepentir
rentar	cambiar	perdonar
repetir	atender	haber
agarrar	mejorar	extrañar
seleccionar	platicar	

Tabla 9 Conjunto V

Para evitar inconsistencias creé una tabla **Tlanextistli** para cada uno de los sets.

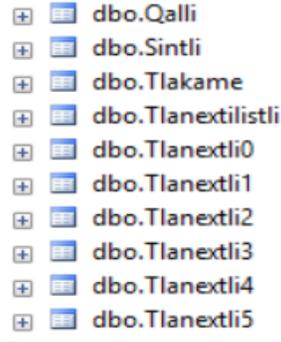


Ilustración 38. Tablas de la base de datos Tlahtolli de la aplicación MainMachiome

En la tabla **Tlanextli** se encuentra el vector de Probabilidades de Aparición de

la iteración anterior en el campo **Pantli**, una modificación no congruente con los datos que se están procesando generará que se atoren ítems y tengan comportamientos erráticos, que se corrigen hasta que el elemento es seleccionado nuevamente, pero pasan varias iteraciones antes que esto suceda.

	MachiotID	Tlamantli	Pantli	Kisa	Qalli
1	0	puntual	0.08000001	1	1
2	1	travieso	0.08000001	1	1
3	2	inteligente	0.08000001	1	1
4	3	ligero	0.08000001	1	1
5	4	absurdo	0.18	0	0
6	5	caliente	0.08000001	2	2
7	6	mamá	0.18	0	0
8	7	primo	0.08000001	0	0
9	8	nuevo	0.08000001	1	1
10	9	limpio	0.08000001	2	2

Ilustración 39. Contenido de una de las tablas Tlanextli

Y sus correspondientes señas:



Ilustración 40 Muestra de video introducido en la aplicación MainMachiome (Investigar)

Capitulo IV: Algoritmos

Comienzo este capítulo con unos dibujos que representa a grandes rasgos la infraestructura de la aplicación, para poder tener una idea de cómo están dispuestos los algoritmos.

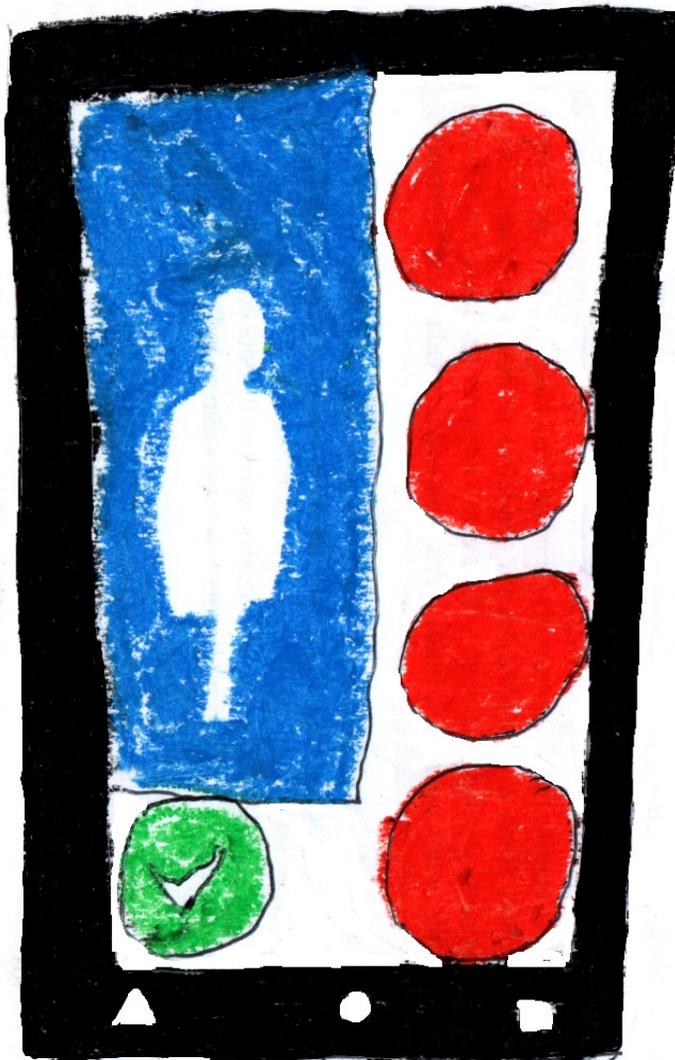


Ilustración 41. Pantalla de la aplicación

Muestro un pequeño esquema de la aplicación para señalar que, a pesar de no ser un algoritmo muy elaborado, hay toda una estructura muy completa para manipular los elementos que serán mostrados en cada uno de las plantillas visibles. Otro detalle es que la aplicación aparece en un celular, porque planeo programarla en Android para su distribución. En esta tesina se limita al

desarrollo de la aplicación en escritorio con CSharp, ya que el desarrollo de la aplicación para celular implica el resolver ciertos requerimientos que van más allá del problema planteado.

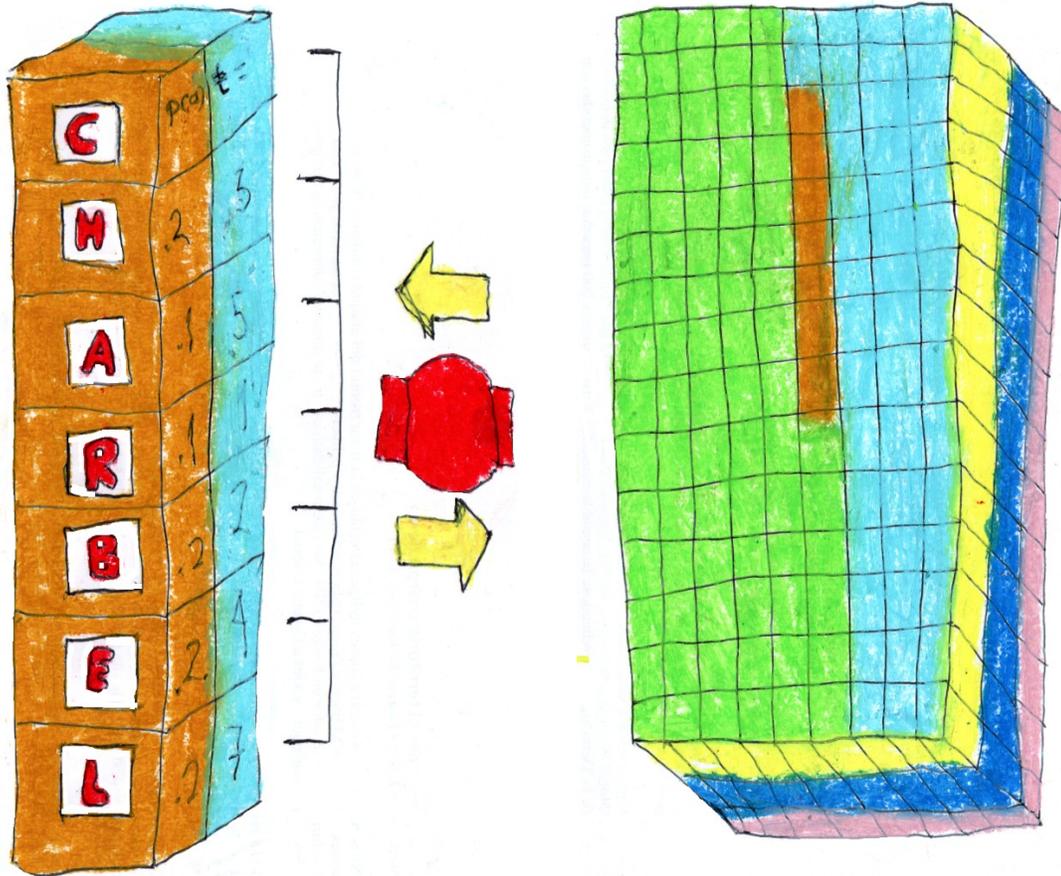


Ilustración 42 y 43. Representación del vector y del registro de cada elemento

En este trabajo desarrollé 4 algoritmos con diferentes propósitos. Definición de probabilidades(A1), Modificación de frecuencias instantáneas(A2), Selector de nuevos ítems(A3) y Modificación de frecuencias con base a la probabilidad de cada ítem(A4). En este par de imágenes si tuviéramos que ubicar a los algoritmos, A1 sería la definición de los bloques de la ilustración de la izquierda, A2, A3 y A4 se encontraría dentro del artefacto rojo que se encuentra en medio, en términos generales este aparato es el administrador que se encarga de los cambios en ambas direcciones entre el vector y el registro. A2 se encargaría de lidiar con los ítems que entran y salen del vector. A3 Se encargaría de manejar los ítems seleccionados de los bloques amarillos del registro de cada ítem. Por último, A4 se encarga de racionar el flujo de cada uno de los elementos del registro gracias al soporte que le ofrece la base de datos, bloques de color rosas. Un pequeño detalle que representé con la perspectiva es que el vector está por encima del registro.

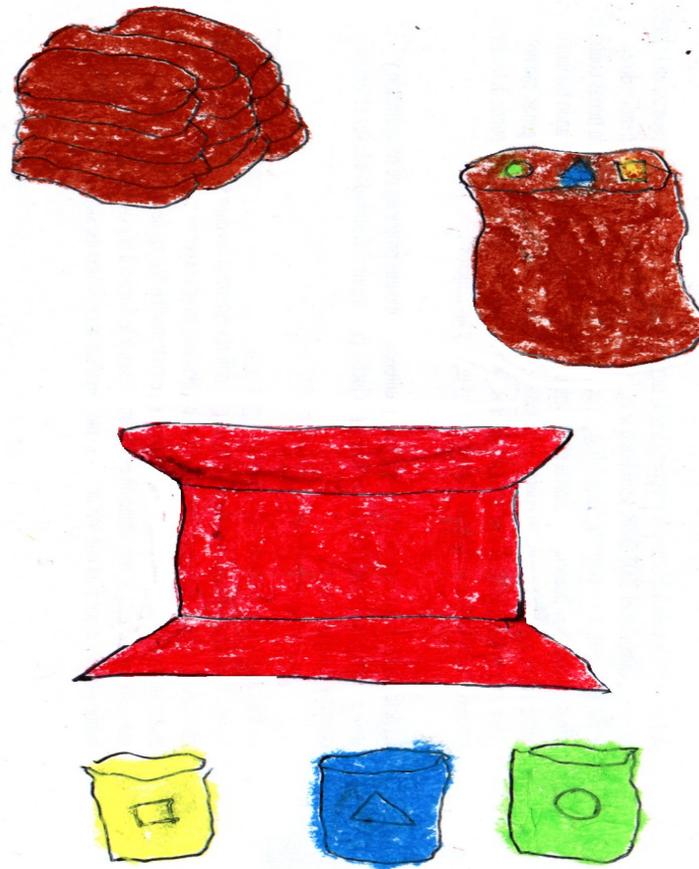


Ilustración 44. Representación del problema que quiero solucionar

Con el diagrama de la ilustración 44 quiero mostrar a grandes rasgos la solución propuesta.

Tenemos unos costales en la parte superior, lo que representan es un conjunto muy grande de diferentes elementos, abajo podemos observar una máquina la cual tiene 2 objetivos. El primero es sacar el mismo número de elementos de una clase, en la ilustración, cuadros, triángulos y círculos para, posteriormente poder elegir a placer la proporción correspondiente de cada costal con respecto de los demás. En la imagen se representaría como hacer más pequeño o más grande los costales de cada elemento.

Al respecto del primer objetivo parecería que la máquina tiene la función de retener los elementos, pero ésta es una ilusión. El truco de producir el mismo número de elementos de cada costal consiste en poner el mismo número de elementos mezclados en los costales, como esto implicaría un número finito de elementos, la solución consiste en poner el número de elementos en los costales a competir y teóricamente alcanzar el mismo número de elementos en el infinito.

La verdadera función de la maquina consisten, en primer término, la dosificación de los elementos para aparentar una selección aleatoria en el flujo a través de compensaciones, y en segundo, el control de las proporciones desde los costales, aunque este último no está del todo bajo control, ya que como se verá más adelante, no llegué a manipular las proporciones a voluntad.

4.1 Simulación.

El objetivo de la simulación fue generar una secuencia a partir del mismo número de elementos (en la aplicación los elementos van cambiando) y una respuesta del usuario similar a las obtenidas por una aplicación con las mismas condiciones. Esto sirvió para 2 propósitos, el primero para la construcción de propuestas de algoritmos y el segundo, poder manipular las probabilidades del simulador.

El código de la última versión de la simulación es:

Números aleatorios para los elementos seleccionados.

B3 = ALEATORIO()

C3 = ALEATORIO()

D3 = ALEATORIO()

E3 = ALEATORIO()

F3 = ALEATORIO()

	1	2	3	4	5
1	0.70	0.86	0.95	0.50	0.71
2	0.58	0.74	0.25	0.41	0.63
3	0.61	0.09	0.74	0.58	0.87
4	0.16	0.45	0.25	0.78	0.98
5	0.91	0.46	0.57	0.73	0.09
6	0.08	0.11	0.96	0.71	0.53
7	0.63	0.67	0.62	0.05	0.94
8	0.98	0.95	0.04	0.90	0.62
9	0.09	0.72	0.24	0.31	0.55
10	0.86	0.63	0.86	0.52	0.45

Ilustración 45. Números aleatorios

Probabilidad de cada elemento.

G3 = .2

H3 = .2

I3 = .2

J3 = .2

K3 = .2

Pa	Pb	Pc	Pd	Pe
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.1975	0.21	0.1975	0.1975	0.2025
0.1963469	0.2086837	0.1963469	0.1963469	0.2077679

Ilustración 46. Probabilidad de cada elemento

Definición de Intervalos.

$$L3 = 0$$

$$M3 = L3 + G3$$

$$N3 = M3 + H3$$

$$O3 = N3 + I3$$

$$P3 = O3 + J3$$

$$Q3 = P3 + K3$$

0.2	0.4	0.6	0.8	1
0.2	0.4	0.6	0.8	1
0.2	0.4	0.6	0.8	1
0.2	0.4	0.6	0.8	1
0.2	0.4	0.6	0.8	1
0.2	0.4	0.6	0.8	1
0.2	0.4	0.6	0.8	1
0.2	0.4	0.6	0.8	1
0.2	0.4	0.6	0.8	1
0.1975	0.4075	0.605	0.8025	1.005
0.1963469	0.4050306	0.6013775	0.7977245	1.0054924

Ilustración 47. Intervalos de probabilidad

Secuencia mostrada por Computador (Simulado).

R3

SI(Y(B3>=\$L3,B3<\$M3),"a",SI(Y(B3>=\$M3,B3<\$N3),"b",SI(Y(B3>=\$N3,B3<\$O3),"c",SI(Y(B3>=\$O3,B3<\$P3),"d",SI(Y(B3>=\$P3,B3<\$Q3),"e",""))))))

S3 =

SI(Y(C3>=\$L3,C3<\$M3),"a",SI(Y(C3>=\$M3,C3<\$N3),"b",SI(Y(C3>=\$N3,C3<\$O3),"c",SI(Y(C3>=\$O3,C3<\$P3),"d",SI(Y(C3>=\$P3,C3<\$Q3),"e",""))))))

T3 =

SI(Y(D3>=\$L3,D3<\$M3),"a",SI(Y(D3>=\$M3,D3<\$N3),"b",SI(Y(D3>=\$N3,D3<\$O3),"c",SI(Y(D3>=\$O3,D3<\$P3),"d",SI(Y(D3>=\$P3,D3<=\$Q3),"e",""))))))

U3 =

SI(Y(E3>=\$L3,E3<\$M3),"a",SI(Y(E3>=\$M3,E3<\$N3),"b",SI(Y(E3>=\$N3,E3<\$O3),"c",SI(Y(E3>=\$O3,E3<\$P3),"d",SI(Y(E3>=\$P3,E3<=\$Q3),"e",""))))))

V3 =

SI(Y(F3>=\$L3,F3<\$M3),"a",SI(Y(F3>=\$M3,F3<\$N3),"b",SI(Y(F3>=\$N3,F3<\$O3),"c",SI(Y(F3>=\$O3,F3<\$P3),"d",SI(Y(F3>=\$P3,F3<=\$Q3),"e",""))))))

1	2	3	4	5
d	e	e	c	d
c	d	b	c	d
d	a	d	c	e
a	c	b	d	e
e	c	c	d	a
a	a	e	d	c
d	d	d	a	e
e	e	a	e	d
a	d	b	b	c
e	d	e	c	c

Ilustración 48. Secuencia simulada

Números aleatorios para una respuesta alternativa

W3 = ALEATORIO()

X3 = ALEATORIO()

Y3 = ALEATORIO()

Z3 = ALEATORIO()

AA3 = ALEATORIO()

1	2	3	4	5
0.40	0.89	0.40	0.37	0.35
0.19	0.07	0.80	0.90	0.15
0.79	0.64	0.51	0.40	0.47
0.01	0.70	0.46	0.32	0.25
0.60	0.63	0.17	0.78	0.05
0.94	0.26	0.10	0.88	0.94
0.81	0.22	0.82	0.45	0.26
0.69	0.95	0.85	0.21	0.89
0.63	0.52	0.90	0.46	0.26
0.08	0.06	0.52	0.39	0.66

Ilustración 49. Números aleatorios

Generación de respuesta alternativa, el objetivo era identificar una probabilidad de un elemento dado que la respuesta era otro elemento. No llegué a realizar separación de tal índice, ni en la simulación, ni en la aplicación. Queda pendiente obtener esas probabilidades.

AB3 =
=SI(R3="a",SI(Y(W3>=0,W3<0.25),"b",SI(Y(W3>=0.25,W3<0.5),"c",SI(Y(W3>=0.5,W3<0.75),"d",SI(Y(W3>=0.75,W3<=1),"e","")))),SI(R3="b",SI(Y(W3>=0,W3<0.25),"a",SI(Y(W3>=0.25,W3<0.5),"c",SI(Y(W3>=0.5,W3<0.75),"d",SI(Y(W3>=0.75,W3<=1),"e","")))),SI(R3="c",SI(Y(W3>=0,W3<0.25),"a",SI(Y(W3>=0.25,W3<0.5),"b",SI(Y(W3>=0.5,W3<0.75),"d",SI(Y(W3>=0.75,W3<=1),"e","")))),SI(R3="d",SI(Y(W3>=0,W3<0.25),"a",SI(Y(W3>=0.25,W3<0.5),"b",SI(Y(W3>=0.5,W3<0.75),"c",SI(Y(W3>=0.75,W3<=1),"e","d")))),SI(R3="e",SI(Y(W3>=0,W3<0.25),"a",SI(Y(W3>=0.25,W3<0.5),"b",SI(Y(W3>=0.5,W3<0.75),"c",SI(Y(W3>=0.75,W3<=1),"d","")))),SI(R3="2"))))

AC3 =
SI(S3="a",SI(Y(X3>=0,X3<0.25),"b",SI(Y(X3>=0.25,X3<0.5),"c",SI(Y(X3>=0.5,X3<0.75),"d",SI(Y(X3>=0.75,X3<=1),"e","")))),SI(S3="b",SI(Y(X3>=0,X3<0.25),"a",SI(Y(X3>=0.25,X3<0.5),"c",SI(Y(X3>=0.5,X3<0.75),"d",SI(Y(X3>=0.75,X3<=1),"e","")))),SI(S3="c",SI(Y(X3>=0,X3<0.25),"a",SI(Y(X3>=0.25,X3<0.5),"b",SI(Y(X3>=0.5,X3<0.75),"d",SI(Y(X3>=0.75,X3<=1),"e","")))),SI(S3="d",SI(Y(X3>=0,X3<0.25),"a",SI(Y(X3>=0.25,X3<0.5),"b",SI(Y(X3>=0.5,X3<0.75),"c",SI(Y(X3>=0.75,X3<=1),"e","d")))),SI(S3="e",SI(Y(X3>=0,X3<0.25),"a",SI(Y(X3>=0.25,X3<0.5),"b",SI(Y(X3>=0.5,X3<0.75),"c",SI(Y(X3>=0.75,X3<=1),"d","")))),SI(S3="2"))))

AD3 =
SI(T3="a",SI(Y(Y3>=0,Y3<0.25),"b",SI(Y(Y3>=0.25,Y3<0.5),"c",SI(Y(Y3>=0.5,Y3<0.75),"d",SI(Y(Y3>=0.75,Y3<=1),"e","")))),SI(T3="b",SI(Y(Y3>=0,Y3<0.25),"a",SI(Y(Y3>=0.25,Y3<0.5),"c",SI(Y(Y3>=0.5,Y3<0.75),"d",SI(Y(Y3>=0.75,Y3<=1),"e","")))),SI(T3="c",SI(Y(Y3>=0,Y3<0.25),"a",SI(Y(Y3>=0.25,Y3<0.5),"b",SI(Y(Y3>=0.5,Y3<0.75),"d",SI(Y(Y3>=0.75,Y3<=1),"e","")))),SI(T3="d",SI(Y(Y3>=0,Y3<0.25),"a",SI(Y(Y3>=0.25,Y3<0.5),"b",SI(Y(Y3>=0.5,Y3<0.75),"c",SI(Y(Y3>=0.75,Y3<=1),"e","d")))),SI(T3="e",SI(Y(Y3>=0,Y3<0.25),"a",SI(Y(Y3>=0.25,Y3<0.5),"b",SI(Y(Y3>=0.5,Y3<0.75),"c",SI(Y(Y3>=0.75,Y3<=1),"d","")))),SI(T3="2"))))

AE3 =

0.67	0.68	0.99	0.34	0.79
0.09	0.21	0.54	0.36	0.57
0.19	0.02	0.18	0.32	0.01
0.21	0.69	0.13	0.42	0.15
0.20	0.74	0.84	0.60	0.10
0.73	0.77	0.74	0.76	0.04
0.08	0.07	0.11	0.99	0.44
0.33	0.53	0.93	0.95	0.58
0.90	0.17	0.96	0.35	0.24
0.43	0.00	0.48	0.22	0.60

Ilustración 51. Números aleatorios

Probabilidad de acierto definida (Manipulación).

AM3 = 0.9

AN3 = 0.9

AO3 = 0.6

AP3 = 0.2

AQ3 = 0.2

Pacierto	Pacierto	Pacierto	Pacierto	Pacierto
0.9	0.9	0.6	0.2	0.2
0.9	0.9	0.6	0.2	0.2
0.9	0.9	0.6	0.2	0.2
0.9	0.9	0.6	0.2	0.2
0.9	0.9	0.6	0.2	0.2
0.9	0.9	0.6	0.2	0.2
0.9	0.9	0.6	0.2	0.2
0.9	0.9	0.6	0.2	0.2
0.9	0.9	0.6	0.2	0.2
0.9	0.9	0.6	0.2	0.2
0.9	0.9	0.6	0.2	0.2

Ilustración 52. Probabilidad de acierto

Respuesta del usuario.

AR3

=

SI(Y(R3="a",AG3<=\$AM3),R3,SI(Y(R3="b",AG3<=\$AN3),R3,SI(Y(R3="c",AG3<=\$AO3),R3,SI(Y(R3="d",AG3<=\$AP3),R3,SI(Y(R3="e",AG3<=\$AQ3),R3,AB3))))))

AS3 =

SI(Y(S3="a",AH3<=\$AM3),S3,SI(Y(S3="b",AH3<=\$AN3),S3,SI(Y(S3="c",AH3<=\$AO3),S3,SI(Y(S3="d",AH3<=\$AP3),S3,SI(Y(S3="e",AH3<=\$AQ3),S3,AC3))))))

AT3 =

SI(Y(T3="a",AI3<=\$AM3),T3,SI(Y(T3="b",AI3<=\$AN3),T3,SI(Y(T3="c",AI3<=\$AO3),T3,SI(Y(T3="d",AI3<=\$AP3),T3,SI(Y(T3="e",AI3<=\$AQ3),T3,AD3))))))

AU3 =

SI(Y(U3="a",AJ3<=\$AM3),U3,SI(Y(U3="b",AJ3<=\$AN3),U3,SI(Y(U3="c",AJ3<=\$AO3),U3,SI(Y(U3="d",AJ3<=\$AP3),U3,SI(Y(U3="e",AJ3<=\$AQ3),U3,AE3))))))

AV3 =

SI(Y(V3="a",AK3<=\$AM3),V3,SI(Y(V3="b",AK3<=\$AN3),V3,SI(Y(V3="c",AK3<=\$AO3),V3,SI(Y(V3="d",AK3<=\$AP3),V3,SI(Y(V3="e",AK3<=\$AQ3),V3,AF3))))))

	1	2	3	4	5
	b	d	b	c	b
	c	a	b	c	a
	d	a	d	c	e
	a	d	b	b	e
	e	d	a	e	a
	a	a	a	e	c
	d	d	d	c	b
	c	d	e	a	e
	a	d	e	b	c
	a	d	c	c	c

Ilustración 53. Respuesta de usuario (simulado)

Compara la respuesta propuesta por la computadora con la generada por el usuario (simulado).

AW3 = SI(AR3=R3,AR3,"")

AX3 = SI(AS3=S3,AS3,"")

AY3 = SI(AT3=T3,AT3,"")

AZ3 = SI(AU3=U3,AU3,"")

BA = SI(AV3=V3,AV3,"")

			c	
c		b	c	
d	a	d	c	e
a		b		e
e				a
a	a			c
d	d	d		
a	d		b	c
	d		c	c
	a		a	a

Ilustración 54. Similitudes entre propuesta generada por la computadora y la respuesta del usuario simulado

4.2 Algoritmo 1: definición de probabilidades

Este fue uno de los algoritmos más costosos del proceso, ya que implicó el diseño de la mayor parte de la estructura que soporta al juego. Para llegar a él, lo primero que hice fue un algoritmo para simular la respuesta de acierto y dependiendo de la probabilidad obtenida generar el algoritmo de definición de probabilidades.

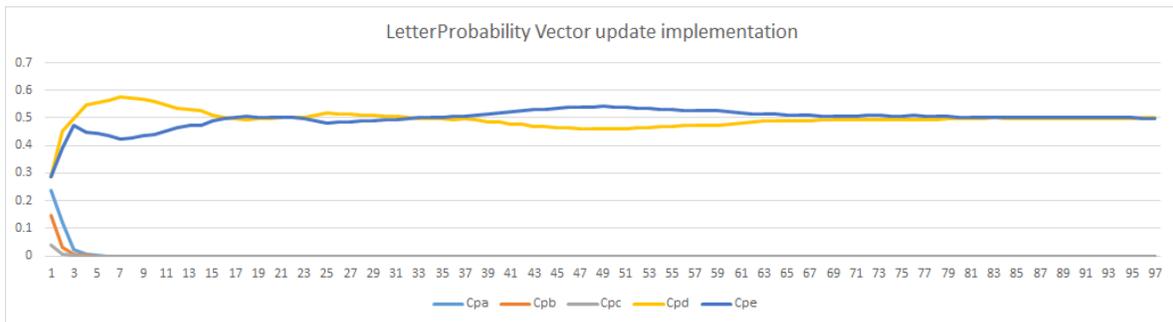


Ilustración 55. Implementación actualizada del vector de probabilidades de letras

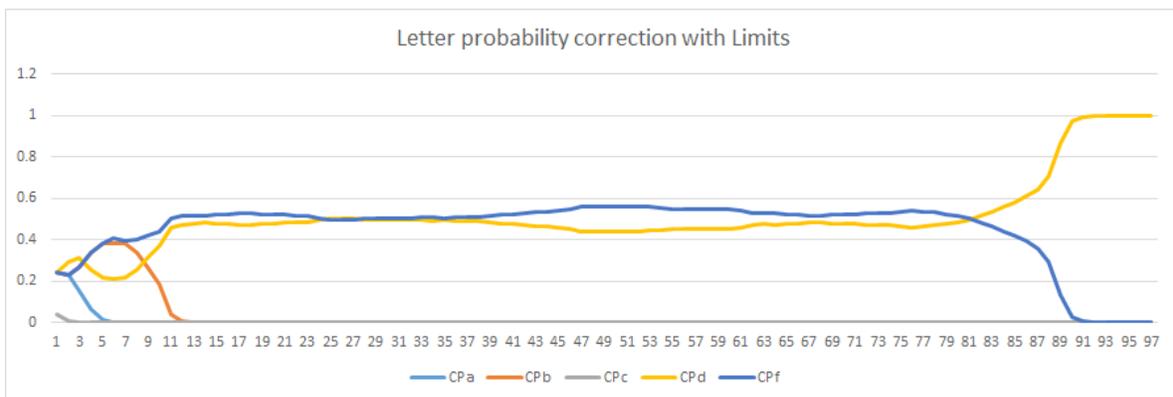


Ilustración 56. Corrección de probabilidades de las letras con límites

En esta imagen implemento un algoritmo de corrección directamente a las probabilidades creando un sesgo irreversible. Posteriormente implementé el algoritmo en la aplicación, la única diferencia es que no mantengo las probabilidades de todo el conjunto, sino manejo las probabilidades independientes por un periodo de tiempo. Para aclarar, la función de este algoritmo es registrar la probabilidad de los elementos seleccionados, podría decir que en teoría es muy sencillo, sin embargo, consiste en el esqueleto de la aplicación.

Las siguientes pruebas eran para comprobar que el algoritmo estaba reconociendo las probabilidades que se introducían.

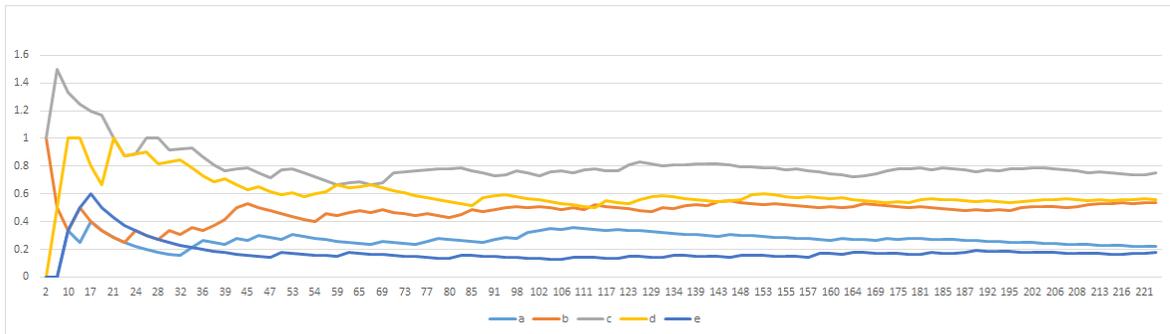


Ilustración 57. Calibración, vector (.2 .5 .8 .5 .2)

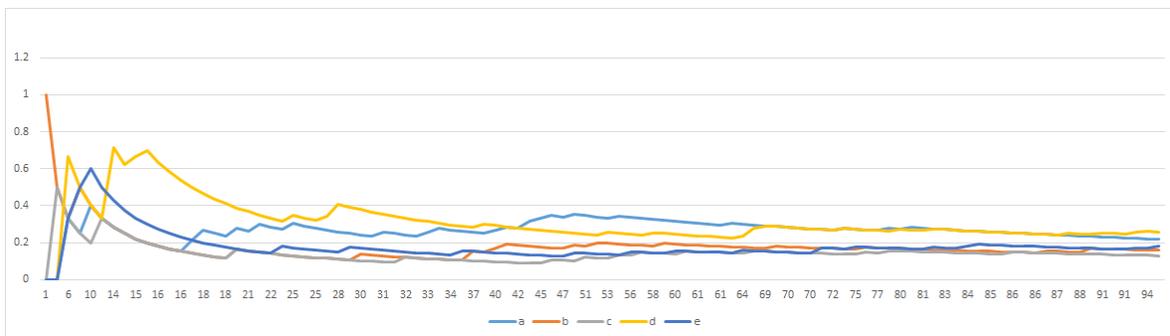


Ilustración 58. Calibración, vector (.8 .8 .6 .4 .4)

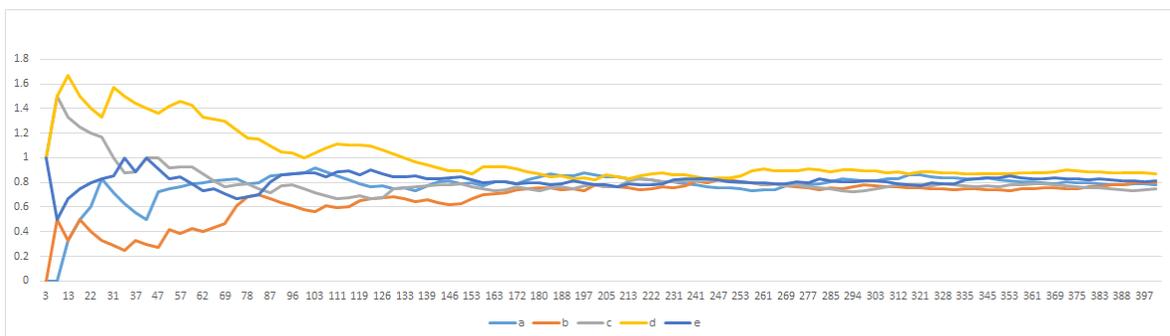


Ilustración 59. Calibración, vector (.8 .8 .8 .8 .8)

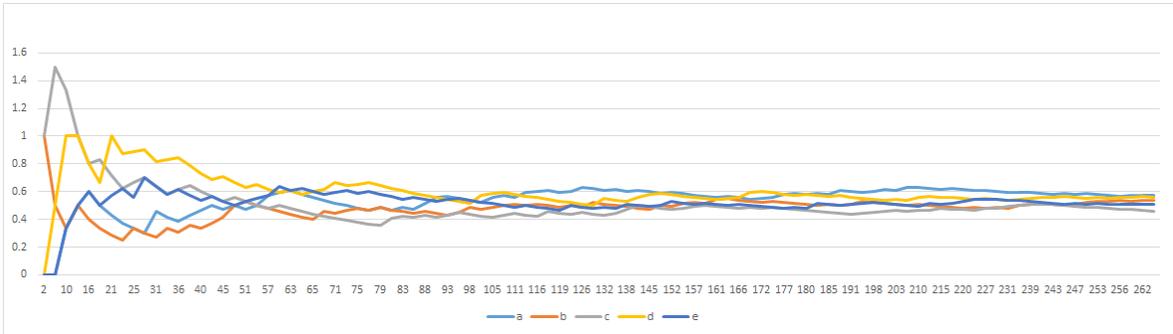


Ilustración 60. Calibración, vector (.2 .2 .2 .2)

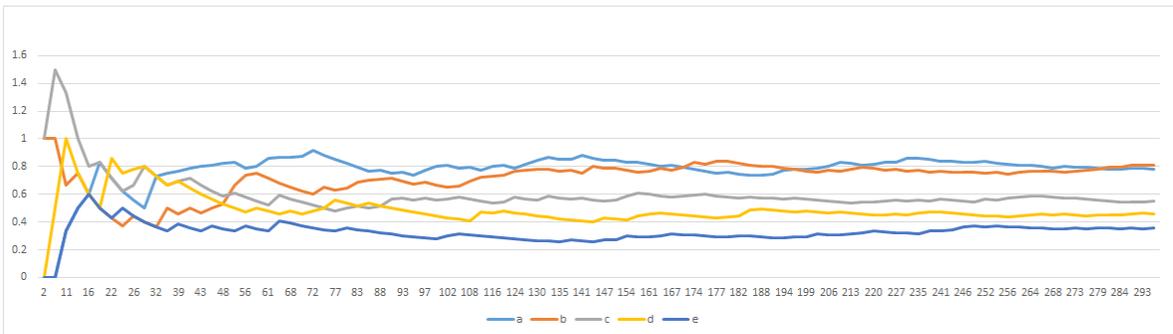


Ilustración 61. Calibración, vector (.5 .5 .5 .5)

Esto me permitía asegurar que mi objetivo era sensato. Y una referente de lo mínimo que debía esperar.

En la hoja de cálculo 5Selection se modela la estructura del selector 5 de 10, Excel ya no permite modificaciones fácilmente.

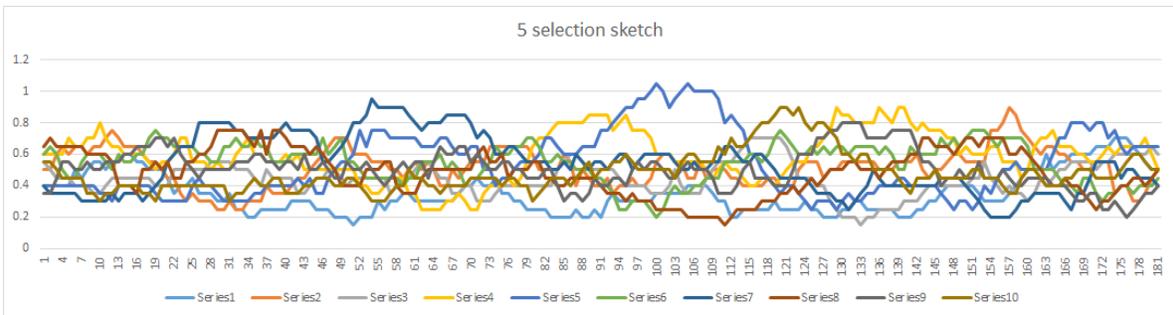


Ilustración 62. Bosquejo de selección de 5 elementos

El siguiente gráfico es la implementación final del algoritmo con una selección de 10 elementos sin llegar a implementar todo el algoritmo, el objetivo era esbozar el algoritmo para que fuese extensible.

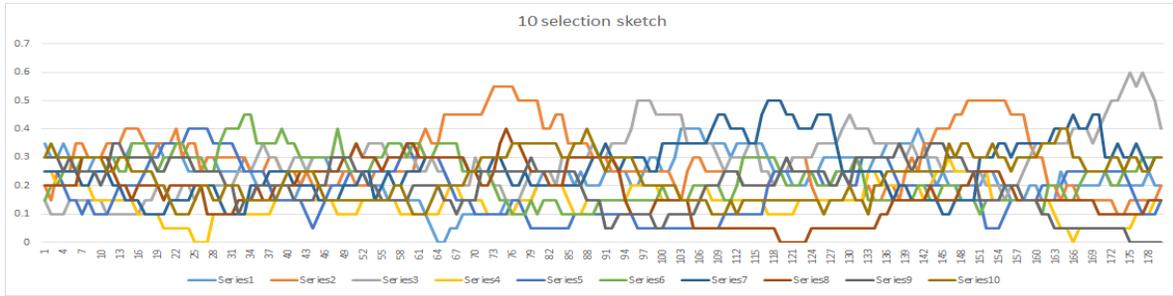


Ilustración 63. Bosquejo de selección de 10 elementos

4.3 Algoritmo 2: Modificación de frecuencias instantáneas de una selección de ítem.

El objetivo es aumentar las frecuencias de los ítems con menor probabilidad de acierto.

Lo más lejos que se pudo llegar con el algoritmo en Excel fue a retroalimentar con la varianza.

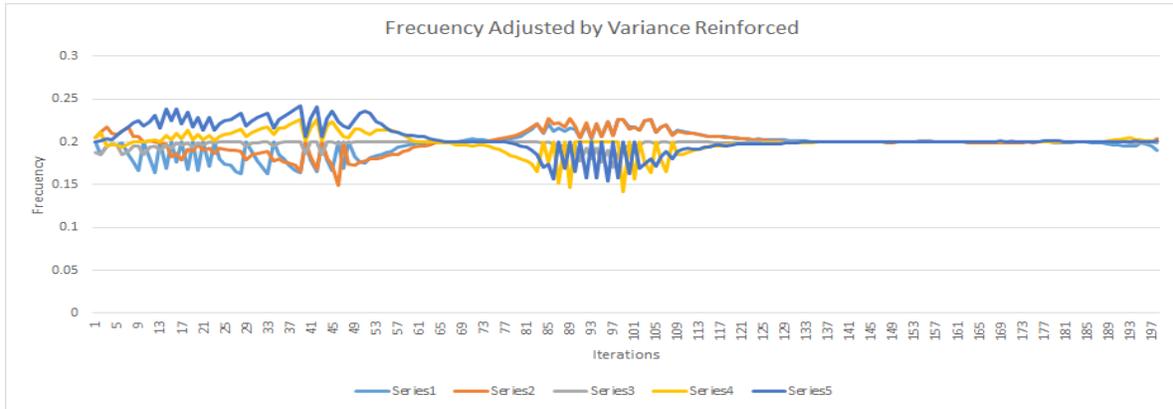


Ilustración 64. Frecuencia ajustada por refuerzo de varianza

Aquí apliqué una combinación lineal de la varianza con la probabilidad obtenida para modificar la probabilidad de cada elemento, pero resultó ser muy sensible en el sentido que modificaba las categorías muy vertiginosamente. Cabe destacar que aquí el algoritmo se retroalimenta.

A pesar que las gráficas son muy similares en apariencia, sus ejes x son diferentes, en las anteriores eran probabilidades de salida, y en este caso son probabilidades de entrada.

Este algoritmo no se implementó en la aplicación, pero implicó un esfuerzo para identificar las características que se tomarían en cuenta en el algoritmo definitivo. El objetivo era beneficiar la probabilidad más baja y reducir las diferencias cuando las probabilidades sean iguales, para que los elementos con los que fallaba más el usuario aparecieran más veces.

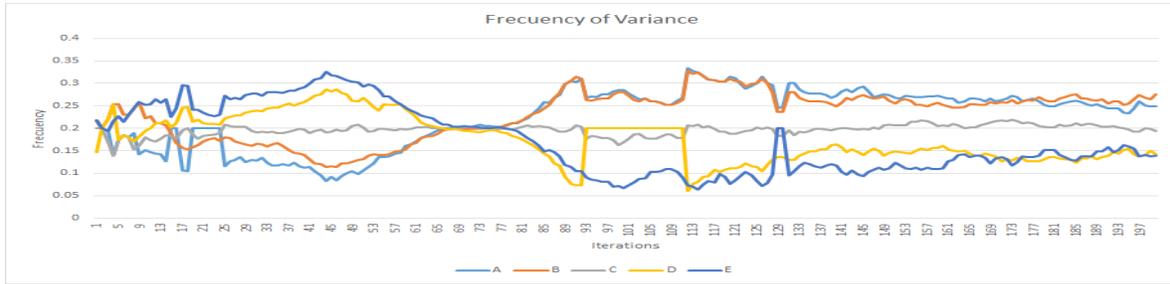


Ilustración 65. Frecuencia de Varianza

No llegué a hacer una implementación en C# del algoritmo desarrollado en Excel, ya que el algoritmo en sí no cumplía los requerimientos deseados, por otro lado, realizar modificaciones en la simulación de Excel implicaba la rescritura de todo el proceso. Así que realicé directamente sobre una tupla las operaciones que deseaba y pasé el registro de los datos a una BD en SQL Server 2016.

En el siguiente cuadro se plasman las operaciones tanto del algoritmo implementado en Excel como el de C# sobre el vector de probabilidades en un tiempo t. En otras palabras, es la vista transversal del proceso longitudinal que se observa en las simulaciones.

Modelo implementado con el simulador.

1	0	0		0	0	0.10	0.1
0.8888	0.1112	0.04010965	0.04010965	-0.15989035	0.15989035	0.10	0.07443508
0.8388	0.1612	0.05814457	0.05814457	-0.14185543	0.14185543	0.10	0.07987704
0	1	0.36069831	0.36069831	0.16069831	0.16069831	0.10	0.12582395
1	0	0		0	0	0.10	0.1
0.5	0.5	0.18034916	0.18034916	-0.01965084	0.01965084	0.10	0.09961384
0	1	0.36069831	0.36069831	0.16069831	0.16069831	0.10	0.12582395
1	0	0		0	0	0.10	0.1
1	0	0		0	0	0.10	0.1
1	0	0		0	0	0.10	0.1
2.7724	1	0.2					1.00557385

Tabla 10. Operaciones aplicadas a un vector con el proceso del simulador

Rediseño del modelo.

			0.00	0.00	0.00	
0.99999999	1E-08	0.1	0.00	0.1	0	0.1
0.99999999	1E-08	0.1	0.00	0.1	0	0.1
0.99999999	1E-08	0.1	0.00	0.1	0	0.1
0.99999999	1E-08	0.1	0.00	0.1	0	0.1
0.99999999	1E-08	0.1	0.00	0.1	0	0.1
0.99999999	1E-08	0.1	0.00	0.1	0	0.1
0.99999999	1E-08	0.1	0.00	0.1	0	0.1
0.99999999	1E-08	0.1	0.00	0.1	0	0.1
0.99999999	1E-08	0.1	0.00	0.1	0	0.1
0.99999999	1E-08	0.1	0.00	0.1	0	0.1
	1E-07	0.1	0.00			1

Tabla 11. Operaciones aplicadas a un vector con el nuevo algoritmo

Entonces implementé el algoritmo en la aplicación Machiotl.

El algoritmo final quedó así.

```

//NORMALIZED avoiding 0 DIVITION.
for (int i = 0; i < Tlatlakolli.Length; i++)
{
    if (SenmochiwaCe != 0) Anelkisa[i] = Tlatlakolli[i] /
SenmochiwaCe;
    else
        Anelkisa[i] = 0;
    //if (Anelkisa[i] != 0) Xiktli++;
    //NochinAnelkisa += Anelkisa[i];
}
//Average
//Tlahko = NochinAnelkisa / Xiktli;
Tlahko = Anelkisa.Average();
//Centered
for (int i = 0; i < Anelkisa.Length; i++)
{
    Tlahkopan[i] = Anelkisa[i] - Tlahko;
}
//Qepki = Tlahkopan.Max() - Tlahkopan.Min();
Qepki = (float).2;
for (int i = 0; i < Tlanextli.Length; i++)
{
    Tlanextli[i] = (float)1 / 10 + Tlahkopan[i] * Qepki;
    Sen += Tlanextli[i];
}
for (int i = 0; i < Tlanextli.Length; i++)
{
    Tlanextli[i] = Tlanextli[i] / Sen;
}

```

A continuación, se describe gráficamente qué sucede en cada paso del algoritmo.

Tlatlakolli (lo relativo en esencia a lo relativo que no es verdad, error) Que no es un error en el sentido matemático, sino el registro de la probabilidad de haber fallado. La Probabilidad de no haber acertado (**Tlatlakolli**) se calcula fácilmente, restando la Probabilidad de acierto (**Nelkisa**) a 1:

$$[\text{Tlatlakolli}] = 1 - [\text{Nelkisa}]$$

$$P_{\text{error}} = 1 - P_{\text{acierto}};$$

Nelkisa es el vector más importante, ya que es la Probabilidad de acierto que se calcula a través de dividir el rastro de Apariciones(Kisa) entre el número de aciertos(Qalli) de la tabla Tlanextli(Lo que aparece).

$$\text{Nelkisa} = \text{Qalli} / \text{Kisa}$$

Cabe destacar que realiza una recolección de datos supuestamente aleatorios, los datos son acomodados por el DBM, proceso que se lleva a cabo entre las tablas Sintli y Tlanextli. Esta tarea, aunque simple es la versión final del equivalente del algoritmo de la hoja de cálculo AnotherTry.

La estructura del vector final es la que tiene en la tabla Tlanextli. Es muy importante entender que no se tiene registrado un valor de Nelkisa porque cada iteración se actualiza y hay que recalcular el vector. En cierto sentido utilizamos variables de desecho que vamos registrando en el log.

	MachiotID	Tlamantli	Pantli	Kisa	Qalli
1	0	puntual	0.08000001	1	1
2	1	travieso	0.08000001	1	1
3	2	inteligente	0.08000001	1	1
4	3	ligero	0.08000001	1	1
5	4	absurdo	0.18	0	0
6	5	caliente	0.08000001	2	2
7	6	mamá	0.18	0	0
8	7	primo	0.08000001	0	0
9	8	nuevo	0.08000001	1	1
10	9	limpio	0.08000001	2	2

Ilustración 66. Tabla de DB Tlanextli, vector de probabilidades

Ahora vamos a ver el proceso paso por paso con un vector de entrada.

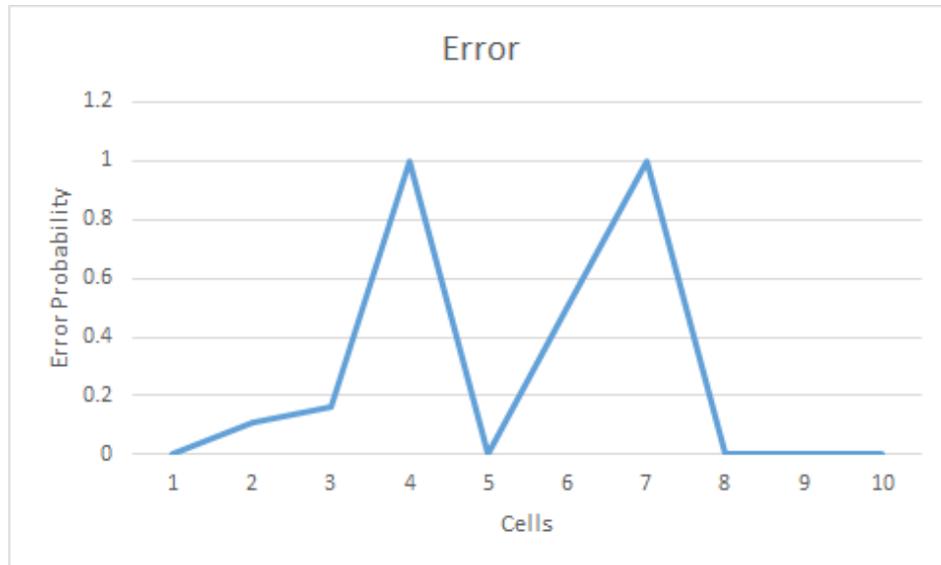


Ilustración 67. Vector de Error obtenido de la respuesta del usuario

Para mantener solo las **Proporciones** dividí entre la suma de todos los elementos. El objetivo es mantener el mayor número de afectación del error. Este es el equivalente del último algoritmo implementado en la versión de la última implementación en AnotherTry (Ilustración 65). Recalcando que, a pesar de ser la misma operación, la implementación en C# y la realizada en la simulación son diferentes.

Tenemos:

$$P_{\text{normalizada}} = P_{\text{error}} / \text{Suma}(P_{\text{error}});$$

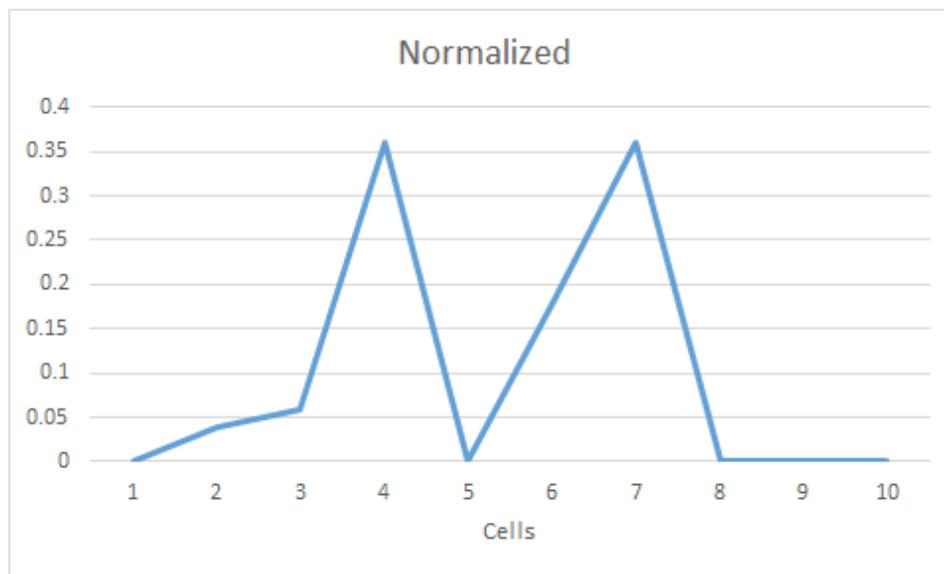


Ilustración 68. Vector normalizado

Procedo a quitarle un vector constante con .1, ya que la suma del vector normalizado es siempre 1 y para obtener la aportación equivalente en cada uno de los términos lo dividimos entre N, con N=10. Tenemos:

$$P_{\text{normalizada}} = e + E(P_{\text{normalizada}})$$

Donde e es el error de cada elemento.

$$\text{Como } E(P_{\text{normalizada}}) = \text{Suma}(P_{\text{normalizada}})/N = V[1/10]$$

$$P_{\text{normalizada}} = e + V[1/10]$$

Obtenemos:

$$e = P_{\text{normalizada}} - V[1/10]$$

Llamaremos a este vector e, Vm, vector de modificaciones.

Esto con el objetivo de centrar la gráfica y hacer la suma de todos los elementos igual a cero.

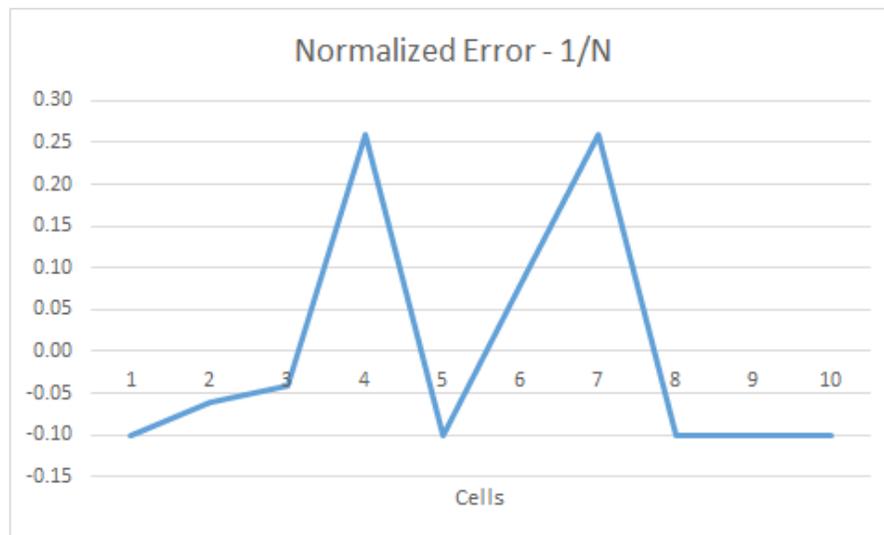


Ilustración 69. Vector normalizado centrado, vector de modificaciones

Añadí un factor constante, en este programa el escalar se llama Quepki y su valor es de .2. Multiplicamos con el vector de modificaciones por el factor y obtenemos el vector de modificaciones atenuado. Esto con el fin de modificar una pequeña parte de las probabilidades.

Tenemos:

$$V_{\text{atenuado}} = \text{Quepki} * V_{\text{modificaciones}}$$

$$V_{\text{atenuado}} = .2 * V_{\text{modificaciones}}$$

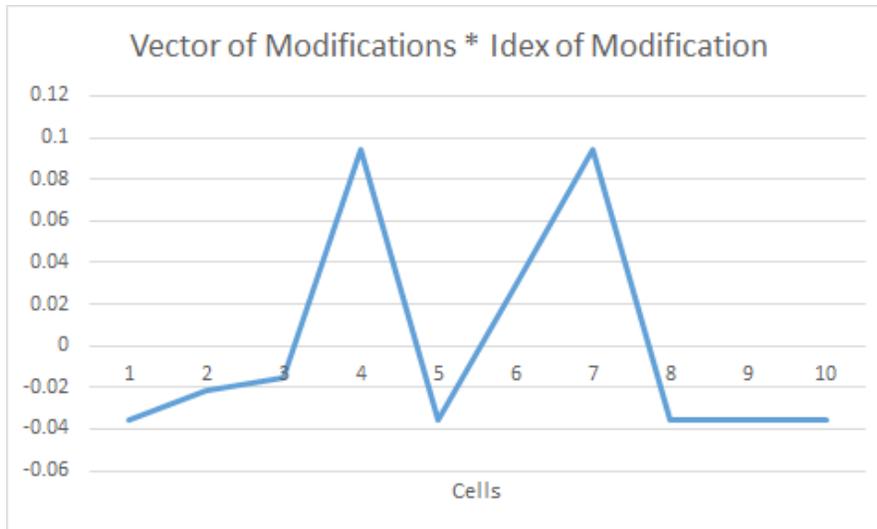


Ilustración 70. Vector de modificaciones ponderado

Finalmente sumamos a un vector fijo, el vector de modificaciones atenuado. Los valores del vector fijo son .1 con el objetivo de hacer a todos los valores equivalentes. El objetivo de sumar el vector de modificaciones es afectar las probabilidades del vector fijo de acuerdo al Error obtenido de las iteraciones anteriores. Un intento de este algoritmo se implementó en la hoja de cálculo. **Reinforced_NormPostCalculo.**

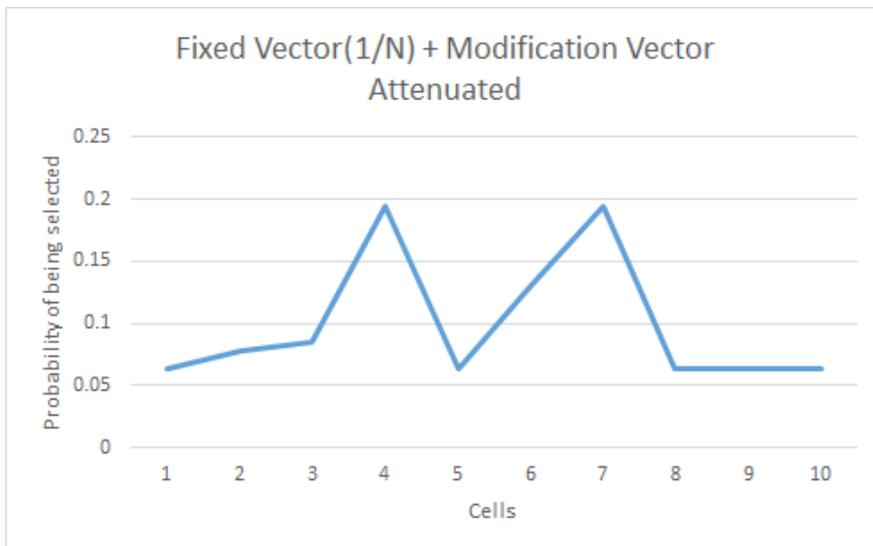


Ilustración 71. Vector de probabilidades

Aunque parece muy sencillo, el trabajo invertido fue considerable. En resumen, lo que hice fue adaptar el vector P_e a un vector media, esto con el objetivo de conservar la mayor proporción de error y en un medio controlado.

Cabe destacar que el algoritmo no toma la probabilidad de la iteración anterior como en las hojas de cálculo, sino que recibe la información del rastro de la probabilidad, con esto quiero decir que guarda una probabilidad por el tiempo que es seleccionada con el objetivo de obtener una P a través del tiempo. En cierto sentido son como variables desechables, ya que la información de las probabilidades será actualizada en la tabla Sintli y los datos de tal iteración en Log. Lo que quiero decir con esto es que la información será separada para ser analizada o modificada.

En este punto introduje el primer log. El archivo se guardaba en formato txt donde registraba el vector de probabilidades de selección cada vez que era modificado.

Log.txt

```

using (StreamWriter Tlaqilo = new StreamWriter(@"C:\Users\
Public\LOG".txt", true))
{
    Yeyekolli = Tlahtolmekatl.Tlahtolme[NikanTlapowalli] +
(char)9;
    Yeyekolli += tbTlahtolli.Text + (char)9;
    if (Qalli)
        Yeyekolli += "1" + (char)9;
    else
        Yeyekolli += "0" + (char)9;
    for(int i = 0;i<10;i++)
    {
        Yeyekolli += Tlahtolmekatl.Tlahtolme[i]+(char)9;
    }
    for(int i = 0; i < 10; i++)
    {
        Yeyekolli += KeskiQalli.Nelkisa[i]+ " ";
    }
    for (int i = 0; i < 10; i++)
    {
        Yeyekolli += Tlahtolmekatl.Axkan[i] + " ";
    }
    Tlaqilo.WriteLine(Yeyekolli);
}

```

Que produce la siguiente salida: A = Item

A
libre
gris
alegre
lleno
malo
alegre
alegre
lleno
alegre
amante

Ilustración 72. Columna de elementos seleccionados

Este es el primer registro que tengo del algoritmo en funcionamiento, como se puede notar las diferencias crecen irremediamente. No tengo un registro del algoritmo, porque se fue corrigiendo hasta el actual, quedando únicamente los logs. Como se puede observar el archivo se compone de 145 registros.

Después tuve que agregar otro grupo de campos.

0.8333333	0.9166667	1	0	0.9285714	0.9090909	1	0	1	0.75
0.8333333	0.9166667	1	0	0.9333333	0.9090909	1	0	1	0.75
0.8571429	0.9166667	1	0	0.9333333	0.9090909	1	0	1	0.75
0.875	0.9166667	1	0	0.9285714	0.9090909	1	0	1	0.75
0.875	0.9090909	1	0	0.9285714	0.9090909	1	0	1	0.8
0.875	0.8333333	1	0	0.9285714	0.9090909	1	0	1	0.8
0.8888889	0.8333333	1	0	0.9285714	0.9090909	1	0	1	0.8
0.8888889	0.8333333	1	0	0.9285714	0.9090909	1	1	1	0.8
0.8888889	0.8333333	1	0	0.9285714	0.9090909	1	1	1	0.8
0.8888889	0.8333333	1	0	0.9285714	0.9090909	1	1	1	0.8

Ilustración 77. Vector procesado

El objetivo era ver el vector de entrada y comparar con el vector de salida (por eso los colores).

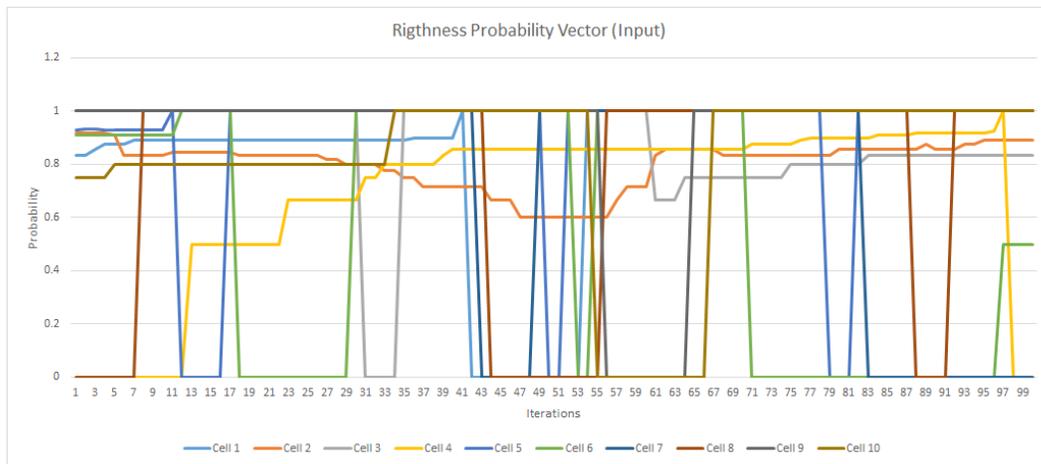


Ilustración 78. Probabilidad obtenida

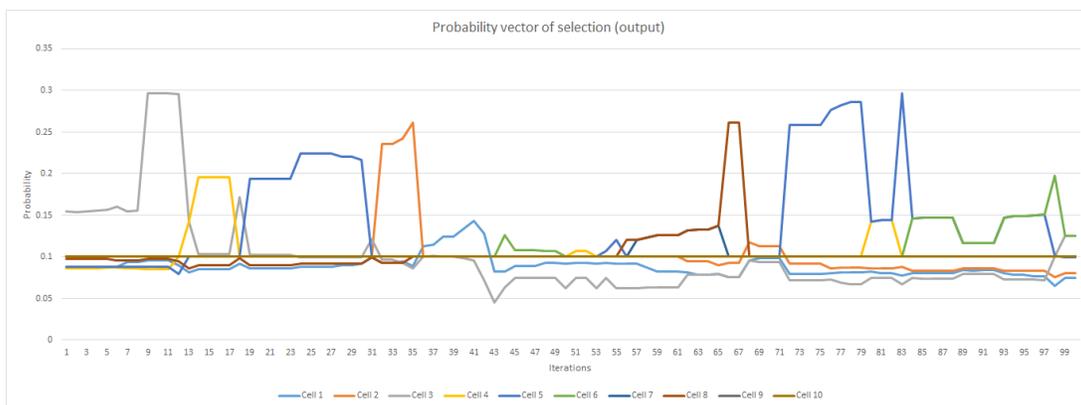


Ilustración 79. Probabilidad generada

Esta corrida fue corta, de 100 iteraciones, le siguió 2 corridas de 200 y 224 respectivamente. El objetivo era que la suma del Vector de probabilidades de selección diera 1; Cosa que no sucedía.

Corrección del método.

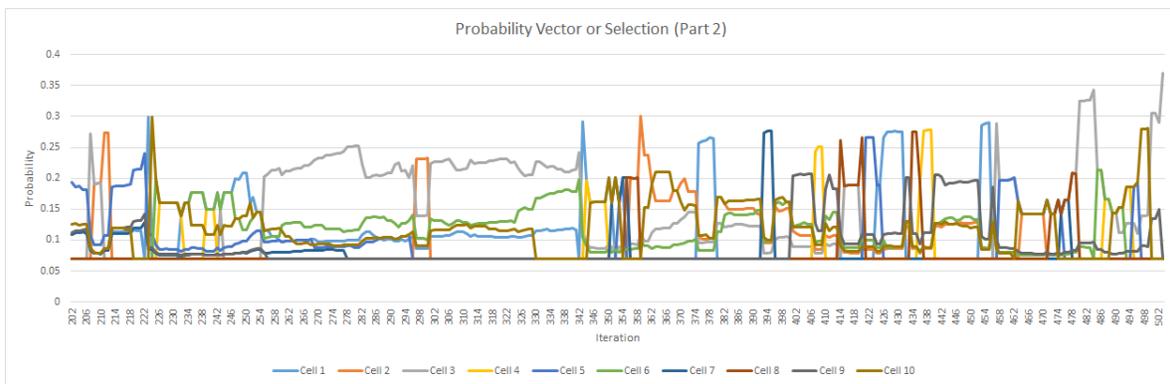
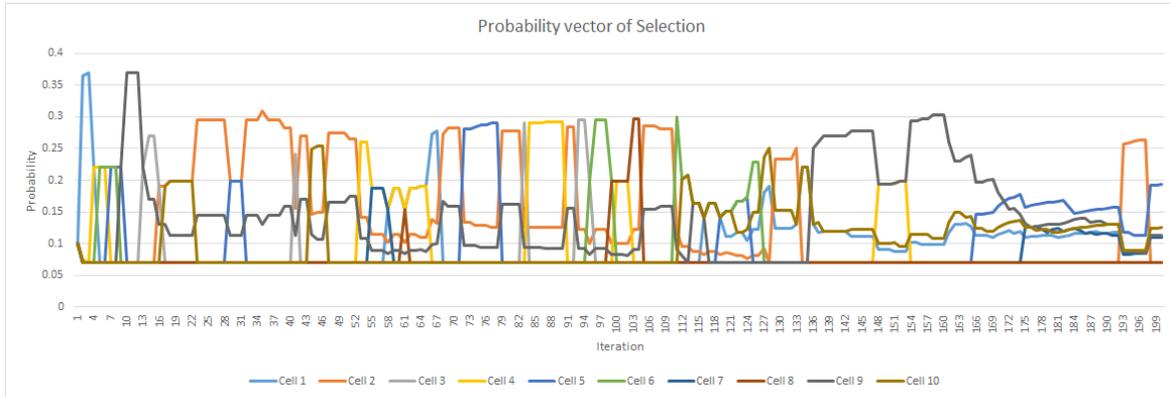
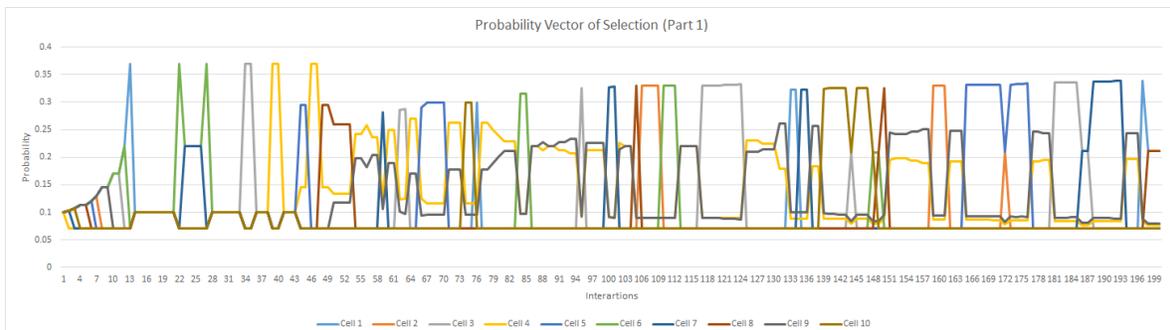
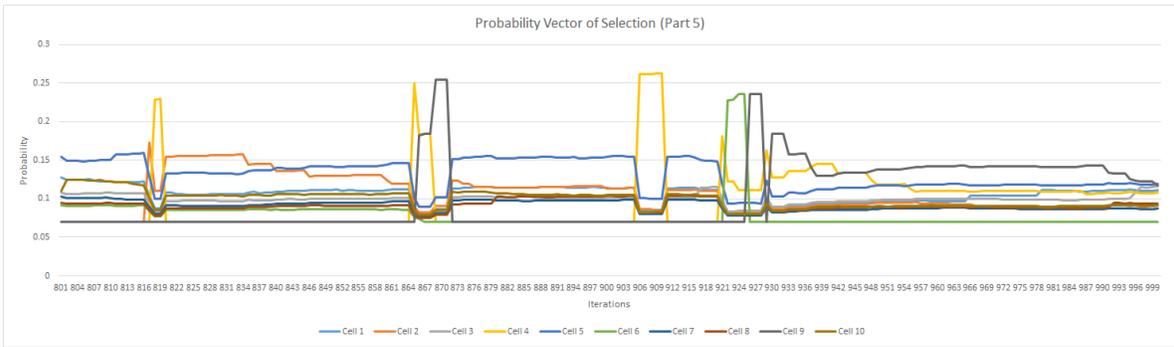
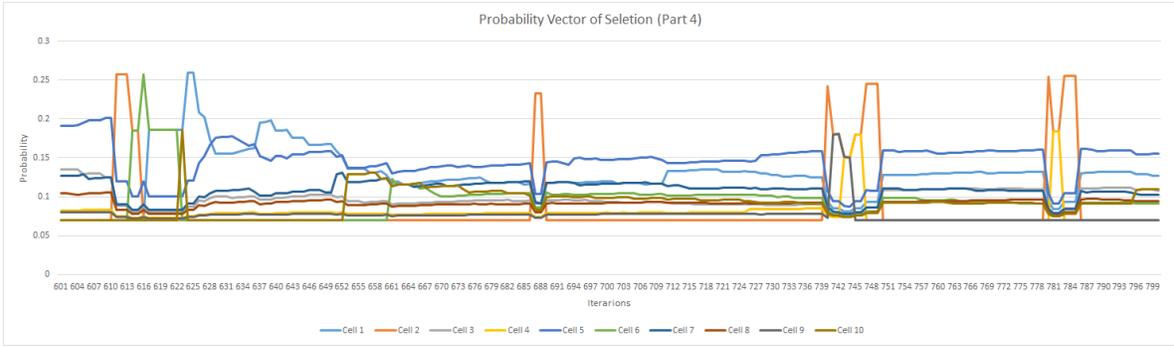
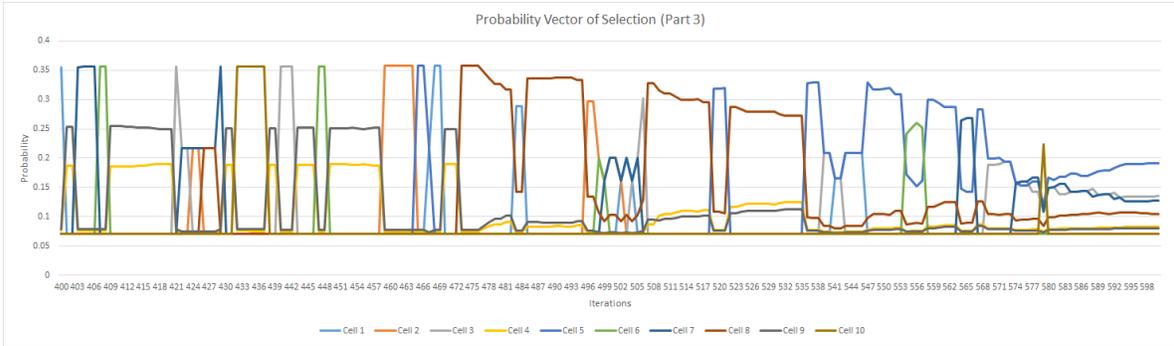
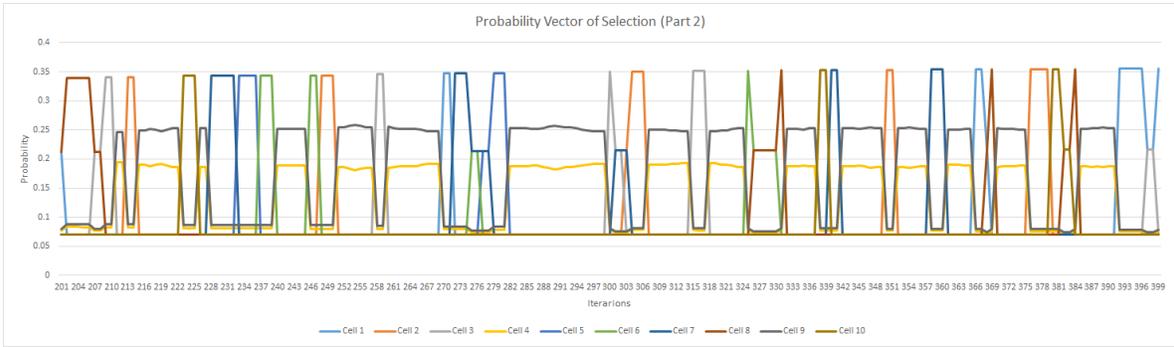


Ilustración 80. Registro de correcciones en el log

En esta corrida se puede notar unas anomalías a partir del 130 y que se corrigió hasta la iteración 370. En la imagen se puede observar una mejora en el comportamiento en la parte final de la gráfica.

Las siguientes fueron corridas llevadas a cabo para reestructurar toda la aplicación y verificar el buen funcionamiento del algoritmo.





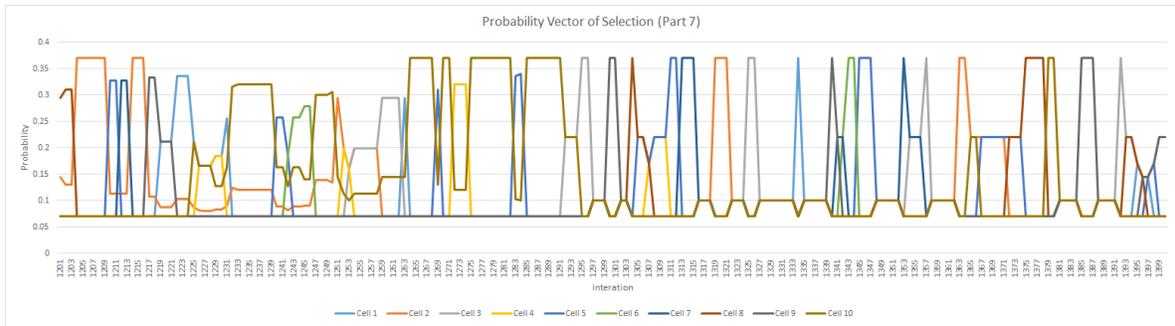
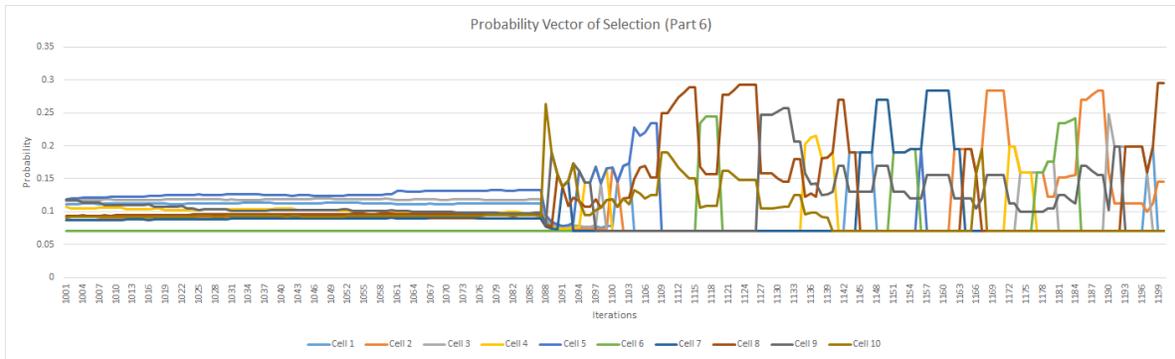
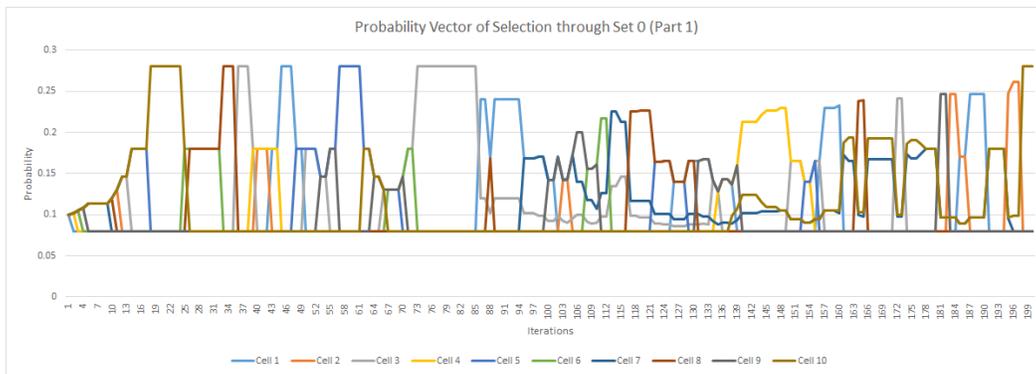
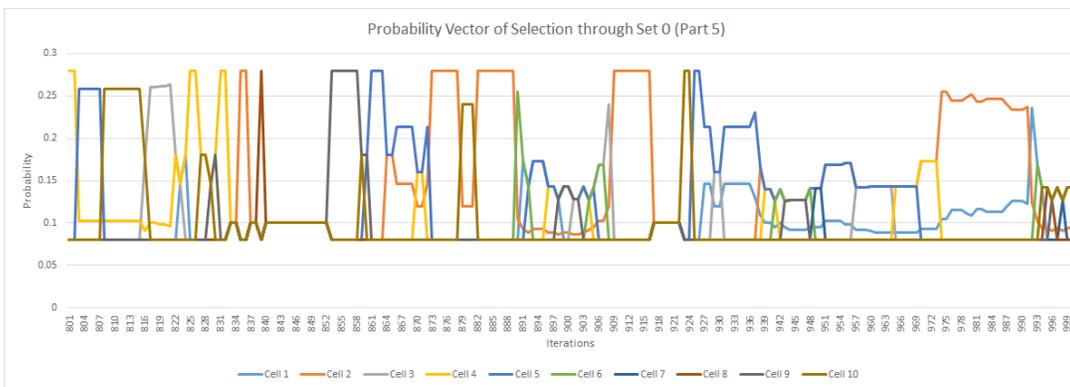
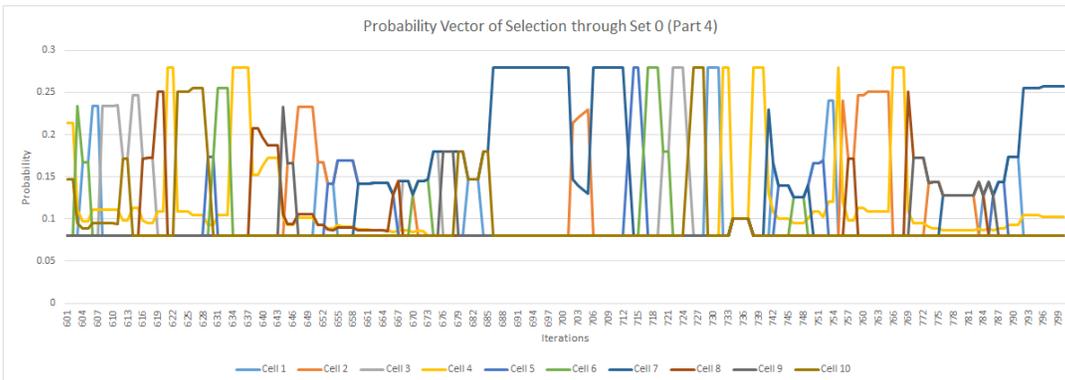
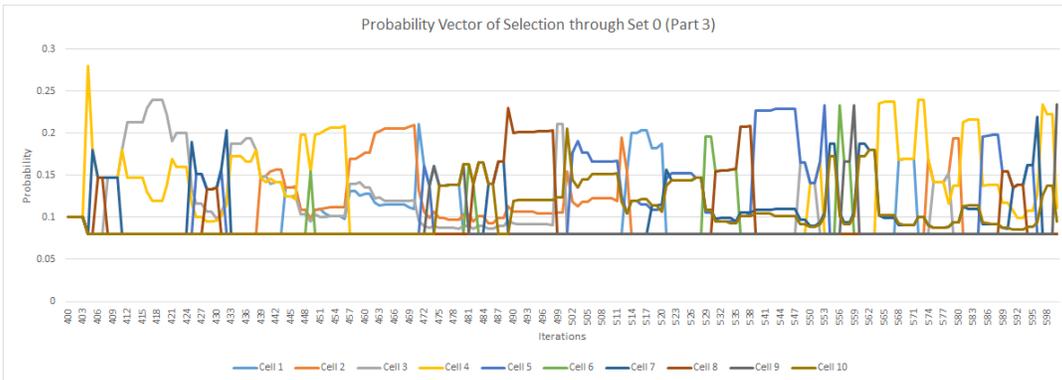
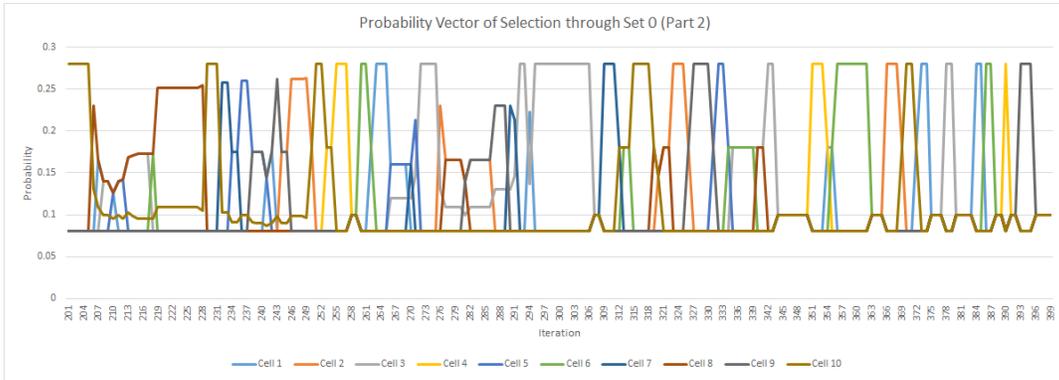


Ilustración 81. Registro de corrección de errores

Esta es la última corrida de 2 que se llevaron a cabo. Esta fue de 1461 iteraciones y la otra de 1562, la primera contiene errores muy graves (No aparece).

Llevé a cabo una corrida simultáneamente con el algoritmo siguiente, el de selección de nuevos ítems (Algoritmo 3), para poder comparar sus características visualmente con este algoritmo (Algoritmo 2), que se logrará apreciar en la siguiente sección. La corrida se ejecutó con el Set 0.





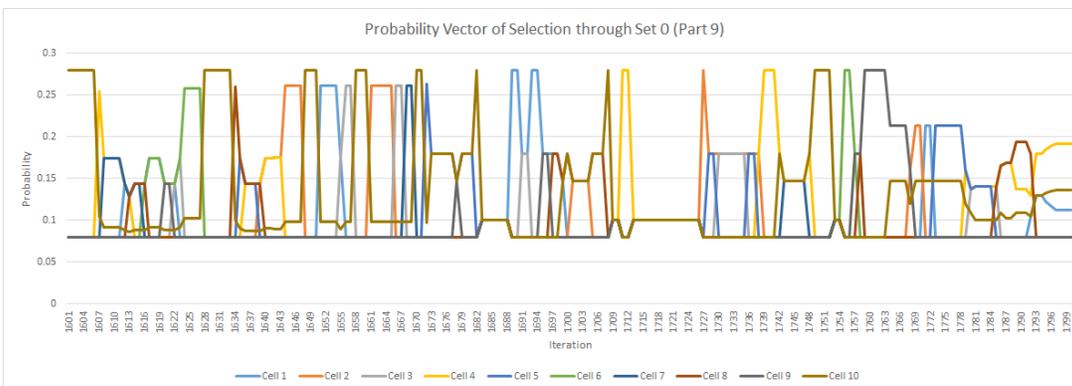
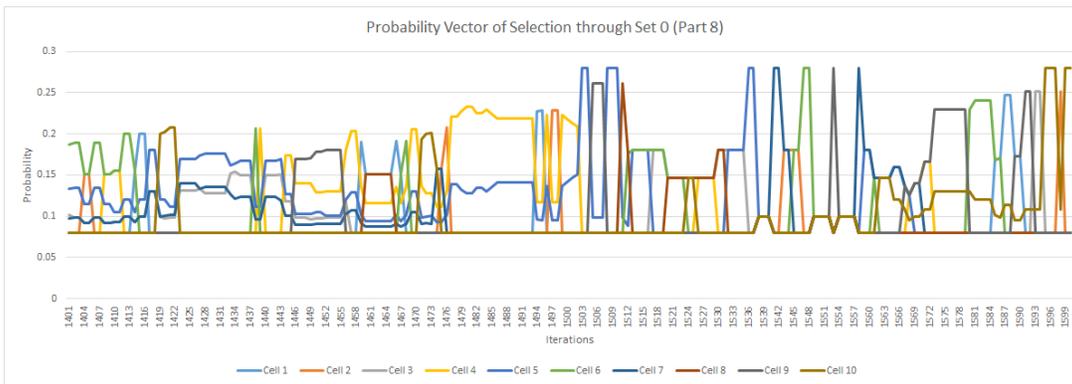
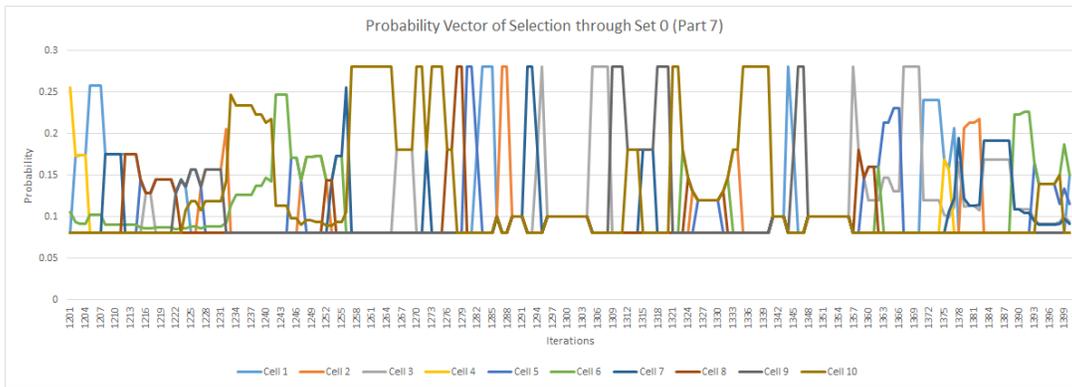
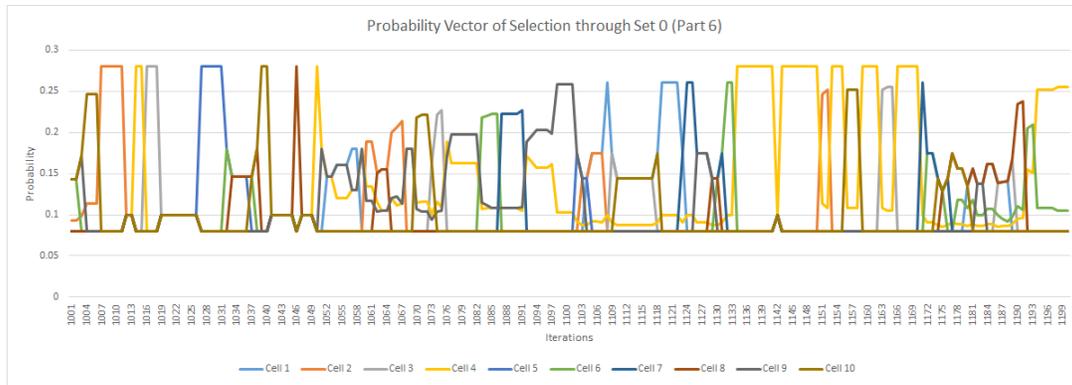


Ilustración 82. Corrida 2, re-implementación del algoritmo

4.4 Algoritmo 3: Selector de nuevos ítems.

Propuesta

El algoritmo consiste en seleccionar una palabra nueva dependiendo de la proporción de palabras pendientes.

Los pasos del algoritmo son:

Recibe 2 variables enteras, Tonalli que es un número aleatorio y TlapowalMekatl que es el número que identifica al grupo seleccionado.

```
Pehpena(int Tonalli, int TlapowalMekatl)
```

Cuenta los elementos del grupo.

```
SELECT COUNT(Pehpena)
FROM [Tlahtolli].[dbo].[Sintli]
WHERE Pantli = " + TlapowalMekatl
```

Cuenta los elementos que ya hayan pasado 'yopanok' del grupo seleccionado.

```
SELECT COUNT(Pehpena)
FROM [Tlahtolli].[dbo].[Sintli]
WHERE Pehpena = 'yopanok' AND Pantli = " + TlapowalMekatl
```

Se calcula:

Proporción de seleccionados = número de seleccionados / número total del grupo

Por último, se compara con el valor aleatorio y si es mayor se seleccionará un ítem nuevo, en caso contrario se seleccionará un ítem de los ya seleccionados.

Para este paso se generó otro log.

Tlahtolme1.txt

```
using (StreamWriter Tlaqilo = new StreamWriter(@"C:\Users\Public\
Tlahtolme" + cbTlahtolMekatl.Text + ".txt", true))
{
    Yeyekolli = Yalwan.Tlahtolme[OkachiAhkopa] + (char)9;
    Yeyekolli += Yalwan.Kisa[OkachiAhkopa].ToString() +
(char)9;
    Yeyekolli += Yalwan.Qalli[OkachiAhkopa].ToString() +
(char)9;
    Yeyekolli += tbKisa.Text + (char)9;
    Yeyekolli += tbKalakowa.Text + (char)9;
    Tlaqilo.WriteLine(Yeyekolli);
}
```

El archivo se exporta a Excel y se seleccionaba en la ListBox Origen de archivos: Unicode (UTF-8) y posteriormente establecer los separadores: Tabulación y Espacio y Finalizar en el Asistente para importar texto.

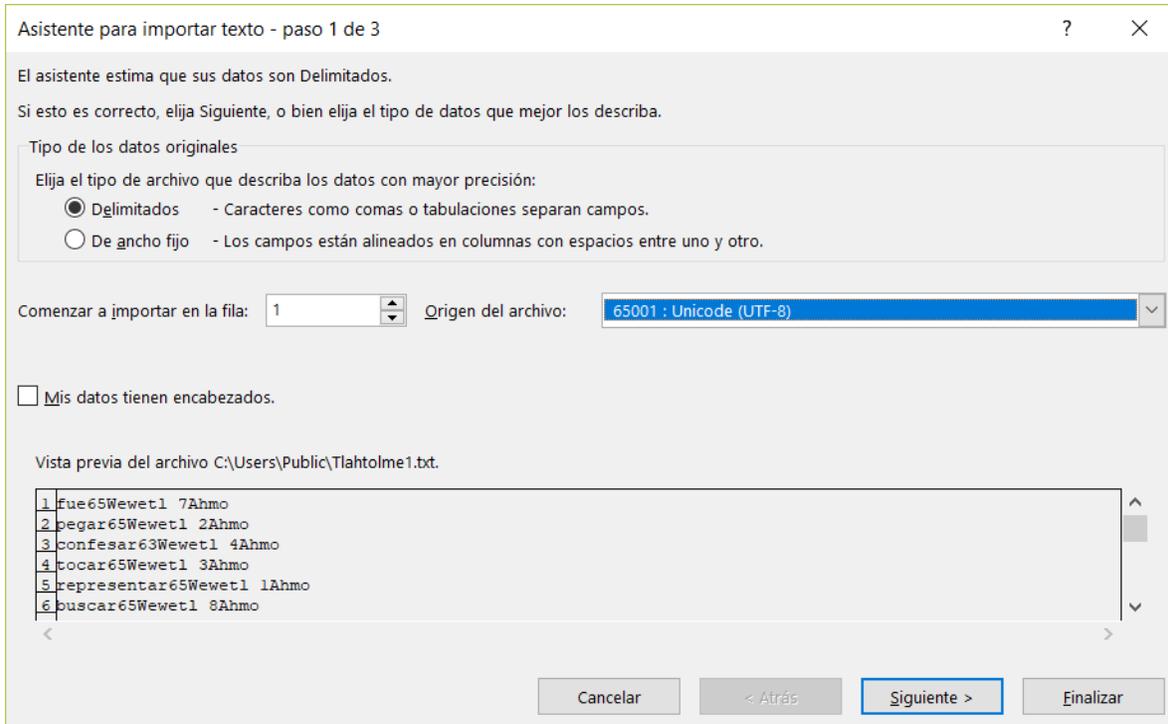


Ilustración 83. Asistente para importar archivos de texto en Microsoft Excel

El archivo contiene 5 filas.

A = Palabra

fue
pegar
confesar
tocar
representar
buscar
sorprender
ganar
hablar
investigar

Ilustración 84. Columna de elementos.

B = Kisa (las veces que salió)

6
6
6
6
6
6
6
6
6
6
6

Ilustración 85. Columna de número de apariciones.

C = Qalli (aciertos)

5
5
3
5
5
5
5
5
3
5
4

Ilustración 86. Columna de número de aciertos.

D = Condición de salida

Wewetl: El elemento permaneció un número de veces mayor al definido.

Ahkopa: El elemento alcanzó la probabilidad 1 y es el elemento de mayor antigüedad.

Wewetl

Ilustración 87. Columna de razón por la que fue expulsada la palabra

E = Posición en el arreglo

7
2
4
3
1
8
6
2
9
0

Ilustración 88. Columna de la posición del vector que es sustituida

F = Selección de elemento

Ahmo: No Seleccionado

Yopanok: Seleccionado

Ahmo
Yopanok
Ahmo

Ilustración 89. Columna de estado del elemento de salida

Para poder graficar el archivo se tiene que hacer un tratamiento de la información.

Primero agregamos una columna con la suma acumulada de Kisa y convertimos a texto el resultado.

El contenido de la celda A2 = A1 + C2 y se replica en las demás celdas.

	A	B	C	D	E	F	G
1							
2	6	fue	6	5	Wewetl	7	Ahmo
3	12	pegar	6	5	Wewetl	2	Ahmo
4	18	confesar	6	3	Wewetl	4	Ahmo
5	24	tocar	6	5	Wewetl	3	Ahmo
6	30	representar	6	5	Wewetl	1	Ahmo
7	36	buscar	6	5	Wewetl	8	Ahmo
8	42	sorprender	6	5	Wewetl	6	Ahmo
9	48	ganar	6	3	Wewetl	2	Ahmo
10	54	hablar	6	5	Wewetl	9	Yopanok

....

545	2094	venir	3	3	Ahkopa	2	Yopanok
546	2097	vender	3	3	Ahkopa	6	Yopanok
547	2100	llover	3	3	Ahkopa	9	Yopanok
548	2103	robar	3	3	Ahkopa	7	Yopanok
549	2106	pagar	3	3	Ahkopa	0	Yopanok
550	2109	investigar	3	3	Ahkopa	6	Yopanok
551	2112	dormir	3	3	Ahkopa	1	Yopanok
552	2115	representar	3	3	Ahkopa	2	Yopanok
553	2118	confesar	3	3	Ahkopa	9	Yopanok
554	2121	necesitar	3	3	Ahkopa	8	Yopanok
555	2125	admirar	4	4	Ahkopa	3	Yopanok

Ilustración 90. Etiquetado de secuencia de aparición

Después ordenamos los datos de acuerdo a la columna B

	A	B	C	D	E	F	G
1							
2	204	admirar	6	5	Wewetl	8	Ahmo
3	500	admirar	6	5	Wewetl	2	Yopanok
4	728	admirar	3	3	Ahkopa	2	Yopanok
5	883	admirar	3	3	Ahkopa	9	Yopanok
6	999	admirar	3	3	Ahkopa	8	Yopanok
7	1175	admirar	3	3	Ahkopa	7	Yopanok
8	1316	admirar	3	3	Ahkopa	2	Yopanok
9	1461	admirar	3	3	Ahkopa	9	Yopanok
10	1602	admirar	3	3	Ahkopa	9	Yopanok
11	1745	admirar	4	4	Ahkopa	6	Yopanok
12	1917	admirar	3	3	Ahkopa	7	Yopanok
13	2125	admirar	4	4	Ahkopa	3	Yopanok
14							
15	234	apoyar	6	6	Wewetl	0	Ahmo
16	464	apoyar	6	6	Wewetl	7	Yopanok
17	740	apoyar	3	3	Ahkopa	3	Yopanok
18	886	apoyar	3	3	Ahkopa	5	Yopanok
19	1011	apoyar	3	3	Ahkopa	4	Yopanok
20	1153	apoyar	3	3	Ahkopa	8	Yopanok
21	1313	apoyar	3	3	Ahkopa	3	Yopanok

...

	A	B	C	D	E	F	G
540	1092	vestir	3	3	Ahkopa	6	Yopanok
541	1159	vestir	3	3	Ahkopa	1	Yopanok
542	1304	vestir	3	3	Ahkopa	1	Yopanok
543	1467	vestir	3	3	Ahkopa	3	Yopanok
544	1596	vestir	3	3	Ahkopa	5	Yopanok
545	1738	vestir	3	3	Ahkopa	0	Yopanok
546	1893	vestir	3	3	Ahkopa	2	Yopanok
547	2046	vestir	3	3	Ahkopa	9	Yopanok
548							
549	114	volar	6	5	Wewetl	6	Yopanok
550	332	volar	6	6	Wewetl	8	Yopanok
551	674	volar	3	3	Ahkopa	1	Yopanok
552	844	volar	3	3	Ahkopa	6	Yopanok
553	960	volar	3	3	Ahkopa	6	Yopanok
554	1117	volar	3	3	Ahkopa	1	Yopanok
555	1258	volar	3	3	Ahkopa	2	Yopanok
556	1446	volar	3	3	Ahkopa	0	Yopanok
557	1554	volar	3	3	Ahkopa	3	Yopanok
558	1723	volar	3	3	Ahkopa	3	Yopanok
559	1881	volar	3	3	Ahkopa	4	Yopanok
560	2058	volar	3	3	Ahkopa	3	Yopanok

Ilustración 91. Log reordenado

Agregamos otra columna.

Donde $H2 = H1 + C2$ y repetimos la operación con cada grupo.

	A	B	C	D	E	F	G	H
1								
2	204	admirar	6	5	Wewetl	8	Ahmo	6
3	500	admirar	6	5	Wewetl	2	Yopanok	12
4	728	admirar	3	3	Ahkopa	2	Yopanok	15
5	883	admirar	3	3	Ahkopa	9	Yopanok	18
6	999	admirar	3	3	Ahkopa	8	Yopanok	21
7	1175	admirar	3	3	Ahkopa	7	Yopanok	24
8	1316	admirar	3	3	Ahkopa	2	Yopanok	27
9	1461	admirar	3	3	Ahkopa	9	Yopanok	30
10	1602	admirar	3	3	Ahkopa	9	Yopanok	33
11	1745	admirar	4	4	Ahkopa	6	Yopanok	37
12	1917	admirar	3	3	Ahkopa	7	Yopanok	40
13	2125	admirar	4	4	Ahkopa	3	Yopanok	44
14								
15	234	apoyar	6	6	Wewetl	0	Ahmo	6
16	464	apoyar	6	6	Wewetl	7	Yopanok	12
17	740	apoyar	3	3	Ahkopa	3	Yopanok	15
18	886	apoyar	3	3	Ahkopa	5	Yopanok	18
19	1011	apoyar	3	3	Ahkopa	4	Yopanok	21
20	1153	apoyar	3	3	Ahkopa	8	Yopanok	24

Ilustración 92. Etiquetado por elemento

Seleccionamos insertar gráfica de dispersión X, Y

Generamos una serie por cada palabra de la columna B y asignamos a cada serie sus correspondientes valores para X en la columna A y para Y en la columna H. Obtenemos las siguientes gráficas.

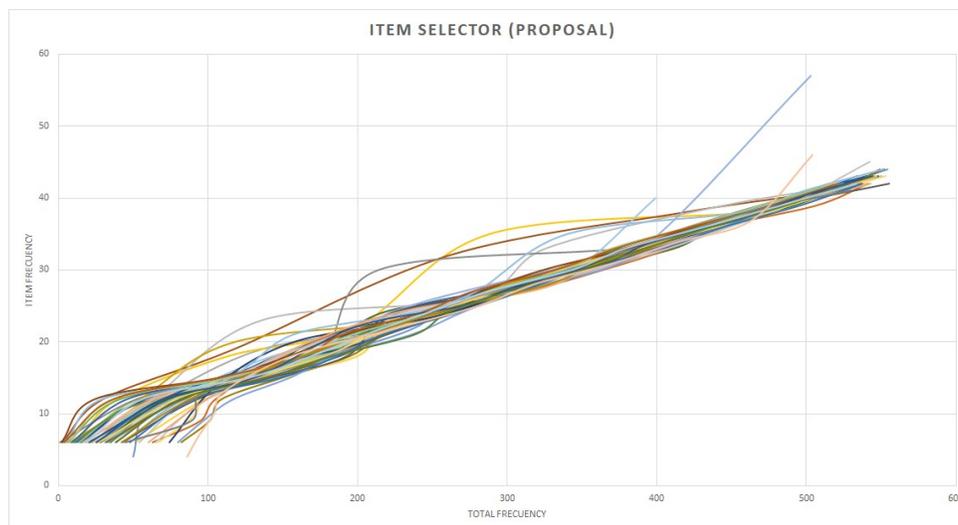


Ilustración 93. Gráfica del log Tlahtolme

En la siguiente imagen se muestra la corrida del segundo set. Lo que quería demostrar era que el comportamiento era consistente. Sin embargo, aproveché para redimensionar el gráfico. Aunque el algoritmo se ejecuta correctamente, no es el más adecuado, ya que como se puede observar en la gráfica todos los elementos aparecen en las primeras 500 iteraciones. El objetivo es postergar en una iteración grande la entrega de todos los elementos con el fin de aumentar el número de iteraciones y por ende las frecuencias por elemento que se necesitan para acabar el juego.

Coloqué por debajo de la gráfica de las frecuencias totales para agregar nuevos ítems (Algoritmo 3) la gráfica del log del algoritmo para modificar las probabilidades instantáneas (Algoritmo 1) para poder ver que ambos algoritmos están midiendo diferentes dimensiones del problema y cómo su respuesta es muy diferente.

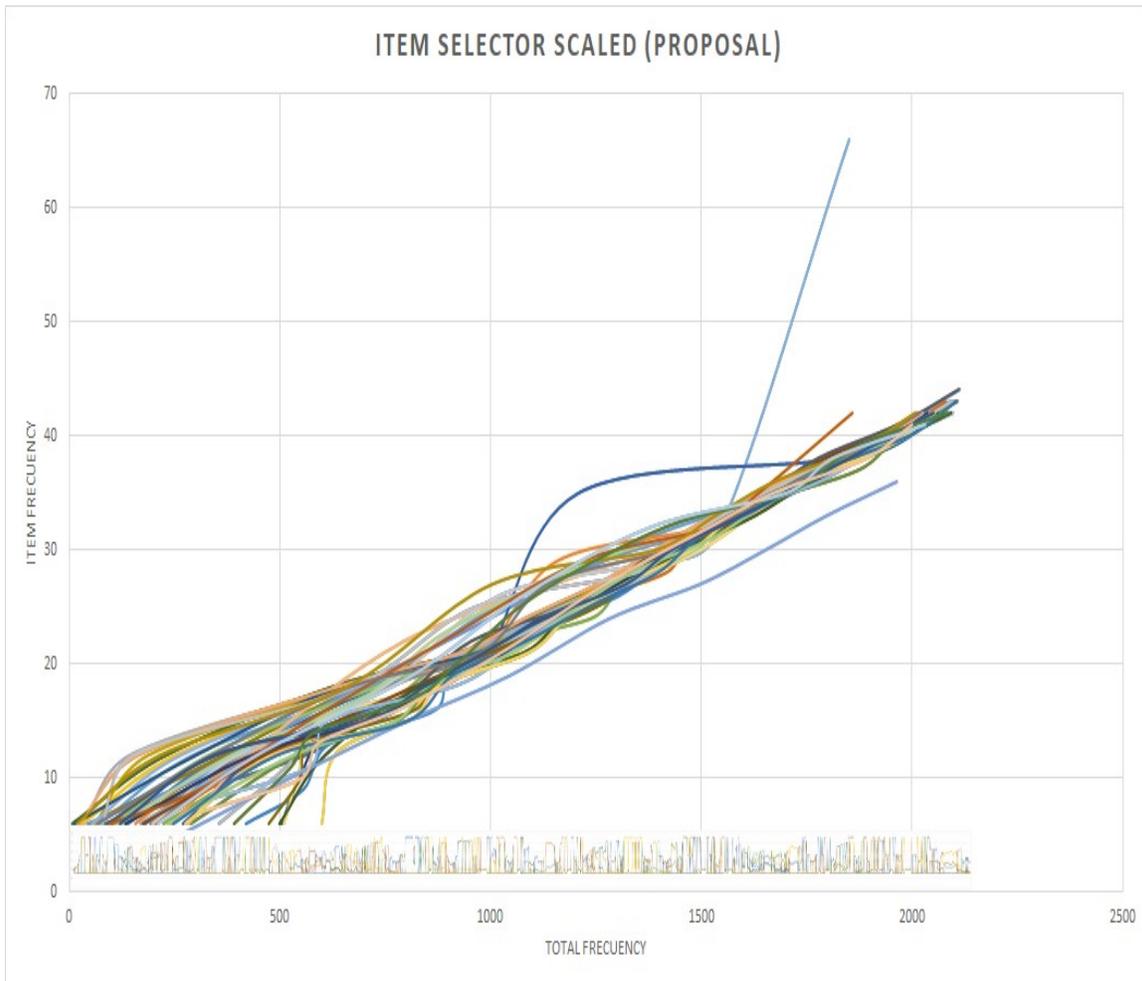


Ilustración 94. Comparación de logs

Se definen 2 funciones, Qi0knepan (Tomar nuevo) y Qi0knepan (Tomar seleccionado)

Los pasos del algoritmo son:

El algoritmo recibe al Tlamankisa (objeto que saldrá), Tlapowalkisa (ubicación en la tabla Tlanextli), Tlapowalmekatl (grupo que le corresponde) y Sehpa (Vector que le corresponde)

Qi0knepan(string TlamanKisa, int Tlapowalkisa, int TlapowalMekatl, int Sehpa)
Seleccionar los campos Kisa (apariciones) y Qalli (aciertos) con los valores de la tabla del grupo correspondiente del elemento que sale.

```
SELECT [Kisa],[Qalli]
FROM [dbo].[Tlanextli" + Sehpa + "]
WHERE [Tlamantli]= '" + TlamanKisa + "'"
```

Seleccionar los campos Kisa(apariciones), Qalli(aciertos), KeskiPehpena(Cuántas veces seleccionado) de la tabla Sintli(Master) del elemento que sale.

```
SELECT [Kisa],[Qalli],[KeskiPehpena]
FROM [Tlahtolli].[dbo].[Sintli]
WHERE [Tlamantli]= '" + TlamanKisa + "'"
```

Se actualizan los valores de Kisa, Qalli, Pehpena (Seleccionar) y KeskiPehpena del elemento que sale del grupo seleccionado en la tabla Sintli (Maestra).

```
UPDATE [Tlahtolli].[dbo].[Sintli]
SET Kisa=" + (YanqikKisa + WeyakKisa) + ",Qalli =" + (YanqikQalli + weyakQalli) + ",Pehpena = 'yopanok',KeskiPehpena = " + KeskiPehpena + "
WHERE Tlamantli = '" + TlamanKisa + "' AND Pantli = " + TlapowalMekatl
```

Selecciona una tupla de la tabla Sintli que tiene el estatus 'ahmo' (no ha sido seleccionado) y pertenece al grupo seleccionado.

```
SELECT TOP (1) *
FROM [Tlahtolli].[dbo].[Sintli]
WHERE Pehpena = 'ahmo' AND Pantli = " + TlapowalMekatl
```

Marca con la etiqueta 'kema' (seleccionado) en la tabla Sintli al elemento que entra

```
UPDATE [Tlahtolli].[dbo].[Sintli]
SET Pehpena = 'kema'
WHERE Tlamantli = '' + YanqikTlamantli + '' AND Pantli = '' + TlapowalMekatl
```

Ingresa el nuevo elemento en la tabla Tlanextli correspondiente con valores igual a 0 en Kisa, Qalli y Pantli (Vector de probabilidades de selección) de la tabla Tlanextli.

```
UPDATE [Tlahtolli].[dbo].[Tlanextli" + Sehpa + "]
SET Tlamantli = '' + YanqikTlamantli + ',Kisa = 0, Qalli = 0, Pantli = 0
WHERE Tlamantli = '' + TlamanKisa + ''
```

Algoritmo de selección de ítem ya seleccionado

Los pasos del algoritmo son:

El algoritmo recibe al Tlamankisa (objeto que saldrá), Tlapowalkisa (ubicación en la tabla Tlanextli), Tlapowalmekatl (grupo que le corresponde) y Sehpa (Vector que le corresponde)

```
QiQitlapan(string TlamanKisa, int TlapowalKisa, int TlapolMekatl, int Sehpa)
```

Las 3 primeras instrucciones son idénticas a las que se encuentran en el algoritmo anterior QiOknepa.

Continúa:

Selecciona los campos TlahtolID (ID de palabra), Tlamantli(Item) del grupo seleccionado de la tabla Sintli(Maestra) donde el estatus es 'yopanok'(Ya ha sido seleccionado) ordenado por el contenido de Kisa(apariciones). Esta pequeña instrucción es la base para uno de los supuestos más importantes de la aplicación que es la equiprobabilidad de cada uno de los elementos de la tabla Sintli.

```
SELECT TOP (1) [TlahtolID],[Tlamantli]
FROM [Tlahtolli].[dbo].[Sintli]
WHERE Pehpena = 'yopanok' AND Pantli = '' + TlapolMekatl + '' ORDER BY Kisa"
```

Y las últimas 3 instrucciones son similares a las del algoritmo anterior.

El algoritmo consiste en ejecutar uno de los algoritmos de selección de Ítems nuevo y ya seleccionado por medio de una función modulo. La variable noTlaka.Keski es un contador de cada elemento desde que comienza el juego.

```
if (noTlaka.Keski % 45 == 0)
    Kalakihtok = true;
```

El inconveniente con este algoritmo es que genera un efecto colateral que es evidente. Cuando un ítem nuevo es seleccionado. Esto se debe a que el nuevo ítem es introducido con valor 0 en el campo Kisa(apariciones) y el algoritmo que se encarga de equilibrar las apariciones de todos los elementos selecciona el menor de todos. Este conjunto de condiciones hace que el nuevo ítem sea seleccionado hasta que haya alcanzado un número mayor que el número menor de los valores del campo.

	TlahtolID	Tlamantli	Pantli	Kisa	Qalli	Pehpena	KeskiPehpena
194	211	viajar	3	51	50	kema	16
195	212	usar	3	63	27	yopanok	8
196	213	restar	3	51	50	kema	15
197	214	servir	3	36	35	kema	11
198	215	mentir	3	60	18	yopanok	4
199	216	enviar	3	74	28	yopanok	8
200	217	pasar	3	24	24	kema	8
201	218	apurar	3	56	12	yopanok	3
202	219	pedir	3	21	20	kema	6
203	220	jugar	3	16	15	kema	4
204	221	copiar	3	17	16	kema	4
205	222	gritar	3	0	0	ahmo	0
206	223	caminar	3	0	0	ahmo	0
207	224	educar	3	0	0	ahmo	0
208	225	subir	3	0	0	ahmo	0
209	226	cancelar	3	0	0	ahmo	0

Ilustración 95. Tabla de DB Sintli

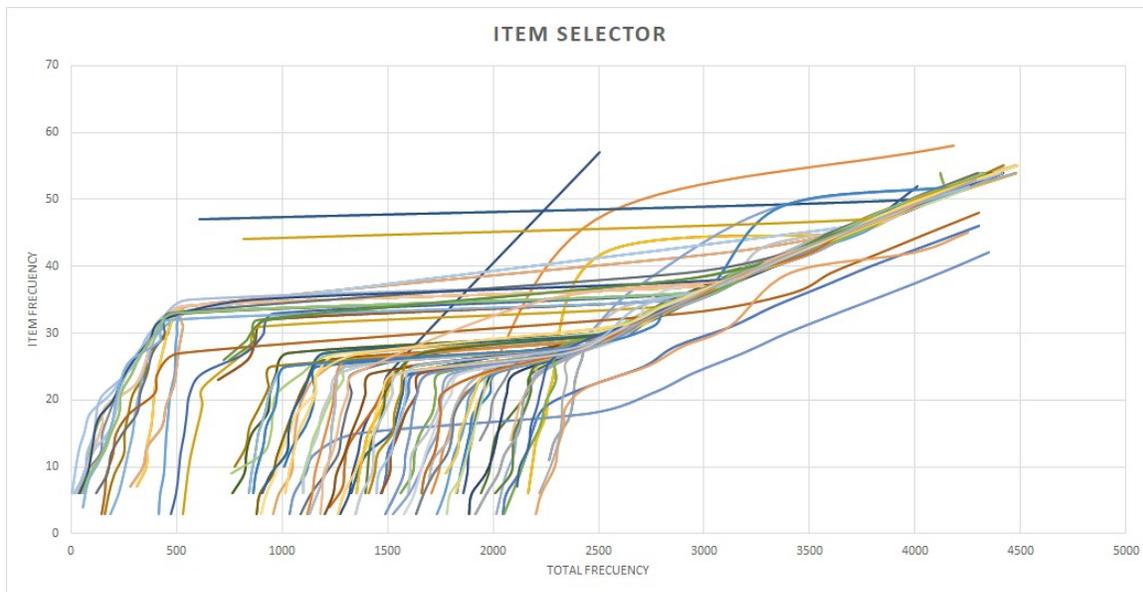


Ilustración 96. Log posterior a la implementación del dosificador

Este es el set 0, el de batalla, que usé básicamente para cualquier tipo de prueba. Usé este set porque estaba consiente que algo no iba a salir bien, lo único que quería era ver el comportamiento. Las inconsistencias que se logran observar fueron causadas, ya que por error llamé a otro prototipo durante la ejecución de la prueba y como tienen en común la base de datos generaron modificaciones en el vector de probabilidades que causaron tales anomalías.

Para corregir tal desfase resultado de la dosificación, (operación modulo) procedí a agregar una variable que introdujera al nuevo ítem con un número que estuviera dentro del rango de los valores del campo Kisa, mi propuesta fue agregarle la media al campo Kisa.

El primer paso fue obtener el número de celdas con valores en el campo Kisa.

```
SELECT COUNT(Kisa)
FROM [Tlahtolli].[dbo].[Sintli]
WHERE Pantli = " + TlapowalMekatl + " AND Kisa !=0
```

Solo si el ítem no ha sido seleccionado.

```
if (Keski != 0)
{
SELECT AVG(Kisa)
FROM [Tlahtolli].[dbo].[Sintli]
WHERE Pantli =" + TlapowalMekatl + "AND Kisa != 0
}
```

Y se almacena en la siguiente instrucción con el nombre Tlanawatl(Relativo a la acción de equilibrar algo).

```
UPDATE [Tlahtolli].[dbo].[Sintli]
SET Pehpena = 'kema', Kisa = "+Tlanawatl+"
WHERE Tlamantli = '"' + YanqikTlamantli + '"' AND Pantli = " + TlapowalMekatl
```

Para tener registro de los cambios creé otra versión de Qi0knepan y creé la función Qi0knepanCe con dichas modificaciones.

A continuación, los resultados obtenidos:

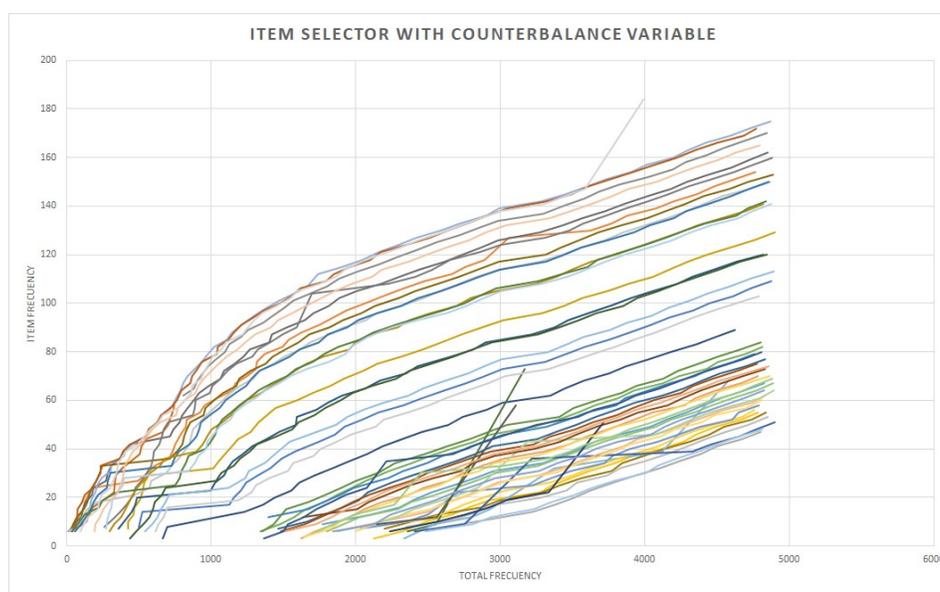


Ilustración 97. Gráfica del log con variable de contrapeso

Este es el set 3 y lo que hice fue introducir una variable de contrapeso. Como se puede ver las curvas son más suaves, pero se genera otro fenómeno subyacente que es la limitación de las probabilidades totales. Esto se puede explicar por la rigidez del algoritmo de definición de probabilidades totales, que define estrictamente la probabilidad de los ítems. Siendo que un ítem es ingresado posteriormente nunca va a alcanzar las probabilidades de los ítems anteriores, porque es equiprobable.

La solución por la cual opté fue mantener el algoritmo de definición de probabilidades totales y reacomodar la curva mediante la eliminación gradual de la variable de contrapeso.

En la propuesta anterior la variable de contrapeso la almacenaba en la misma columna Kisa, para poder afectarla tuve que agregar otro campo Tlanawatilli(algo relativo al concepto de equilibrar algo) a la tabla Sintli(Master).

Y creé otra función, QiQitlapan0me para mantener registro de las modificaciones que le hice a QiQitlapan.

La modificación consistía en:

Leer el campo Tlanawatilli.

```
SELECT [Kisa],[Qalli],[Tlanawatilli],[KeskiPehpena]
FROM [Tlahtolli].[dbo].[Sintli]
WHERE [Tlamantli]= '' + TlamanKisa + '' AND Pantli = '' + TlapolMekatl
```

Si el campo poseía un valor mayor-igual a 2 restarle 2. La condición de que sea igual o mayor que 2 es para asegurar que hay algo que restar. El valor que se resta fue seleccionado, ya que en la mayoría de los casos la selección de un ítem es 3, con lo cual quiero asegurar que se desplace el ítem 1 lugar que en el registro de probabilidades totales lo desplaza varios ítems, siendo altamente probable en general.

```
if (WeyakTlanawatilli >= 2)
    WeyakTlanawatilli-=2;
```

Por último, como separé la variable de contrapeso Tlanawatilli, la tengo que agrupar con el contenido del campo Kisa.

```
SELECT TOP (1) [TlahtolID],[Tlamantli]
FROM [Tlahtolli].[dbo].[Sintli]
WHERE Pehpena = 'yopanok' AND Pantli = '' + TlapolMekatl + ''
ORDER BY Kisa + Tlanawatilli"
```

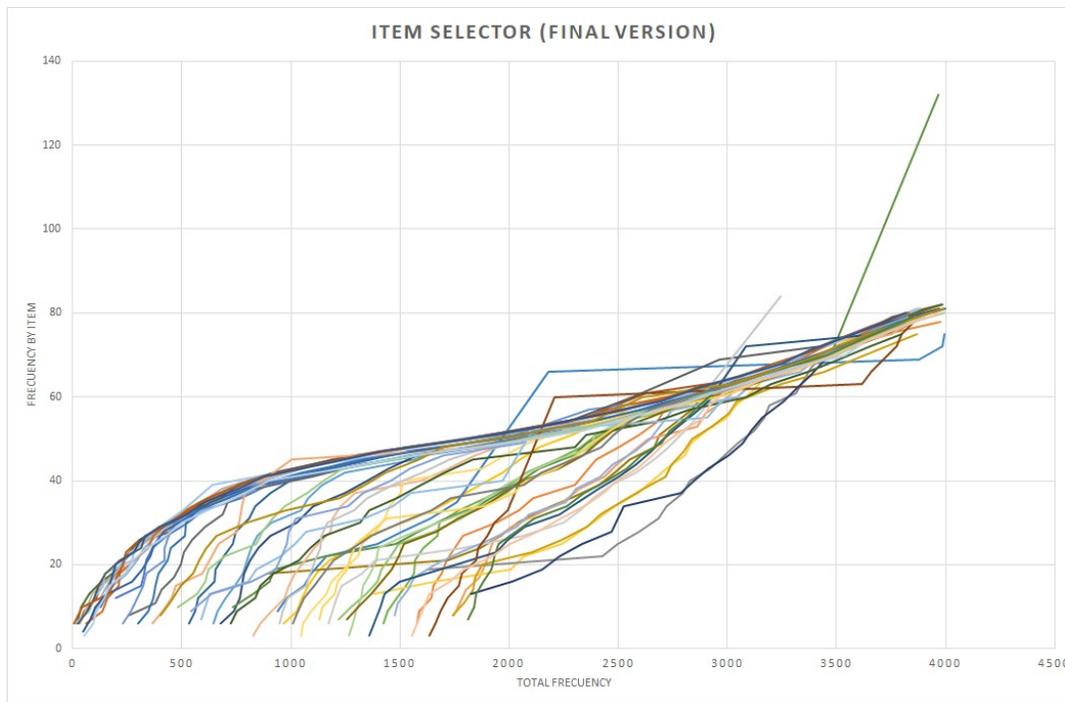


Ilustración 98. Gráfica del log con la variable de contrapeso neutralizada

Aquí se puede ver como la modificación resolvió la no convergencia de cada una de las frecuencias, también cabe señalar la suavidad con que lo hace. También hay que tener en cuenta que la gráfica es la foto de como quedo el proceso, si se hiciera una gráfica del comportamiento de Tlanawatilli a través del tiempo sería como un renacuajo que se aplana hasta desaparecer.

4.5 Algoritmo 4: Modificación de frecuencias con base a la probabilidad de cada ítem.

Para comenzar se tuvo que rediseñar la base de datos nuevamente.

	TiahtolID	Tlamantli	Pantli	Qalli	Kisa	Tlanawatilli	Sesehpa	OmTlanawatilli	Pehpena	KeskiPehpena
1	1	dar	4	117	118	0	3	78	yopanok	38
2	2	dudar	4	119	120	0	3	78	yopanok	38
3	3	acostumbrar	4	118	119	0	3	78	yopanok	38
4	4	inventar	4	187	194	0	1	0	yopanok	55
5	5	romper	4	119	120	0	3	80	yopanok	38
6	6	confundir	4	129	131	0	3	98	yopanok	37
7	7	aumentar	4	136	138	0	2	56	yopanok	43
8	8	trabajar	4	119	119	0	3	78	yopanok	39
9	9	construir	4	119	120	0	3	78	yopanok	38
10	10	conquistar	4	136	138	0	2	57	yopanok	43
11	11	verbos	4	118	118	2	3	78	yopanok	39
12	12	despertar	4	118	119	1	3	78	yopanok	37
13	13	aguantar	4	132	135	2	2	57	yopanok	36
14	14	corregir	4	189	192	1	1	0	kema	61
15	15	componer	4	115	117	2	3	78	yopanok	36

Ilustración 99. Rediseño de la tabla Sintli

Se puede ver el campo Tlanawatilli que fue la solución del problema anterior. Ahora introducimos Sesehpa que es el factor de modificación y OmTlanawatilli que al igual que la otra variable es una variable de contrapeso.

En la ejecución 4000 se realiza la ejecución del algoritmo porque las probabilidades son estandarizadas. En resumen, el algoritmo implementado es un diagrama de frecuencias. Es una operación sencilla, el único objetivo era demostrar que era posible afectar las probabilidades.

```
if (noTlaka.Keski == 4000)
```

llamando a la función:

```
WehkaTonalli(int TlapowalMekatl)
```

WehkaTonalli (podría traducirse como la esencia alejada, antigua). Lo primero que hacemos es calcular todas las probabilidades de acierto y esto es muy sencillo como vimos en el primer algoritmo:

$$Pt = Qalli / Kisa$$

Hay que aclarar en este punto que, aunque los algoritmos 1 y 4 utilizan los mismos conceptos los escenarios son muy diferentes, en el 1 tuve que cazar la probabilidad, se realiza con ayuda de la base de datos. En el 4 son probabilidades totales, históricas.

Falta investigar cual es el mejor método de clasificación.

El pseudo-código es el siguiente:

Definimos el rango:

Rango = Max(Pt)-Min(Pt)

Intervalo = Rango / 3

Definimos intervalos.

Intervalo1 = Min(Pt) + Intervalo

Intervalo2 = Min(Pt) + Intervalo

Intervalo3 = Min(Pt) + Intervalo

Asignamos el índice del Intervalo en el campo Sesehpa que será el factor de modificación

Fin del algoritmo.

Generamos otra versión de

```
QiQitlapan0me(string TlamanKisa, int Tlapowalkisa, int TlapolMekatl, int  
Sehpa)
```

El algoritmo de selección de ítems que ya han sido seleccionados.

Lo único que hace es calcular la variable de contrapeso OmTlanawatilli a las categorías 2 y 3, generando un desfase de frecuencias.

```
if (Sesehpa > 1)  
{  
    Sesehpa--;  
    OmTlanawatilli = WeyakOmTlanawatilli + Tlamanalkisa * Sesehpa;  
}
```

Y finalmente tomamos este valor al momento de ordenar.

```
SELECT TOP (1) [TlahtolID],[Tlamantli]  
FROM [Tlahtolli].[dbo].[OkseSintli]  
WHERE Pehpena = 'yopanok' AND Pantli = " + TlapolMekatl + "  
ORDER BY Kisa + Tlanawatilli + OmTlanawatilli
```

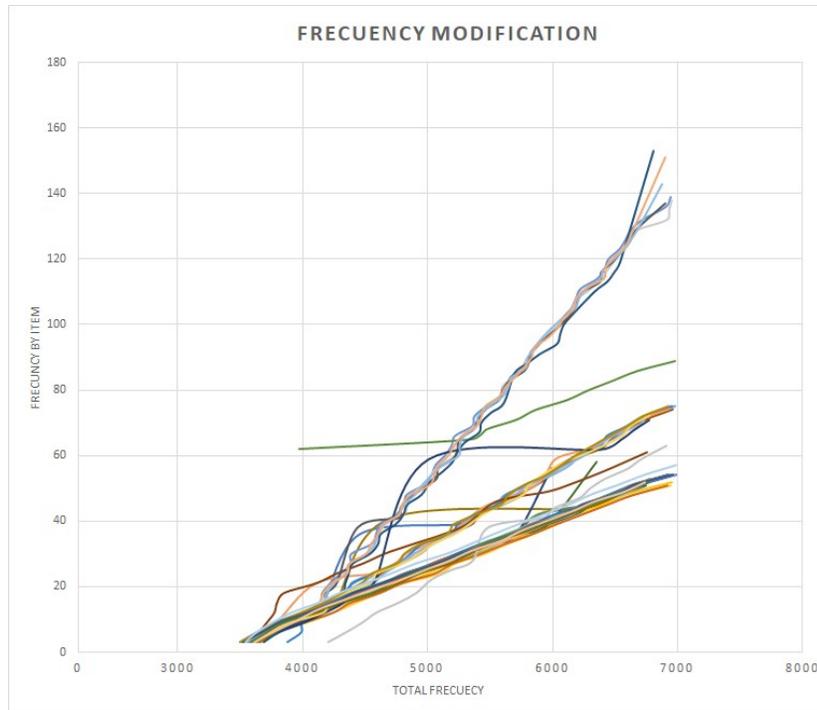


Ilustración 100. Gráfica del log al aplicar el algoritmo de modificación de frecuencias

A partir de la ejecución del 4000 se aplica el algoritmo para modificar las probabilidades de aparición.

En la siguiente gráfica se muestra los elementos correspondientes a cada clase. Cabe recordar que la función para definir los intervalos implementada en la aplicación consiste en definir el rango y posteriormente dividirlo en 3 segmentos iguales. Es posible que exista una mejor distribución, sin embargo, es imperativo tener mayor información sobre los usuarios para tener en cuenta las condiciones y proponer una estrategia adecuada.

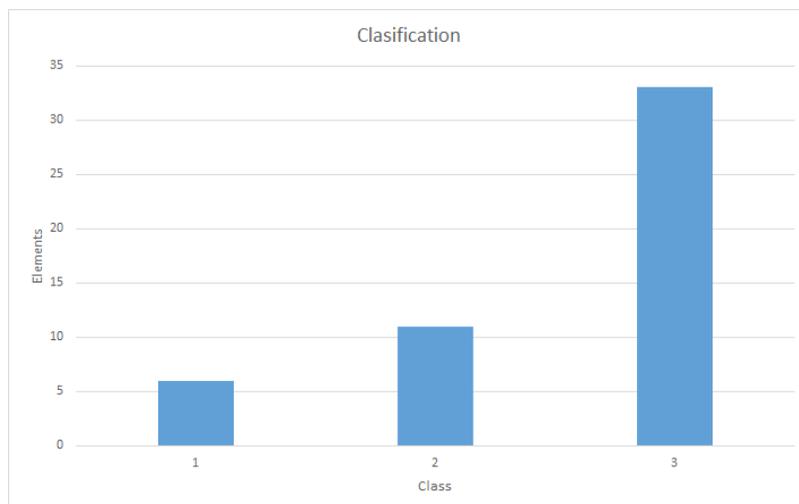


Ilustración 101. Número de elementos distribuidos en 3 categorías

Simultáneamente se obtuvo la gráfica de probabilidades que se muestra a continuación.

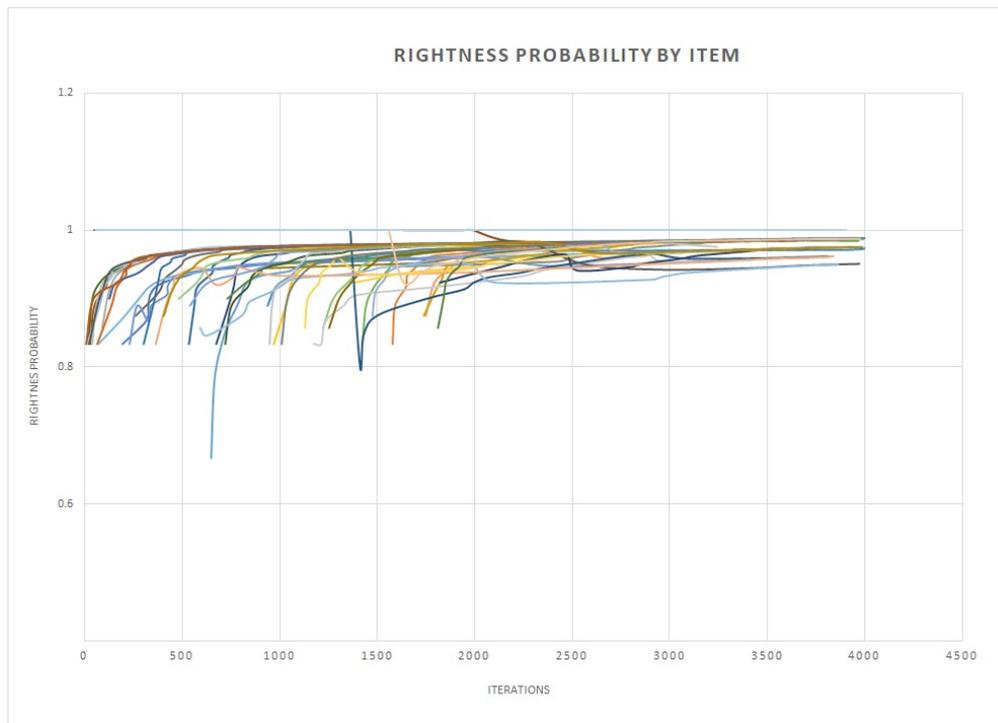


Ilustración 102. Log de Tlahtolme con probabilidad de acierto sobre los elementos

Decidí no analizar esta información en el presente trabajo hasta tener información estadística que me permita inferir en que dimensiones voy a tratar esta información. La tarea está programada para posterior investigación.

Otra cuestión importante que queda pendiente es saber si es posible tener las probabilidades bajo control empalmando el algoritmo 4 con los resultados del Algoritmo 3. Ya que las modificaciones del Algoritmo 4, hasta el momento, son irreversibles.

Conclusiones

En la introducción se planteó controlar las frecuencias de cada uno de los elementos de un grupo grande, puede permitir generar secuencias de suministro con propiedades deseadas. Hasta este punto se puede enunciar que se cumplió el objetivo, los algoritmos son el pegamento entre el vector de selección y la totalidad de elementos, DB, esto permite tener el control de ciertos parámetros. La estructura nos permite controlar propiedades tales como, velocidad de ingreso de elementos, suavidad de suministro, índice de repetición y definición de probabilidades de selección proporcionales.

Cada una de las propiedades obedece a satisfacer una necesidad.

La velocidad de ingreso de elementos nos permite definir en qué momento deseamos introducir un elemento nuevo. Esta propiedad es un simple condicional (está implementado con el operador modulo) que es posible gracias a la estructura que pone los elementos a competir.

La suavidad de suministro, ésta propiedad consiste en el flujo gradual de repeticiones de los elementos, sin cargarse a un elemento excesivamente y/o arbitrariamente.

Índice de repetición, consiste en que tan consecutivas son las repeticiones de un elemento en particular. Si bien no la defino yo, indirectamente se puede controlar, variando condiciones y características del vector. Habrá que desarrollar un modelo que nos permita determinar los parámetros que sí puedo alterar. Sin embargo, se presenta dos dificultades; la diferencia de magnitudes y el comportamiento dinámico. Con respecto a las magnitudes, es posible que no se pueda determinar un comportamiento estable, ya que las secuencias son de orden superior y con respecto al comportamiento dinámico, ignoro si existe un patrón reconocible en el comportamiento de los usuarios y que por obvias razones se modifica al interactuar.

Por último, **la definición de probabilidades**, con esta propiedad se busca modificar las probabilidades de aparición de acuerdo al desempeño del usuario, se le agrega el adjetivo de proporcionales, porque la definición no debe ser abrupta. Al igual que el índice de suministro es posible controlarla indirectamente, pero a diferencia, la modificación es irreversible.

A nivel de experiencia pude percibir que uno no plantea el problema, únicamente los esboza y al implementarlo se va encontrando cosas diferentes a las que esperaba, hasta que los problemas se van develando por sí solos.

La matemática y la programación son 2 herramientas muy poderosas, la matemática, porque te permite “ver” la solución y si no puedes, buscas la manera hasta quedar satisfecho y la programación, principalmente porque al menos para este caso significa la implementación, además, gracias a la programación se pudo sortear ciertos problemas como por ejemplo el algoritmo 3, es una solución completamente programática, matemáticamente sería muy complejo resolverlo y el resultado no llegaría a ser tan bueno como el que se obtuvo programáticamente.

De las principales herramientas usadas en este trabajo, son el álgebra lineal y probabilidad por parte de las matemáticas. Programación procedimental y bases de datos en lo que respecta a la computación. Cabe mencionar que para el trabajo pendiente hay modelos matemáticos que se perfilan para resolver el problema de la búsqueda de un contrincante competente o el orden de aparición de los nuevos elementos.

También creo pertinente señalar que un punto importante en general de este trabajo es el abordar problemas que parecen simples o triviales con una visión matemática-programática. Esto es complicar las cosas, en contra de lo que el principio de Okham dicta; “La pluralidad no se debe postular sin necesidad.” Sin embargo, nos encontramos en un mundo globalizado con acceso a dispositivos muy poderosos, por esta razón creo que vale la pena complicar la solución del problema si esto significa una pequeña mejoría, cuanto más si es en varias dimensiones.

Otro aspecto positivo fue la implementación de la aplicación con la LSM, ya que me permitió vincular algo que me apasiona, y que tiene un impacto positivo en la sociedad, con las matemáticas. Aunque la estructura y los algoritmos pueden ser usados con cualquier contenido, recomendablemente que sean en el mismo dominio, que no sea una asociación muy compleja y por último que el número de elementos sea mayor a 100.

Al respecto del trabajo futuro quedan mucho por hacer, el esqueleto ya está hecho, pero hace falta muchas mejoras e implementación de otros módulos:

- Diseño de infraestructura para análisis de respuesta de usuarios.
- Encuesta.
- Contrastación de la encuesta con datos recabados.
- Interacción entre algoritmos.
- Ajuste de variables.
- Algoritmo de selección de contrincante.

- Algoritmo de mapeo de score.
- Definición de orden de aparición.
- Generar contenido de cultura sorda y distribuirlo a lo largo de todo el juego.

Referencias

- Escalona, M. (2000). Los ordenadores en el proceso de enseñanza-aprendizaje de las ciencias. fundamentos para su utilización, (1994), 1-13.
- Holcomb, L. V. (2017). Estudio de encuesta sobre la traducción y la interpretación en México 2017. La discapacidad en México , datos al 2014. (2014).
- López García, L. A., Rodríguez Cervantes, R. M., Zamora Martínez, M. G., & San Esteban Sosa, S. (2006). *Mis manos que hablan, lengua de señas para sordos*. Cd. México: Trillas.
- McGonigal, J. (2011). *Reality Is Broken: why games make us better and how they can change the world*. Penguin Press.
- Sharma, R. (2017). Computer Assisted Learning – A Study, 4(2), 102-105.
- Torrente, J., Frossard, F., Padmore, M., Mellini, B., Delli, A., & Dumitrache, A. (2009). *Production of Creative Game-Based Learning Scenarios A Handbook for Teachers*. Proactive.
- Whitton, N. (2010). *Learning with Digital Games*. Routledge.

Bibliografía

- Celma, Matilde (2003), Bases de datos relacionales, Madrid, Pearson Educación, S.A.
- MacDonald, Matthew(2012), Pro WPF 4.5 in C#, USA, Apress.
- Olson, Judith, Ways of Knowing in HCI (2014) Springer.
- Sharp, John (2015), Microsoft Visual C# Step by Step, USA, Microsoft Press