



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

PÁGINA WEB DIDÁCTICA APLICADA A LA ENSEÑANZA DE LA
CALIDAD EN LA INGENIERÍA DE SOFTWARE

T E S I S

QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN

P R E S E N T A:

MIGUEL ALEXIS RODRIGUEZ PEÑA
BRANDON ELIU CERVANTES JUAREZ

ASESOR:
ING. JOSÉ ANTONIO AVILA GARCIA

Ciudad Nezahualcóyotl, 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.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

Facultad de Estudios Superiores Aragón

DIRECCIÓN

**BRANDON ELIU CERVANTES JUAREZ
PRESENTE.**

En contestación a la solicitud de registro de fecha 4 de febrero del año en curso, presentada por Miguel Alexis Rodríguez Peña y usted, relativa a la autorización que se les debe conceder para que el profesor, Ing. JOSÉ ANTONIO ÁVILA GARCÍA, pueda dirigirles el trabajo de **TESIS** intitulado "**PÁGINA WEB DIDÁCTICA APLICADA A LA ENSEÑANZA DE LA CALIDAD EN INGENIERÍA DE SOFTWARE**", con fundamento en el punto 6 y siguientes, del Reglamento para Exámenes Profesionales en esta Facultad, y toda vez que la documentación presentada por usted reúne los requisitos que establece el precitado Reglamento; me permito comunicarle que ha sido aprobada su solicitud.

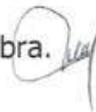
Aprovecho la ocasión para reiterarle mi distinguida consideración.

Atentamente
"POR MI RAZA HABLARÁ EL ESPÍRITU"
Nezahualcóyotl, Estado de México a 21 de febrero de 2020.
EL DIRECTOR


M. en I. FERNANDO MACEDO CHAGOLLA



 C p Secretaría Académica.
C p Jefatura de la Carrera de Ingeniería en Computación.
C p Asesor de Tesis.

FMC/JGPO/bra. 



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

Facultad de Estudios Superiores Aragón

DIRECCIÓN

**MIGUEL ALEXIS RODRIGUEZ PEÑA
PRESENTE.**

En contestación a la solicitud de registro de fecha 4 de febrero del año en curso, presentada por Brandon Eliu Cervantes Juarez y usted, relativa a la autorización que se les debe conceder para que el profesor, Ing. JOSÉ ANTONIO ÁVILA GARCÍA, pueda dirigirles el trabajo de **TESIS** intitulado "**PÁGINA WEB DIDÁCTICA APLICADA A LA ENSEÑANZA DE LA CALIDAD EN INGENIERÍA DE SOFTWARE**", con fundamento en el punto 6 y siguientes, del Reglamento para Exámenes Profesionales en esta Facultad, y toda vez que la documentación presentada por usted reúne los requisitos que establece el precitado Reglamento; me permito comunicarle que ha sido aprobada su solicitud.

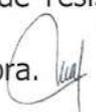
Aprovecho la ocasión para reiterarle mi distinguida consideración.

Atentamente
"POR MI RAZA HABLARÁ EL ESPÍRITU"
Nezahualcóyotl, Estado de México a 21 de febrero de 2020.
EL DIRECTOR


M. en I. FERNANDO MACEDO CHAGOLLA



 C p Secretaría Académica.
C p Jefatura de la Carrera de Ingeniería en Computación.
C p Asesor de Tesis.

FMC/JGPO/bra. 

Dedicatorias

Agradecimientos de: Miguel Alexis Rodríguez Peña.

A mis padres por haberme brindado su apoyo y consejos que me han ido guiando durante toda mi vida, sé que no ha sido fácil, pero como pueden ver que valió la pena, Además de estarme aguantado todo este tiempo que si bien sé que no podre regresarles merecidamente todo lo que me han brindado y es por ello que al menos este proyecto me permita devolverles un poco de eso.

A mi hermano por ser amigo y soportarme tal como soy.

A mis abuelos que, aunque algunos de ellos ya se fueron, hubiera deseado que estuvieran conmigo y juntos celebrar, pero donde quiera que estén, estoy seguro de que estarán orgullosos por verme es por eso que dedico esta obra, a mi abuelo Sixto que siempre ha estado conmigo y que se festejara conmigo este momento.

A mis tíos que puedo decir que siempre me apoyaron cuando más los necesite y que siempre me han enseñado a conducirme con honestidad.

A mis amigos, aunque han sido pocos siempre nos apoyamos sin pedir nada a cambio, por la confianza y alegrías que me brindaron en estas mareas que navegamos juntos y los momentos en que sentíamos que el barco se nos hundía.

A mis profesores por la enseñanza que cada uno de ustedes dejo en mí y aunque algunas veces no entendí con el paso del tiempo he ido comprendiendo poco a poco lo que me decían.

A mi profesor y asesor de tesis por el apoyo que nos brindó desde que lo conocimos, así como consejos, orientación, el tiempo en que nos estuvo corrigiendo y nos condujo a culminar de forma exitosa este proyecto.

A la institución, por haberme permitido ser parte de esta gran comunidad estudiantil, así como las amistades que forje dentro de ella durante estos valiosos y grandísimos años de los cuales valió la pena cada segundo.

Y finalmente, quiero agradecerme a mí, por creer en mí, por nunca rendirme, por intentar hacer más cosas bien que mal y por no dejar de intentarlo

Alexis eres un crack

Dedicatorias

De Brandon Eliu Cervantes Juárez.

Mamá, gracias por todo, eres el soporte y la fuerza que nos mantiene a flote a mis hermanos y a mí, siempre dándonos ánimos cuando más los necesitábamos y apoyándonos siempre de la mejor manera.

A mis hermanos, por todas las cosas buenas y malas que hemos vivido juntos y siempre mantenernos unidos.

A mi novia Linda, que me apoyo he hizo esforzarme más de lo que yo me hubiera esforzado, sin duda eres de las mejores cosas que me paso a lo largo de esta etapa universitaria.

A mis amigos, que sin ellos esta aventura universitaria no hubiera sido la que fue, un viaje en barco.

A mis profesores por haberme enseñado y soportado todo este tiempo.

A la escuela e institución por haberme formado.

Índice.

Introducción	6
1.1. Control de calidad en la ingeniería de software	8
1.1.2. Ingeniería	9
1.2. Software	10
1.2.1. Clasificación del software	12
1.2.2. Breve historia del origen del Software	17
1.3. Ingeniería de Software	18
1.3.1. Objetivos de la Ingeniería de software	20
1.3.2. Breve historia de la ingeniería de software	22
1.3.3. Evolución de la Ingeniería de Software	24
1.4. Concepto de Calidad	27
1.4.1. Orígenes de la Calidad	28
1.4.2. Concepto de control	32
1.4.3. Control de Calidad	34
1.5. Calidad en el proceso de desarrollo de Software	37
1.5.1. Casos de errores en el software	40
1.6. Normas y estándares de calidad utilizados para el desarrollo de software.	42
1.6.1. CMMI	41
1.6.2. PSP	44
1.6.3. TSP	45
1.6.4. PMBOK	47
1.7. Metodologías para el control de calidad del software.	57
1.7.1. Metodologías tradicionales	58
1.7.2. Metodologías ágiles	58
2.1 Calidad de un sistema de software.	62
2.1.1 Costo en la calidad de software	63
2.1.2 Sistema de control de calidad en el software	63
2.1.3 Garantía de la calidad del software (SQA)	64
2.1.4 Metodología SQA	66
3.1 ¿Qué son las ISO?	68
3.1.1 Ventajas de las ISO	69

3.1.2 Familias de las normas ISO	69
3.2 ISO 9004-2 : 2008	72
3.2.1. ISO 9000-3 : 2008	73
3.2.2 ISO 12207	74
3.2.3 ISO/IEC 9126	77
4.1. Concepto de “Pagina Web”.	81
4.2. Software utilizado.	83
4.2.1. HTML	83
4.2.2. CSS	84
4.2.3. PHP	86
4.2.4. Visual Studio Code	87
4.2.5. Javascript	88
4.2.5.1. Librería JQuery	89
4.3. Instalación del software	90
4.3.1. Instalación del software en Windows 10	90
4.3.2. Instalación del software en Ubuntu Budgie 18.04 LTS	97
4.4. Desarrollo de la página web en HTML	109
4.4.1. Estructura básica del HTML	109
4.4.2. Creación del archivo index.html.	111
4.4.3. Estructura HTML para el registro de usuario	116
4.5. Desarrollo de la página web en CSS	121
4.5.1. Creación del archivo CSS para index.html	122
4.5.2. Creación del archivo CSS para la página de registro	128
4.5.3. Haciendo el sitio web responsivo (Media queries)	134
4.6 Desarrollando los efectos de la plataforma con JQuery.	136
4.7. Desarrollo de la base de datos en XAMPP con MySQL	140
4.8. Implementación del código PHP en los HTML	149
4.9. Actualización de la página (HTML, CSS y JS)	158
Conclusion	178
Anexo: Código completo del proyecto	180
Referencias	214

Introducción.

Con este trabajo queremos apoyar a aquellos alumnos, compañeros y a cualquiera que quiera sumergirse en este ámbito de la Calidad en la Ingeniería de Software. Ya que por desgracia uno como estudiante que empieza a desarrollarse pasa por alto constantemente o simplemente no le presta la atención que realmente debería tener.

Se explican los conceptos al inicio de la tesis, desde los principios básicos de la Calidad e Ingeniería de Software, para que el lector se familiarice con todo, desde un estudiante hasta alguien que no esté familiarizado con el tema, haciendo de esta una lectura amena en la que los conceptos presentados permitan al lector comprenderlos de forma fácil.

Presentado de forma breve la historia de la Ingeniería de Software y la Calidad, así como los estándares que son más empleados en el desarrollo de software.

Se muestra la forma en la que fue desarrollada la página web, que tiene la intención de ser didáctica e informativa para que la información presentada en este trabajo también pueda ser presentada de manera digital. Explicando que software se utilizó, como fue instalado, como lo empleamos y como se desarrolló cada una de las clases del proyecto, para aquel que quiera realizarlo por cuenta propia no tenga problema alguno.

Comprendemos que, en Ingeniería de Software y Calidad, el uso de la tecnología se ha vuelto algo cotidiano, ya que algunas normas como lo pueden ser las ISO son usadas una vez que los egresados entren al mundo laboral, es por ello que es de suma importancia para la educación de los estudiantes de Ingeniería en Computación dentro de la Facultad de Estudios Superiores Aragón.

Es por ello que buscamos contribuir a la enseñanza y crear una página web simple e intuitiva para que los alumnos tengan a su alcance la información aquí recolectada, apoyando con esto a su educación y desarrollo como estudiantes. Mostrando la importancia que tiene la Calidad en la Ingeniería de Software a través de conceptos, normas y estándares bajo los que se rige la Calidad en la Ingeniería de Software.

Capítulo 1.

Control de calidad en la Ingeniería de Software.

1.1. Control de calidad en la ingeniería de software

Quizá para muchos estudiantes el aspecto de la calidad en el desarrollo de software es algo a lo que a veces no se le da mucha importancia, pasa desapercibido, o por malas prácticas no es tomado en cuenta. Pero, ¡sí que es importante! Por ejemplo: ¿cuántas veces, como usuarios nos hemos encontrado con aplicaciones o páginas web que fallan?, o ¿alguna vez has encontrado un aparato electrónico cuyo software es desastroso o simplemente falla de la noche a la mañana?, ahí tenemos un claro ejemplo de la falta de calidad en el software, que para el usuario puede resultar frustrante, pero a la empresa o compañía le resulta en graves pérdidas económicas, incluso en algunos casos estos fallos de software han costado la vida de personas.

Y esto no es un problema aislado, actualmente la mayoría de software está compenetrado en los bienes y servicios de la sociedad, pueden ser áreas tales como las de la salud, las telecomunicaciones, la seguridad nacional, la economía y muchas más. Todo esto a escala mundial. Resultaría realmente desastroso que fallara por malas prácticas, es por eso que

existen normas y prácticas que se deben seguir, para que un software sea confiable, eficaz y eficiente.

1.1.2. Ingeniería

Debemos empezar por conocer qué es la ingeniería, antes de adentrarnos más en esta investigación, es por ello que mostraremos diferentes conceptos con el fin de conceptualizar posteriormente; Julián Pérez Porto y María Merino a través de la web *definicion.de* (2009) dicen que:

“La ingeniería es el estudio y la aplicación de las distintas ramas de la tecnología. El profesional en este ámbito recibe el nombre de ingeniero. La actividad del ingeniero supone la concreción de una idea en la realidad. Esto quiere decir que, a través de técnicas, diseños y modelos, y con el conocimiento proveniente de las ciencias, la ingeniería puede resolver problemas y satisfacer necesidades humanas.”

La web de *significados.com* (2016) dice lo siguiente:

“Se conoce como ingeniería a la disciplina que se vale de un conjunto de conocimientos de tipo técnico, científico, práctico y empírico para la invención, el diseño, el desarrollo, la construcción, el mantenimiento y la optimización de todo tipo de tecnologías, máquinas, estructuras, sistemas, herramientas, materiales y procesos.”

Según Adrián, Yirda (2019) en la web de *conceptodefinicion.de* mencionan lo siguiente:

“La ingeniería es una profesión en la que los conocimientos científicos y empíricos se aplican para la conversión óptima de los materiales y fuerzas de la naturaleza en usos prácticos para la humanidad, así como, la invención, perfeccionamiento y utilización de la técnica industrial, y a la resolución de problemas técnicos-sociales. Esta disciplina también es considerada como un arte, debido a que la capacidad imaginativa y de creación del ser humano sobresale para concebir cosas que aún no existen, y es por medio de la aplicación de sus conocimientos científicos que transforma esas ideas en acción o en una realidad.”

1.2. Software

Es importante conocer los distintos conceptos que se tiene sobre el software, estos ayudan a comprender mejor qué es el software, o a que hace referencia este término; la *RAE (2018)* dice:

“1. m. Inform. Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora”

La web *significados.com (2019)*, menciona sobre el software lo siguiente:

“Software es un término informático que hace referencia a un programa o conjunto de programas de cómputo, así como datos, procedimientos y pautas que permiten realizar distintas tareas en un sistema informático.

Comúnmente se utiliza este término para referirse de una forma muy genérica a los programas de un dispositivo informático, sin embargo, el software abarca todo aquello que es intangible en un sistema computacional.

Software es un término procedente del idioma inglés, que es aceptado por la RAE y que no posee una traducción que se ajuste al español.”

Según *Somerville*, en el libro *Ingeniería de Software (2005)* dice que:

“... el software no son sólo programas, sino todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta. Por lo general, un sistema de software consiste en diversos programas independientes, archivos de configuración que se utilizan para ejecutar estos programas, un sistema de documentación que describe la estructura del sistema, la documentación para el usuario que explica cómo utilizar el sistema y sitios web que permitan a los usuarios descargar la información de productos recientes.”

Mientras que *Pressman (2010)* menciona que:

“El software de computadora es el producto que construyen los programadores profesionales y al que después le dan mantenimiento durante un largo tiempo. Incluye programas que se ejecutan en una computadora de cualquier tamaño y arquitectura, contenido que se presenta a medida de que se ejecutan los programas de cómputo e información descriptiva tanto en una copia dura como en formatos virtuales que engloban virtualmente a cualesquiera medios electrónicos. (Pág. 1)”

Edgar Serna (2013), en su libro *Libro Blanco de la Ingeniería de Software en América Latina*, define al software de la siguiente manera:

“En términos generales se refiere a la información que se almacena en algún tipo de medio digital. El software es la interfaz y la lógica abstracta que vincula al ser humano con una tecnología tangible y/o otro software, permitiendo una interacción y retroalimentación mutuas. En una visión más técnica, el término se utiliza para describir una colección de programas informáticos, algoritmos, procedimientos y la documentación necesaria que realiza algunas tareas en un sistema, como operar los equipos y dispositivos tecnológicos relacionados, y la gestión de la información para la toma de decisiones.” (Pág. 17)

Mientras que Marco Alfredo Cedano Olvera, Alfredo Cedano Rodríguez, José Antonio Rubio González, Arlem Carolina Vega Gutiérrez, en su libro “Fundamentos de computación para ingenieros” (2014), definen al software de la siguiente manera:

“En computación, el software en sentido estricto es todo programa o aplicación programada para realizar tareas específicas. Es el soporte lógico e inmaterial que permite que la computadora pueda desempeñar tareas inteligentes, conduciendo a los componentes físicos o hardware con instrucciones y datos a través de diferentes tipos de programas. El software es el conjunto de instrucciones que las computadoras utilizan para manipular datos. Sin él, la computadora sería un conjunto de mecanismos sin utilizar. Al cargar los programas en una computadora, la máquina actuará como si recibiera una educación instantánea; de pronto “sabe” cómo pensar y cómo operar.” (Pág. 92)

Por lo que a partir de estas definiciones es posible conceptualizar el software como *aquel programa que contiene instrucciones necesarias para que un sistema de cómputo se ejecute*

correctamente, abarcando también sus datos y documentación. Así mismo se puede decir que es todo lo no físico o intangible dentro de un sistema de cómputo.

1.2.1. Clasificación del software.

A partir del concepto anterior de software se debe comprender que no todo el software es igual y que no todo el software se utiliza para lo mismo. Por ejemplo, no es lo mismo un software que se utiliza para editar texto, que un software de ofimática.

Sobre su clasificación Cedano et al. (2014) mencionan que:

“... según las funciones que realizan pueden ser clasificados en: de sistema, de aplicación o de programación.”

Con lo anterior expuesto podemos afirmar que el software se clasifica en tres tipos, “Software de Sistema”, “Software de aplicación” y “Software de programación”.

1. Software de sistema.

Sobre el Software de Sistema, la definición que da la web de *significados.com* (2019) es la siguiente:

“Es el programa responsable de la ejecución de todas las aplicaciones necesarias para que un sistema opere correctamente. Suele confundirse con el sistema operativo, pero va más allá, ya que también incluye las herramientas de optimización, los controladores de dispositivo y los servidores, entre otros componentes.”

Sobre este software *Cedano et al. (2014)* dice lo siguiente:

“Se llama software de sistema o software de base al conjunto de programas que sirve para interactuar con el sistema, delegando el control sobre el hardware y dándole el soporte a otros programas. El software del sistema se divide en tres tipos: sistema operativo, controladores de dispositivos y utilerías.”

Estela Raffino (2018) en la web de *concepto.de* explica sobre el Software de Sistema:

“El software de sistema se distingue del software de aplicación en que no le ofrece al usuario la posibilidad de cumplir sus tareas, sino que se ocupa de mantener funcionando el sistema informático. Dicho de otro modo, se trata de un sistema operativo: un programa diseñado para hacer funcionar la máquina y mantenerla encendida y lista, dispuesta para la ejecución de los programas de aplicación.”

La web de *cavsi.com* (s.f.) da las siguientes definiciones sobre el software de sistema:

“El software del sistema se encarga de controlar, integrar y administrar los componentes de hardware de un sistema informático, proporcionando un entorno amigable para que los usuarios puedan ejecutar otras aplicaciones de software.

El software del sistema es un conjunto de programas de software que permiten al usuario interactuar con los dispositivos sin perderse en la complejidad técnica del equipo. Estos programas son la base de la arquitectura del software, incluyendo las partes que regulan las funciones de entrada/salida.

El software del sistema incluye los sistemas operativos, controladores de dispositivos, servidores, sistemas de ventanas y programas de utilidades. Permitiendo coordinar todos los dispositivos internos o externos de la computadora como impresora, mouse, teclado, monitor, etc.”

Así mismo la web *TecnoMagazine.com* (2019) menciona sobre el Software de Sistema lo siguiente:

“El software del sistema es, en términos simples, el que nos permite interactuar con un dispositivo por medio de su hardware.

Este software base está compuesto principalmente por el sistema operativo y por los drivers o controladores, y en menor medida también podemos incluir a las librerías dentro de esta categoría.

Otros tipos de software que también pueden ser clasificados como software de sistema son los gestores de arranque, las interfaces de línea de comandos, las interfaces gráficas y el BIOS.”

Conceptualizando lo anterior, podemos afirmar que el Software de Sistema es el encargado de controlar e interactuar con una computadora o cualquier dispositivo actual (tablets, laptops, etc.), dando también el control sobre el hardware adicional que estos manejen (pantallas, discos, impresoras, teclados, puertos, etc.). A su vez también son de gran ayuda para el mantenimiento de otros programas dentro del mismo sistema operativo. El Software de Sistema incluye los siguientes:

- Sistemas operativos (y algunos componentes dentro de él).
- Controladores de dispositivos.
- La BIOS. (Basic Input/Output System / Sistema Básico de Entrada y Salida)
- Herramientas de corrección y optimización.
- Programas utilitarios.
- Servidores.

2. Software de Aplicación

El Software de Aplicación se describe en la web de *significados.com* (2019) de la siguiente manera:

“Un software de aplicación es un programa diseñado para facilitar algunas tareas específicas para cualquier medio informático, ya sea en computadores, tablets o celulares.

Los videojuegos, los programas de diseño asistido (como CAD), el software utilizado en las telecomunicaciones, las aplicaciones de productividad empresarial o educativa son algunos ejemplos de este tipo de programas.”

Cedano et al. (2014) define el Software de Aplicación de la siguiente forma:

“Son los programas diseñados para o por los usuarios para facilitar la ejecución de tareas específicas en la computadora, como pueden ser las aplicaciones ofimáticas (procesador de texto, hoja de cálculo, programa de presentación, sistema de gestión de base de datos), u otros tipos de software especializados como software médico, software educativo, editores de música, programas de contabilidad, entre otros.”

La web *cavsi.com* (s.f.) lo define de la siguiente forma:

“El software de aplicación utiliza las capacidades del computador para realizar una tarea específica. El software de aplicación es capaz de manipular texto, números, gráficos y archivos de audio y video. El software de aplicación se utiliza muchas veces para fines comerciales, encontrados en el campo educativo, de negocios y médico.

El software de aplicación se conoce simplemente como aplicación o «app».”

La web *247tecno.com* (s.f.) define al Software de Aplicación de la siguiente manera:

“Se denomina como software de aplicación o una aplicación programa, a un tipo de software de computadora el cual está diseñado para la realización de un grupo de funciones, actividades o tareas, las cuales son coordinadas por el usuario y se ejecutan para su beneficio. Las aplicaciones normalmente se desarrollan para un sistema operativo en particular ya sea Windows, Mac o Linux entre otros. En muchas ocasiones este término se lo abrevia como app o aplicación.”

Estela Raffino (2018) en la web de *concepto.de* explica sobre el Software de Aplicación:

“...se entiende por software de aplicación, programas de aplicaciones o en algunos casos aplicaciones, al conjunto de los programas informáticos generalmente instalados en el sistema por el usuario, y diseñados para llevar a cabo un objetivo determinado y concreto, de tipo lúdico, instrumental, comunicativo, informativo, etc.

Dicho de otro modo: todos aquellos programas que no tienen que ver con el funcionamiento del computador, sino que instalamos en él para darle funciones determinadas como herramienta de trabajo (hoja de cálculo, procesador de palabras, programas de diseño gráfico, etc.), de ocio (videojuegos, reproductores de audio o video, etc.) o de información (enciclopedias digitales, navegador de Internet, etc.), entre otras.”

De acuerdo a las citas anteriores podemos conceptualizar que el Software de Aplicación es aquel software que está dirigido al usuario, cuya función principal es ayudarlo en ciertas tareas específicas. Como pueden ser los programas de ofimática (Office, Libre Office), estaciones de

trabajo de audio digital (REAPER, Cubase, Ableton Live, Pro Tools, FL Studio), así como también los videojuegos. Algunos otros ejemplos del Software de Aplicación son:

- Software de ofimática.
- Software para trabajo de audio.
- Software de educación o de enseñanza.
- Software para telecomunicaciones (Internet y derivados)
- Software para uso médico
- Software para diseño
- Software creado específicamente para empresas o personas
- Bases de datos
- Videojuegos

3. Software de Programación

El Software de Programación es definido por Cedano et al. (2014) como:

“El software de programación es el conjunto de herramientas que permiten al desarrollador informático formular programas usando diferentes alternativas y lenguajes de programación.

Este tipo de software incluye esencialmente compiladores, intérpretes, ensambladores, enlazadores, depuradores, editores de texto y un entorno de desarrollo compuesto que tiene las herramientas anteriores y normalmente cuenta con una avanzada interfaz gráfica de usuario (GUI).” (Pág. 99 -100)

Estela Raffino (2018) en la web de concepto.de dice sobre el Software de Programación lo siguiente:

“Se conoce así a los lenguajes de programación, es decir, a los programas que sirven para diseñar otros programas. Para ello el usuario o programador cuenta con diversas herramientas de asistencia en la escritura del código de la nueva pieza de software, tales como funciones de prueba o interfaces amigables.

Todos los programas existentes provienen de un software de programación, y cada software de este tipo permite “escribir” un lenguaje o protocolo diferente, aunque muchos, claro, son el equivalente a políglotas.”

La web de *significados.com* (2019) define al Software de Programación como:

“Son todas las herramientas que permiten el desarrollo de nuevo software. Por ello, no suelen ser utilizadas por el usuario final sino por los programadores.”

Con las definiciones anteriormente citadas es posible definir al Software de Programación como el software que se utiliza para crear software. Principalmente dirigido a programadores. Usando lenguajes de programación. Estos incluyen:

- Compiladores
- Intérpretes
- Depuradores
- Enlazadores
- Editores de texto
- IDE [Integrated Development Environment / Entorno de Desarrollo Integrado o Entorno de Desarrollo Interactivo] (Software que incluye compilador, intérprete, depurador, enlazador, editor de texto y una interfaz gráfica de usuario o GUI)

1.2.2. Breve historia del origen del Software.

El software en sus comienzos no era visto como un producto, sino como un añadido del hardware que variaba dependiendo la empresa o a la persona al que iba dirigido. A principios de la década de los 70s era común que los desarrolladores compartieran entre ellos libremente sus programas, pero esto cambió a finales de esta misma década en donde las compañías empezaron a imponer restricciones en contra del usuario, con el uso de licencias.

A principios de la década de los 80s las computadoras empezaron a utilizar software y sistemas operativos privativos.

Esto trajo para los desarrolladores muchos inconvenientes y trabas para su trabajo, ya que, por ejemplo, si un desarrollador encontraba problemas en el software de alguna compañía y este estaba dispuesto a arreglarlo sin cobrar nada, la compañía no lo permitía, y lo único que se podía hacer era reportar el problema y esperar a que lo solucionaran, cosa que les llevaba mucho tiempo.

1.3. Ingeniería de Software

Hay varios conceptos sobre la Ingeniería de Software que son necesarios tener en cuenta, siendo algunos los siguientes:

Edgar Serna (2013), lo define de la siguiente forma:

“Es la disciplina ingenieril que proporciona y aplica los métodos y herramientas necesarios para construir software de calidad, ajustado al presupuesto, en un plazo determinado y en un contexto de constante cambio de requisitos. Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, es decir, es la aplicación de ingeniería al software.”

La web de *micarrerauniversitaria.com* (s.f.) define a la Ingeniería de Software de la siguiente manera:

“La Ingeniería de Software es la rama de la ingeniería que estudia todo lo relacionado con la informática o sistemas de computación, con una orientación metódica, ordenada y cuantificable al incremento, ejecución y conservación del software.”

Así mismo, esta misma web *micarrerauniversitaria.com* (2019) agrega:

“La ingeniería de software, también, incorpora el análisis precedente de la situación, el bosquejo del proyecto, el desarrollo del software, el ensayo necesario para comprobar su funcionamiento correcto y poner en funcionamiento el sistema.”

Julián Pérez y Ana Gardey (2009) en la web de *definicion.de* escriben sobre la ingeniería de Software lo siguiente:

“La ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos (software).

Esta disciplina trasciende la actividad de programación, que es el pilar fundamental a la hora de crear una aplicación. El ingeniero de software se encarga de toda la gestión del proyecto para que este se pueda desarrollar en un plazo determinado y con el presupuesto previsto.

La ingeniería de software, por lo tanto, incluye el análisis previo de la situación, el diseño del proyecto, el desarrollo del software, las pruebas necesarias para confirmar su correcto funcionamiento y la implementación del sistema.”

Florencia Ucha (2013) en la web *definicionabc.com* define la Ingeniería de Software de la siguiente forma:

“La Ingeniería de Software es aquella disciplina que se ocupa del desarrollo, la operación y el mantenimiento del software o programas informáticos [...] Entonces, la ingeniería de software implica un trabajo integral, es decir, se produce un análisis del contexto, se diseña el proyecto, se desarrolla el correspondiente software, se efectúan las pruebas para asegurar su correcto funcionamiento y finalmente se implementa el sistema.”

Para *Fritz Bauer (1969)*:

“Ingeniería de Software es el establecimiento y uso de principios robustos de ingeniería, orientados a obtener software que sea fiable y funcione de manera eficiente sobre máquinas reales”.

Según *Sommerville* (2005):

“La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza.” (pag.6)

Mientras que la definición de la IEEE (Institute of Electrical and Electronics Engineers / El Instituto de Ingeniería Eléctrica Electrónica) en su norma [IEEE93a] dice que:

“La ingeniería de software es: 1) La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software. 2) El estudio de enfoques según el punto 1”

Sintetizando los conceptos anteriores nos es posible conceptualizar que la Ingeniería de Software es una práctica que se encarga mediante métodos y técnicas de desarrollar software de todo tipo y a darle mantenimiento al mismo, para evitar y corregir errores.

1.3.1. Objetivos de la Ingeniería de software

Según *Olga Roa (s.f.)*, profesora de la Pontificia Universidad Javeriana de Cali, menciona respecto a los objetivos de la Ingeniería de Software que:

“Como elemento aclaratorio, es favorable observar algunos objetivos primordiales, definidos por James Martín, para la programación estructurada. Tales objetivos pueden ser fácilmente atribuibles a la ingeniería de software. Según este criterio, los objetivos primordiales de la ingeniería de software (por lo menos algunos) son:

- Lograr programas de alta calidad de un comportamiento predecible
- Lograr programas que sean fácilmente modificables (y mantenibles)
- Simplificar los programas y el proceso de desarrollo de programas
- Lograr mejores predicciones y controles en el proceso de desarrollo
- Acelerar el desarrollo de sistemas

- Aminorar los costos del desarrollo de sistemas “

Así mismo la Escuela Técnica Superior de Ingeniería de Sistemas Informáticos de la Universidad Politécnica de Madrid en su página web menciona sobre los objetivos que tiene la Ingeniería de Software lo siguiente:

- “* Diseñar aplicaciones informáticas que se ajusten a las necesidades de las organizaciones
- * Dirigir y coordinar el desarrollo de aplicaciones complejas.
- * Intervenir en todas las fases del ciclo de vida de un producto.
- * Estimar los costes de un proyecto y determinar los tiempos de desarrollo.
- * Hacer el seguimiento de costes y plazos.
- * Dirigir equipos de trabajo de desarrollo software.
- * Organizar la realización de pruebas que verifiquen el correcto funcionamiento de los programas y que se ajustan a los requisitos de análisis y diseño.
- * Diseñar, construir y administrar bases de datos.
- * Dirigir y asesorar a los programadores durante el desarrollo de aplicaciones.
- * Introducir procedimientos de calidad en los sistemas, evaluando métricas e indicadores y controlando la calidad del software producido.
- * Organizar y supervisar el trabajo de su equipo de los técnicos de mantenimiento y los ingenieros de sistemas y redes. “

De acuerdo a las citas anteriores, se apreció que los objetivos siempre van dirigidos al mejoramiento y mantenimiento del software.

1.3.2. Breve historia de la ingeniería de software

La ingeniería de software surge a finales de los años 60s, en una conferencia para analizar la llamada “Crisis de software”.

Sobre la crisis del software Pons (2010) dice:

“La crisis del software es un término informático acuñado en 1968, en la primera conferencia organizada por la OTAN sobre desarrollo de software. La causa de esta crisis reside en la complejidad inherente a la tarea de construir sistemas de software, y también en los cambios constantes a los que tiene que someterse el software para adaptarse a las necesidades cambiantes de los usuarios y a las innovaciones tecnológicas.”

Claudia Pons, Roxana Giandini y Gabriela Pérez, en el libro *Desarrollo de Software dirigido por modelos (2010)* mencionan sobre la “crisis del software” lo siguiente:

“Históricamente, el proceso de desarrollo de software ha resultado caro, riesgoso, incierto y demasiado lento para las condiciones de negocio modernas. Estos inconvenientes dieron origen al concepto de “crisis del software” que prácticamente surgió conjuntamente con la creación del software. La crisis del software es un término informático acuñado en 1968, en la primera conferencia organizada por la OTAN sobre desarrollo de software. La causa de esta crisis reside en la complejidad inherente a la tarea de construir sistemas de software, y también en los cambios constantes a los que tiene que someterse el software para adaptarse a las necesidades cambiantes de los usuarios y a las innovaciones tecnológicas.”

El autor Gonzalo León Serrano en su libro *Ingeniería de Sistemas de Software (1996)* menciona la evolución de la Ingeniería de Software en 4 etapas, las cuales son:

1. La programación como base del desarrollo (1955-1965).
2. La génesis (1965-1975).
3. La consolidación (1975-1985).
4. Hacia una ingeniería (1985-1995).

De las cuales menciona lo siguiente:

“• La programación como base del desarrollo (1955-1965).

Énfasis absoluto en la tarea de escribir el código en un lenguaje de programación. Alrededor de los nuevos lenguajes de alto nivel, los programadores se alejan de la estructura de los ordenadores y comienzan a acercarse a la complejidad de las aplicaciones de usuario.

• La génesis (1965-1975).

Ligada a la crisis de la programación se plantea la necesidad de controlar el proceso de desarrollo. Se definen modelos de ciclo de vida como una referencia en la que enmarcar las actividades requeridas [...]. El concepto de ciclo de vida en cascada surge de la necesidad del Departamento de Defensa de EE.UU. de disponer de una documentación normalizada para todas las etapas del desarrollo y poder controlar en base a ella a los suministradores de productos software.

• La consolidación (1975-1985).

El control de las actividades de desarrollo debería permitir gestionar el proceso. Durante esta etapa aparecen métricas para estimar a priori el coste o el tamaño del sistema; se difunde el uso de métodos de desarrollo. Con ello, el programador se convierte en analista, diseñador o gestor. Se vislumbra la idea de ingeniero (software).

• Hacia una ingeniería (1985-1995).

Aceptando una consolidación de las tecnologías de software, la mejora viene de la mano de un mejor conocimiento de los procesos con el fin de incrementar la calidad de los productos. Aparece una gestión sofisticada del proceso de desarrollo ligada al control de riesgos y a la madurez de los procesos.”

Algunos de los detonantes en la crisis del software fueron:

- Que el desarrollo de software era algo nuevo para la época, por lo que no se tenía la suficiente capacitación para llevarlo a cabo de forma correcta. Y en su mayoría había aprendido en base a experimentación.

- Las pocas técnicas de programación empezaban a quedarse obsoletas, esto debido a que en esos años el desarrollar software se veía más como un trabajo artístico, donde no había lineamientos ni estándares.
- La aparición de nuevo hardware complicaba el desarrollo de nuevo software, ya que este no se podía adaptar a las nuevas especificaciones y por lo tanto se tenía que hacer los proyectos desde cero.
- La mala redacción de código, errores en los requerimientos de hardware, errores de diseño, y errores de documentación.
- Por esta razón, para poder solucionar todos los problemas que se plantearon en la conferencia se decidió crear una nueva rama para la ingeniería, la Ingeniería de Software.
- Abarcando a todo lo que a software se refiere, como es la documentación necesaria para instalar, usar, desarrollar y darle el mantenimiento necesario a esos programas.

1.3.3. Evolución de la Ingeniería de Software

La Ingeniería de Software ha evolucionado constantemente, de acuerdo a la información del artículo publicado por Piattini, M. (2016) para la revista *Bit & Byte*, año 2. (Número 4), podemos sintetizar la evolución de la Ingeniería de Software de la siguiente manera:

➤ **1940 – 1950**

En estas décadas el hardware tenía un costo mucho más elevado a comparación del software, teniendo como consecuencia una importancia menor. Se consideraba que el software se podía desarrollar de la misma manera que el hardware; y los primeros ingenieros ocupados del software también desarrollaban el hardware.

➤ **1960 -1970**

En esta década se empieza a hacer cada vez más evidente la diferencia del hardware y del software y que este no puede ser tratado de la misma manera. Es la época de los famosos “códigos espagueti”, que eran códigos muy difíciles de entender, incluso por quien los escribía; También es la época de los llamados “héroes” que después de varias noches sin dormir conseguían arreglar a último minuto el software para cumplir los plazos marcados.

En la NASA (*National Aeronautics and Space Administration /La Administración Nacional de la Aeronáutica y del Espacio*) /IEEE de 1966; y las conferencias de la OTAN (*La Organización del Tratado del Atlántico Norte*) en 1968 y 1969, se analizó la llamada “crisis del software”, y se plantearon ideas fundamentales, como lo es la “reutilización” o “arquitectura del software”. El artículo de Dijkstra “To Statement Considered Hatmful (A la declaración considerada perjudicial) impulso la programación estructurada y en el congreso de IFIP (*International Federation for Information Processing / La Federación Internacional para el Procesamiento de la Información*) se citó por primera vez el concepto de “factoría o fábrica de software”.

➤ **1970 – 1980**

En esta década las organizaciones empezaron a comprobar que el costo del software supera a los del hardware. David Parnas (de los primeros pioneros canadienses de la ingeniería de software) propone la descomposición modular y el concepto de ocultamiento de información (information hiding), Peter. P. Chen el modelo Entidad-Relacion y Winston W. Royce el modelo de ciclo de vida en cascada.

La formación de los profesionales de la Ingeniería del Software se centra entonces en las metodologías estructuradas (Warnier, Jackson, Myers, Yourdon y Constantine, Gane y Sarson, Demarco, etc.) que supusieron un avance importante en el análisis y diseño de software.

➤ **1980 - 1990**

En esta década se automatizó parte del ciclo de vida del software, apareciendo la primera generación de herramientas CASE (*Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora*) y los lenguajes de programación orientados a objetos.

Estos ya habían aparecido a finales de los sesenta, pero fue en esta década donde empezaron a ganar popularidad, con la aparición de C++, Objective-C y Eiffel.

La formación de los profesionales del software requiere entonces el manejo de las herramientas CASE, comprender el gran cambio de paradigma que supone la orientación a objetos, y adquirir conocimientos sobre los procesos software y los modelos de madurez.

➤ **1990 - 2000**

En esta década se desarrollaron los modelos relacionados con la mejora de procesos de software, como son Ideal (*International Digital Electronic Access Library / Biblioteca internacional de acceso electrónico digital*), TSP (*Team Software Process / Equipo de Proceso de Software*) o PSP (*Personal Software Process / Proceso de Software Personal*), y las normas y estándares de calidad como la ISO 9126, ISO 12207, ISO 9000-3, etc. En esta década también se consolidó la orientación a objetos como aproximación para el desarrollo de sistemas informáticos y con esto aparecieron más de 100 metodologías, que terminaron por dar lugar a la aparición del UML (Lenguaje Modelado Unificado)

Los clásicos problemas del mantenimiento de software, hicieron plantearse a muchas organizaciones la conveniencia de externalizar (outsourcing) sus procesos de mantenimiento, impulsando la creación por parte de muchas empresas de centros y unidades dedicadas específicamente a la externalización.

➤ **2000 - 2010**

Se firma el “Manifiesto Ágil” como intento de simplificar la complejidad de las metodologías existentes y en respuesta a los modelos “pesados” tipo CMM (Capability Maturity Model / Modelo de Capacidad y Madurez). También surgen los métodos híbridos, que combinan la adaptabilidad de los ágiles con la formalidad y documentación de los métodos rigurosos. Cabe destacar también que en esta década se difunden el Desarrollo Software Dirigido por Modelos y las líneas o familias de productos software, que suponen un esfuerzo al Ingeniero del Software al trabajar con modelos de alto nivel como elemento principal del desarrollo y mantenimiento de software.

➤ 2010 - Actualidad

En esta década es mayor la integración entre la Ingeniería del Software y la Ingeniería de Sistemas.

También surge la Ingeniería del Software Continua, y su correspondiente tecnología y filosofía DevOps (Development Operations / Desarrollo de operaciones), que logran reducir el tiempo entre que se compromete un cambio en el sistema y que se ponga en producción normal.

De acuerdo a lo anterior, aunque la Ingeniería de Software ha crecido en capacidades considerables aún le falta evolucionar para poder adaptarse a las nuevas tecnologías que están por surgir. Es por esto que también el ingeniero en computación debe aprender y mantenerse en constante crecimiento para que pueda resolver estos problemas y crear software de calidad.

1.4. Concepto de Calidad

Normalmente asociamos este término a que es algo que está bien hecho, pero, ¿realmente eso es todo lo que significa? El concepto de calidad según la *American Society For Quality Control (1974)* define la calidad como:

“la totalidad de los rasgos y características de un producto o servicio que se sustenta en su habilidad para satisfacer las necesidades establecidas o implícitas”

Mientras que la *International Standard Organization*, mejor conocida como *ISO* en su norma 9000, menciona lo siguiente:

“la totalidad de las características de una entidad (proceso, producto organismo, sistema o persona) que le confieren aptitud para satisfacer las necesidades establecidas e implícitas”

Por lo tanto, el término de calidad se puede resumir como las características que debe de tener un producto para cumplir con lo establecido y satisfacer las necesidades del consumidor.

1.4.1. Orígenes de la Calidad

Sintetizando el escrito de Jose Cruz Ramirez (s.f.) sobre los orígenes de la calidad tenemos que los primeros estudios sobre calidad se hicieron en los años 30, antes de la Segunda Guerra Mundial, donde la calidad no mejoraba sustancialmente, pero continuamente se experimentaba para lograr que esta mejorará. Los primeros estudios sobre calidad se hicieron en Estados Unidos.

En el año de 1933 el Doctor Walter Shewhart, de los Bell Laboratories aplicó el concepto de control estadístico por primera vez con propósitos industriales, cuyo propósito era mejorar en términos de costo-beneficio las líneas de producción; el resultado obtenido fue que gracias al uso de la estadística de manera eficiente se elevó la productividad y se disminuyeron los errores, estableciendo un análisis específico del origen de las mermas (*“Técnicamente una merma es una pérdida de utilidades en término físico.” Wikipedia 2019*), con el fin de elevar la productividad y la calidad.

Walter Shewhart maestro de Joseph Juran, fue un matemático que introdujo la estadística como medio de gestión de la calidad en los años de 1931 y 1935, debido a esto se convirtió en uno de los hombres más buscados por el ejército norteamericano.

Al estallar la Segunda Guerra Mundial en 1939, el control estadístico del proceso se convirtió en un arma secreta de la industria, esto llevó a los norteamericanos a crear el primer sistema de aseguramiento de la calidad vigente en el mundo.

El objetivo principal de esta creación era establecer con absoluta claridad, que a través de un sistema novedoso era posible garantizar los estándares de calidad, de tal manera que se evitara, sobre todo, la pérdida de vidas humanas; uno de los principales interesados en esto fue la industria militar estadounidense, ya que para los militares era indispensable el evitar que muchos jóvenes norteamericanos murieran porque simplemente sus paracaídas no abrían.

En octubre de 1942 de cada mil paracaídas fabricados, por lo menos un 3.45 no se abrieron, lo que significó una gran cantidad de jóvenes soldados norteamericanos muertos, como

consecuencia de esto a partir de 1943 se intensificó la búsqueda para establecer los estándares de calidad para evitar aquella tragedia, pero no solamente los paracaídas estaban defectuosos, también hubo una gran cantidad de fallas en el armamento de Estados Unidos que proporcionaban a su vez a sus aliados. Las fallas eran encontradas principalmente en el armamento pesado.

Para elevar la calidad se crearon las primeras normas de calidad del mundo mediante el concepto moderno del aseguramiento de la calidad, para lograr crear un verdadero control de calidad se ideó un sistema de certificación de la calidad, que el ejército de los Estados Unidos inició desde antes de la guerra. Estas normas funcionaron precisamente en la industria militar donde fueron llamados como las normas Z1, estas tuvieron un gran éxito para la industria norteamericana permitiendo elevar los estándares de calidad evitando así el derroche de vidas humanas; En 1935 Gran Bretaña también aplicó con el apoyo de los Estados Unidos una serie de normas de calidad a su industria militar. A las normas de calidad británicas se les conoce como el sistema de normas 600; Para los británicos era importante participar en la guerra con un mejor armamento con garantías de calidad, así que adoptaron la norma norteamericana Z1.

Con la aparición de las “Normas Británicas 1008”, los británicos garantizaron mayores estándares de calidad en sus equipos. A su vez, otros países del mundo no contaban con aseguramiento de calidad tan efectivo, esto llevó a los Estados Unidos y Gran Bretaña a elevar el nivel de productividad de sus equipos, bajar el número de muertes humanas ocasionadas por la mala calidad del mismo, y por supuesto, garantizaron establecer garantías de calidad sobre su equipo, aparatos y elementos técnicos primero que ninguna otra nación en el mundo. Países como la URSS, Japón, Alemania tuvieron estándares de calidad, pero no se comparaban a los estadounidenses que eran por mucho superiores; esto determinó en gran medida a que las muertes humanas fueran mucho mayores de su lado.

Entre los años 1940 y 1943 el doctor Edwards Deming uno de los grandes estadistas de su tiempo, trabajó en la Universidad de Stanford capacitando a cientos de ingenieros militares en el control estadístico del proceso, muchos de estos militares fueron capacitados en la implementación de las normas de calidad Z1 a través de una serie de entrenamientos en donde

el aseguramiento de la calidad era lo primordial. Entre los años 1942 y 1945 Deming contribuye a mejorar la calidad de la industria norteamericana dedicada a la guerra. Al finalizar la guerra Deming fue invitado por el comando militar de ocupación norteamericana a Japón, ahí tuvo un papel fundamental en cuanto a la elevación de la calidad; ya que en 1947 inició sus primeros contactos con ingenieros japoneses, ya en 1950 fue invitado por el presidente de la JUSE (Unión de Ingenieros Científicos Japoneses) a impartir cursos que iniciaron en 1950.

Aquí fue donde Deming el ya llamado “padre de la calidad japonesa” usó en Japón su modelo administrativo para el manejo de la calidad. Qué fue de gran ayuda debido a que en Japón no se tenían antecedentes claros de la calidad, y su calidad era un completo desastre.

La fama de Deming llegó a Estados Unidos 20 años después de sus pláticas en Japón, cuando Kinishi Koyanagi, un extraordinario implementador de calidad en Japón llegó a Estados Unidos e imparte un seminario en Rochester, en este seminario habló sobre los 13 progresos de las trece compañías japonesas que elevaron su calidad gracias a la teoría de Deming.

Debe quedar claro que la calidad tanto en Europa como en Japón y Estados Unidos detonó después de terminar la Segunda Guerra Mundial y que en este periodo las naciones del mundo se organizaron para crear y elevar los estándares de la calidad, es por ello que el antecedente de la ISO ligado a hace 50 años, cuando entre 1950 y 1996 la calidad se convirtió en una súper tendencia en el mundo entero.

El papel que jugaron los japoneses en el procedimiento de la calidad fue importante, pero no fueron solo ellos los que invirtieron en el procedimiento de la elevación de la calidad, también Alemania inició un impulso sorprendente por elevar la calidad y convertirla en algo fundamental; en Estados Unidos Joseph Juran fue un detonador esencial para este desarrollo, a los nombres de Deming en Japón y Joseph Juran también en Japón debe agregarse el nombre de Phillip Crosby que en los años de 1960 inició una revolución de la calidad en Estados Unidos, a su vez Armand Feigenbaum, fue otro gran impulsor del control de calidad.

El Doctor Armand V. Feigenbaum fue quien creó el concepto de gestión de la calidad o de gestionar la calidad, también introdujo un programa de calidad de la General Electric que aplicó por primera vez en Total Quality Control en Estados Unidos. En 1951 aparece su libro “Total Quality Control” que fue exitoso en Europa a partir del año 1961. Fue nombrado director de todas las unidades de producción del mundo de la General Electric y así difundió en esa compañía sus conocimientos acerca de calidad.

Tanto a Ishikawa como a sus discípulos se le atribuyen los cambios que tuvo Japón, ya que se desarrolló un modelo nacional de calidad extraordinario.

El control de calidad nació bajo el concepto japonés de calidad total, y bajo el concepto de las normas de calidad ISO 9000, y bajo otros conceptos que buscan la gestión de calidad como objetivo estratégico.

Entre enero y julio de 1951, los japoneses aplicaron los conceptos de mejora continua de Deming en 45 plantas. Estas plantas tuvieron éxito en cuanto a la implementación de sistemas que permitieron elevar la calidad, debido a esto a partir del verano de 1951 los japoneses quedaron muy agradecidos hacia Deming y esto los llevó a la creación del “Premio Deming a la Calidad” que a partir de noviembre de 1951 es entregado a una empresa o institución. En la actualidad el “Premio Deming” se entrega a la organización que mejora sus procesos administrativos o productivos. Cabe decir que este premio es el antecedente más remoto de otros premios internacionales como el Premio Malcom Baldrige de Estados Unidos.

Las políticas para elevar la calidad y la productividad han servido de herramienta de diagnóstico a una gran cantidad de organizaciones y empresas ya que el movimiento mundial por la calidad se convirtió en una mega tendencia desde 1952.

China entre los años de 1990 y 1995 capacitó a más de 70 millones de personas en el conocimiento de la calidad total; en Japón hay más de 10,000 especialistas en calidad donde aparte más de 5 millones de personas se les puede considerar perfectamente entrenadas en el manejo de las disciplinas de calidad; en Canadá más de 5,000 personas también son

especialistas en calidad y en los Estados Unidos la cifra puede llegar a ser de casi 350,000 personas capacitadas y habilitadas en el manejo de sistemas de mejora continua. Es por eso que se dice que la calidad se ha globalizado.

En la década de 1990 las normas han sido actualizadas, a partir del año 2000 la ISO 9000 regula a los sistemas de comercio mundial en Occidente.

Por otro lado, la calidad se desarrolló en Latinoamérica poco a poco. Algunos de los países de América Latina como Argentina, Brasil, México, Venezuela, Colombia, Chile, Perú iniciaron grandes intentos en torno a la calidad, pero aun así las industrias de estos países carecieron de los niveles de competitividad internacional

La era de la información enfocada al cliente, la era de la calidad, de la nueva competitividad, el nacimiento de Asia como un nuevo poder global, y de Japón como amo del siglo XXI, fue gracias a la globalización de la calidad, una nueva estrategia para competir, entender las necesidades del cliente, y satisfacer la demanda de los mercados.

De acuerdo con lo anterior se puede decir que la calidad surgió como una necesidad que actualmente se ha ido agigantando, siendo necesaria para todo servicio en la actualidad y es gracias a las doctrinas que se han aplicado que está evolucionando desde sus inicios.

1.4.2. Concepto de control

Algunas de las definiciones de control que la RAE (2014) en su vigesimotercera edición menciona son las siguientes:

- “1. m. Comprobación, inspección, fiscalización, intervención.
2. m. Dominio, mando, preponderancia.
3. m. Oficina, despacho, dependencia, etc., donde se controla.
4. m. puesto de control.

5. m. Regulación, manual o automática, sobre un sistema.”

Como se puede ver este concepto puede llegar a ser algo caótico si no se sabe a qué se refiere o como es se va a aplicar. Imagina que en un trabajo de desarrollador de software te dicen que de ti depende el control del equipo de trabajo y lo primero que haces es amordazar a tus colaboradores para que no dejen de trabajar. ¿Suenan raro, no?, pero debemos de saber la importancia de lo que esto significa; ya que la definición de calidad puede cambiar dependiendo de la percepción personal de cada individuo.

Antes de que existieran las normas que se tienen hoy en día, se llegaron a aplicar no muy satisfactoriamente distintas formas de control, pero estas fallaban por varias razones, siendo algunas las siguientes:

1. No se tenía muy en claro a qué se referían con “control” y como aplicarlo, este era aplicado a como mejor les funcionara, siendo más idealista.
2. No se tenían técnicas de control, ni análisis basado en estadísticas. No tenían métodos ni procesos, ya que abundaba un enorme conformismo, e incluso cuando se cometieron errores, no se adoptaban medidas para evitar su reaparición.
3. Se utilizaban sólo métodos complicados, el control no se ejercía desde un punto de vista amplio y general. Mientras que tampoco había información para todos los trabajadores, sólo los que estaban en un cargo de responsabilidad eran informados.
4. No había una buena comunicación con todos los departamentos, haciendo que las órdenes fueran contradictorias.

Según Ishikawa (1989) dice que el control consiste en:

" ... comprobar si el trabajo se está haciendo de acuerdo con las políticas, las órdenes, los planes y las normas y, si no, adoptar la acción para corregir cualquier desviación y evitar su reaparición, y luego seguir con el plan "

Por lo anterior el control sería una inspección a los métodos de producción, verificando si este cumple con las normas, políticas, ordenes, planes, etc., que se le son impuestas, con el fin de entregar un producto de calidad. Y en caso de que no cumpla con alguna de las características requeridas, corregirla y así evitar la reaparición de este u otros problemas relacionados.

1.4.3. Control de Calidad

Sobre el control de calidad Ishikawa (1989) afirma lo siguiente:

"El control de calidad consiste en el desarrollo, diseño, producción, comercialización y prestación del servicio de productos y servicios con una eficacia del coste y una utilidad óptimas, y que los clientes comprarán con satisfacción." (Pág. 2)

Con esto se crea una forma de verificar los estándares del producto o servicio a lo largo del proceso de fabricación y disminuir la probabilidad de llevar el producto incorrecto al mercado. Su nombre significa control de calidad. La definición final son las condiciones que debe cumplir un producto para que el consumidor o cliente consuma el producto.

Es importante que cualquier sistema industrial tenga un control de calidad ya que con esto se puede monitorear las actividades de producción y eliminar errores, omisiones o defectos. Además, de que evalúa la calidad del sistema y las razones de estas distorsiones y errores en el diseño del producto se deben a otros métodos que se han identificado y corregido, por lo que se pueden mejorar y corregir.

1.4.3.1 Implementación del control de calidad

Para implementar el control de calidad, debe identificar los factores clave que deben revisarse en el proceso de fabricación y luego identificar las herramientas para evaluar estos pasos.

Es importante recordar que el control de calidad se logra no solo en todas las partes de la cadena de producción, sino también en toda la cadena de producción a través de un sistema de gestión de inspección, verificación visual, pruebas y análisis. Sin embargo, todos los miembros de una organización pueden ejercer un gran poder en cualquier momento. Esto significa que el control de calidad es responsabilidad de todas las partes involucradas en la fabricación de productos y servicios, notificaciones de errores, notificaciones de fallas o métodos de implementación inadecuados.

IVE Consultores menciona lo siguiente acerca de la implementación del control de calidad en un producto

“1. Define el lote y muestra

Se trata de la parte inicial antes de empezar a realizar las inspecciones oportunas. Debes definir qué es lo que vas a inspeccionar.

De toda tu producción diaria, qué lote vas a inspeccionar.

Y dentro del lote, qué muestra vas a seleccionar como un número de unidades representativas que te permitan extrapolar los resultados para toda la producción.

2. Define el estándar

Define para ese producto qué atributos y variables vas a medir que determinan que la fabricación del mismo está dentro del estándar de calidad que hayas determinado.

3. Recolecta las unidades de la muestra

Recolecta las unidades que vas a inspeccionar y analiza si cumplen o no los estándares en relación a los atributos y variables que has definido previamente.

Esta recolecta de unidades la puedes llevar a cabo:

- Durante la producción: en algún momento, a lo largo de los distintos pasos de la producción.
- Después de la producción: al finalizar la producción.

4. Realiza el control de calidad

Este es el momento en el que inspeccionas cada unidad recolectada para aceptarla o rechazarla según los estándares de calidad establecidos en base a los atributos y variables.

En esta inspección sólo pueden ocurrir dos cosas:

- Que la unidad seleccionada NO tenga defectos.
- Que la unidad seleccionada SÍ tenga defectos.

Para las unidades que has detectado defectuosas decide si:

- Reprocesarla: en caso de que se puedan subsanar los defectos y se pueda convertir en una unidad con la calidad exigida.
- Desecharla: en caso de que no se puedan subsanar los defectos y por tanto la unidad no cumpla los estándares de calidad exigidos.

En ocasiones y dependiendo del número de unidades desechadas podrás tomar la decisión de desechar el lote completo producido ya que el riesgo de que existan más unidades defectuosas que quedaron fuera de la muestra es alto.

5. Analiza las unidades desechadas

Averigua si para las unidades desechadas existe una causa común que produce la degradación de la calidad y por lo tanto el incumplimiento del estándar fijado.

Podrías analizar también las unidades reprocesadas, pero yo me centraría sólo en las unidades desechadas ya que son las que representan el mayor coste de “no calidad” para dichas unidades.

6. Toma una decisión

Determina si debes de modificar algo en el proceso productivo: ajustes de equipos, formación al personal, modificación de materia prima utilizada.”

1.5. Calidad en el proceso de desarrollo de Software

Pressman (2010) menciona dice sobre la mala calidad de software lo siguiente:

“Actualmente, la calidad del software es preocupante, pero, ¿de quién es la culpa? Los clientes culpan a los desarrolladores, pues afirman que sus prácticas descuidadas producen software de mala calidad. Los desarrolladores culpan a los clientes (y a otros participantes) con la afirmación de que las fechas de entrega irracionales y un flujo continuo de cambios los obligan a entregar software antes de haber sido validado por completo. ¿Quién tiene la razón? Ambos, y ése es el problema.”

Esta declaración es bastante acertada, pues en el proceso de desarrollo de software tanto las malas prácticas de los desarrolladores, como lo tornadizos y poca comunicación que tiene el cliente con el desarrollador, llega a dar como resultado software de mala calidad.

Pero no todo es malo, actualmente la calidad en el proceso de desarrollo de software ha avanzado mucho, esto debido a que las compañías de desarrollo han adoptado y puesto en marcha nuevas técnicas y tecnologías, como es el caso del desarrollo orientado a objetos y el soporte asociado de herramientas CASE. Al mismo tiempo se han adoptado técnicas de control de calidad en las industrias de desarrollo de software. No obstante, la calidad en el desarrollo de software es un concepto que no es directamente comparable con la calidad en la manufactura de un producto. (Somerville, 2005). Ya que se tienen que tratar de una forma distinta, aunque lo ideal fuera que se trataran de la misma forma.

Somerville (2005) menciona los siguientes problemas que ocurren con respecto al software:

“1. La especificación se orienta hacia las características del producto que el consumidor quiere. Sin embargo, la organización desarrolladora también tiene requerimientos (como los de mantenimiento) que no se incluyen en la especificación.

2. No se sabe cómo especificar ciertas características de calidad (por ejemplo, mantenimiento) de una forma no ambigua.
3. Como se indicó en la Parte 1, en la que se estudió la ingeniería de requerimientos, es muy difícil redactar especificaciones concretas de software. Por lo tanto, aunque un producto se ajuste a su especificación, los usuarios no lo consideran un producto de alta calidad debido a que no responde a sus expectativas.”

Dado los siguientes puntos podemos inferir que los procedimientos de calidad no pueden ser tomados y aplicados de manera perfecta, porque los requerimientos son variados respecto a quien va dirigido. Aun así, el tener un software de calidad es la meta que tienen la mayoría de desarrolladores. Pressman (2010) hace énfasis en 3 puntos importantes sobre el desarrollo de software, tales puntos son:

“1. Un *proceso eficaz de software* establece la infraestructura que da apoyo a cualquier esfuerzo de elaboración de un producto de software de alta calidad. Los aspectos de administración del proceso generan las verificaciones y equilibrios que ayudan a evitar que el proyecto caiga en el caos, contribuyente clave de la mala calidad. Las prácticas de ingeniería de software permiten al desarrollador analizar el problema y diseñar una solución sólida, ambas actividades críticas de la construcción de software de alta calidad. Por último, las actividades sombrija, tales como administración del cambio y revisiones técnicas, tienen tanto que ver con la calidad como cualquier otra parte de la práctica de la ingeniería de software.

2. Un *producto útil* entrega contenido, funciones y características que el usuario final desea; sin embargo, de igual importancia es que entrega estos activos en forma confiable y libre de errores. Un producto útil siempre satisface los requerimientos establecidos en forma explícita por los participantes. Además, satisface el conjunto de requerimientos (por ejemplo, la facilidad de uso) con los que se espera que cuente el software de alta calidad.

3. Al *agregar valor para el productor y para el usuario* de un producto, el software de alta calidad proporciona beneficios a la organización que lo produce y a la comunidad de usuarios finales. La organización que elabora el software obtiene valor agregado porque el software de alta calidad requiere un menor esfuerzo de mantenimiento, menos errores que corregir y poca asistencia al cliente. Esto permite que los ingenieros de software dediquen más tiempo a crear nuevas aplicaciones y menos a repetir trabajos mal hechos. La comunidad de usuarios obtiene valor agregado porque la aplicación provee una capacidad útil en forma tal que agiliza algún proceso de negocios. El resultado final es 1) mayores utilidades por el producto de software, 2) más rentabilidad

cuando una aplicación apoya un proceso de negocios y 3) mejor disponibilidad de información, que es crucial para el negocio.” [Pressman Ingeniería de software. Un enfoque practico (2010) (Pág. 340-341)]

Estos tres puntos definen en profundidad el concepto de “calidad en el software”, mencionando las 3 características que tiene; tener un proceso eficaz, ser un producto útil, y agregar valor para el productor y el usuario.

Con todo esto es importante mencionar el papel que juega la gestión de calidad en el proceso de elaboración de software, Somerville (2005) afirma que:

“La gestión de la calidad provee una comprobación independiente de los procesos de desarrollo software. Los procesos de gestión de la calidad comprueban las entregas del proyecto para asegurarse que concuerdan con los estándares y metas organizacionales...”

Es importante recalcar que el equipo encargado de la gestión de calidad, debe de ser un equipo independiente, que no esté asociado con ningún grupo de desarrollo, y deberá informar al gestor de proyecto todas las comprobaciones llevadas a cabo. *La razón de esto es que los gestores del proyecto deben mantener el presupuesto y la agenda. Si aparecen problemas, éstos pueden verse tentados de comprometer la calidad del producto para mantener su agenda.* Por lo que con esta independencia en el equipo de calidad no se verán comprometidas la calidad y los objetivos organizacionales por consideraciones de presupuesto y agenda. Somerville (2005)

Sobre las actividades principales en la gestión de calidad del software, Somerville dice que:

“La gestión de calidad del software se estructura en tres actividades principales:

1. Garantía de la calidad. El establecimiento de un marco de trabajo de procedimientos
2. Planificación de la calidad. La selección de procedimientos y estándares adecuados a partir de este marco de trabajo y la adaptación de éstos para un proyecto software específico.
3. Control de la calidad. La definición y fomento de los procesos que garanticen que los procedimientos y estándares para la calidad del proyecto son seguidos por el equipo de desarrollo de software. (Pág. 589)”

De acuerdo a lo anterior las actividades en la gestión de calidad del software están hechas para tener un mayor control sobre el proceso del producto.

1.5.1. Casos de errores en el software.

Como ya se ha mencionado anteriormente, los fallos en el software pueden llegar a ser catastróficos. A continuación, se hace un listado recogiendo de varias páginas de Internet (Actualidad RT, OpenMind BBVA, JavierGarzas.com) algunos accidentes ocasionados por fallas en el software.

1. El cohete Ariane 5, lanzado el 4 de junio de 1996 por la Agencia Espacial Europea, explotó 37 segundos después de ser lanzado por un error de programación en el módulo de gestión haciendo que el número real de 64 bits (coma flotante) relacionado con la velocidad se convirtiera en un entero de 16 bits. Y cómo estas variables ocupan un diferente tamaño en la memoria provocaron una serie de bugs que afectaron a las computadoras de vuelo y congelaron todo el hardware de la nave. Teniendo un costo de 500 millones de dólares, según la web de JavierGarzas.com
2. La sonda espacial Mars Climate Orbiter de la NASA, lanzada a Marte el 11 de diciembre de 1998, después de sobrevolar Marte por 286 días, explotó en su atmósfera. Esto debido a que el software calculó mal la distancia sobrevolando a 57 km de los 150 previstos. Se dice que este fallo fue por convertir mal las millas inglesas a kilómetros. Este fallo costó según la web de Actualidad RT fue de 327 millones de dólares.
3. La empresa de inversiones Knight Capital perdió 500 millones de dólares en tan solo media hora debido a que sus computadoras empezaron a comprar y vender millones de acciones fuera de control. Teniendo también como consecuencia que el precio de sus acciones cayese en un 75% en dos días.
4. La sonda espacial Mariner I, lanzada en 1962, tuvo que ser destruida 293 segundos después de su lanzamiento debido a que tuvo una falla en su trayectoria, esto debido a

que su software hizo omisión de un guion en la lectura de su programa, sacando a la sonda de su ruta prevista. Según la web de JavierGarzas.com este fallo tuvo un costo de 18,5 millones de dólares

5. La máquina de radioterapia Therac-25 causó entre los años de 1985 y 1987 al menos 6 accidentes en donde los pacientes recibieron sobredosis extremas de radiación. Debido a un error en su código de programación que obligaba al programa a repetir la misma acción en varias ocasiones. Como consecuencia de esto varios pacientes perdieron la vida por exceso de radiación.
6. El error del microprocesador Intel Pentium en el año de 1994, el profesor Thomas Nicely de la Universidad de Lynchburg descubrió un error en la unidad de coma flotante del Intel Pentium haciendo que devolviera un valor erróneo por exceso, como consecuencia Intel tuvo que sustituir todos sus microprocesadores defectuosos. Según la web de JavierGarzas.com este reemplazo de microprocesadores tuvo un costo de 475 millones de dólares.
7. El sistema de defensa antimisiles estadounidense Patriot en Dhahran (Arabia Saudita) que durante la guerra del golfo el 25 de febrero de 1991, tuvo un fallo en su software provocando que funcionara con retraso, por consecuencia no seguía el lanzamiento de misiles en tiempo real. Este retraso provocó que no detectara un misil Scud – Al Hussein, impactando en una base militar estadounidense desencadenando la muerte de 28 norteamericanos y cientos de heridos.
8. El efecto 2000 fue causado por la mala costumbre de los programadores de omitir la centuria en el año para el almacenamiento de fechas, esto lo hacían para economizar memoria, dando como resultado de que las máquinas no mostraban la fecha correctamente pasando del 31 de diciembre de 1999 a 1 de enero de 1900 en lugar de poner el año 2000. Según la web de JavierGarzas.com el costo por reparar estos daños fue de 296,7 billones de dólares.

1.6. Normas y estándares de calidad utilizados para el desarrollo de software.

Las normas y estándares tienen un papel de gran importancia para el desarrollo de software, ya que estas manejan reglas establecidas y parámetros importantes para su buen desarrollo.

A continuación, se muestran las normas y estándares de calidad que se utilizan en el desarrollo de software y proyectos informáticos.

1.6.1. CMMI (Capability Maturity Model Integration / Modelo de la Madurez de Capacidades de Integración).

Gómez (2017) a través de la página MegaPractical dice sobre el CMMI:

“CMMI es un modelo que contiene las mejores prácticas y que provee a las organizaciones de aquellos elementos que son esenciales para que los procesos de negocio de las mismas sean efectivos.”

Con esto podemos entender que el CMMI es un conjunto de prácticas para mejorar los procesos de desarrollo de proyectos, gestión de proveedores y gestión de servicios. Gomez (2017) en la misma web de MegaPractical menciona lo siguiente sobre el surgimiento de este modelo:

“El modelo CMMI fue inicialmente desarrollado para los procesos relativos al desarrollo e implementación de Software por la Carnegie-Mellon University. Este vio la luz por primera vez en el año 1987 como Capability Maturity Model CMM. Dicho nombre, tanto como los cinco niveles de la representación por etapas, están inspirados en el modelo de madurez Manufacturing Maturity Model de Crosby.

En principio el modelo CMM era aplicado en programas de defensa, pero lo cierto es que este ha logrado gran aceptación, tan es así que ha sido sometido a varias revisiones e iteraciones. Debido a su éxito se llevó a cabo el desarrollo de modelos CMM para para diversos ámbitos más allá del software.”

Por otro lado, las ventajas que tiene utilizar el CMMI según Gómez (2017) a través de la página MegaPractical son:

“Hacer uso del modelo CMMI para el desarrollo de software, no solo permite optimizar procesos de negocios, sino que también trae consigo una serie de beneficios, entre ellos los siguientes:

- La gestión y la ingeniería de las actividades se encuentran entrelazadas de una manera explícita, tan es así que facilita el reconocimiento de los objetivos del negocio.
- Permite hacer la incorporación de la experiencia adquirida en otras zonas de las mejores prácticas. Algunos ejemplos serían la medición, gestión de riesgos y de proveedores.
- Poder aplicar prácticas de alta madurez mucho más robustas.
- Cumplir de forma mucho más completa con las normas ISO.”

El CMMI representa un metamodelo y cada área de este proceso se valora contra metas y prácticas específicas, es por eso que se clasifica de acuerdo a los niveles de capacidad. Retomando y resumiendo a Pressman (2010) los niveles son los siguientes:

Nivel 0: Incompleto: El área de proceso no se realiza o no logra todas las metas y objetivos definidos por la CMMI

Nivel 1: Realizado: El proceso ya se ejecutó y se logró su objetivo.

Nivel 2: Administrado: Se satisfacen todos los criterios del nivel 1 de capacidad. Y se tiene la capacidad de repetir el éxito de los resultados ya obtenidos.

Nivel 3: Definido: Se logran todos los criterios del nivel 2 de capacidad. La forma de desarrollar los proyectos ya se encuentra establecida, con la documentación necesaria y las métricas utilizadas para su correcta ejecución.

Nivel 4: Administrado cuantitativamente: se logran todos los criterios del nivel 3 de capacidad. Los proyectos utilizan objetivos medidos y con valores cuantificables para cubrir las necesidades de los clientes en la organización. A lo que se reduce como el uso de métricas para gestionar a la organización.

Nivel 5: Optimizado: se logran todos los criterios del nivel 4 de capacidad. El área de proceso se encuentra adaptada y optimizada, utilizando medios cuantitativos para satisfacer las necesidades del cliente y tener una mejora continua.

1.6.2. PSP. Personal Software Process / Proceso de Software Personal.

El proceso de software personal abreviado comúnmente como “PSP” es un conjunto de prácticas, que sirven para ayudar a los ingenieros de software, Humphrey (2001) explica que:

“El Proceso Software Personal (PSP)SM fue diseñado para ayudar a los ingenieros del software a hacer bien su trabajo. Muestra cómo aplicar métodos avanzados de ingeniería a sus tareas diarias. Proporciona métodos detallados de planificación y estimación, muestra a los ingenieros cómo controlar su rendimiento frente a estos planes y explica cómo los procesos definidos guían su trabajo.” (Humphrey, 2001, Pág. 2)

Con esto comprendemos que el PSP pretende disciplinar a los ingenieros de software para que mejoren sus capacidades de desarrollo de software. A parte de lo ya mencionado el PSP tiene como objetivos según la web Wikipedia (2019) los siguientes:

“PSP pretende formar ingenieros de software con métodos disciplinados para mejorar su desarrollo personal de software. PSP le ayuda a los desarrolladores a:

- Mejorar sus habilidades de estimación y planeación.
- Hacer compromisos que se puedan cumplir.
- Administrar la calidad de sus procesos.
- Reducir la cantidad de defectos en sus productos.”

Este aprendizaje del PSP debe ser comparado con las practicas que un médico hace antes de poder ejercer realmente, o todo el tiempo de práctica que realiza un atleta antes de una competencia. Ya que, en el caso del software, si un ingeniero no tiene este entrenamiento

deberá aprenderlos sobre la marcha en el trabajo, y esto representa una pérdida tanto económica como de tiempo, a su vez que representa un riesgo. Humphrey (2001).

Resumiendo lo anterior, podemos decir que el PSP es un método de autoconocimiento, que le sirve al desarrollador como práctica para poderse desempeñar de una manera eficiente en grandes proyectos.

1.6.3. TSP. Team Software Process / Equipo de Proceso de Software

El Equipo de Proceso de Software es similar al PSP, pero este está enfocado en el trabajo en equipo para el desarrollo de programas con varias de miles líneas de código, la Web Wikipedia dice del TSP lo siguiente:

“En combinación con el Personal Software Process (PSP), el llamado Team Software Process (TSP) proporciona un marco de trabajo de procesos definidos que está diseñado para ayudarle a equipos de gerentes e ingenieros a organizar y producir proyectos de software de gran escala, que tengan tamaños mayores a varios miles de líneas de código. El objetivo del TSP es mejorar los niveles de calidad y productividad de un proyecto de desarrollo de software de un equipo, con el fin de ayudarlos a alcanzar los acuerdos de costos y tiempos en dicho desarrollo.” (Wikipedia, 2019)

Para los desarrolladores que quieran aplicar un TSP es necesario que primero dominen el sistema PSP, ya que primero deben conocerse a sí mismos para poder trabajar en armonía con un equipo de trabajo. En el TSP se implementan diferentes fases, según Cristina Rodriguez en su blog web de Sistemas de Calidad dice sobre las fases lo siguiente:

“Dentro de TSP se implementan diversas fases las cuales son:

- Despegue: Está diseñado para comenzar con el proceso de construcción de los equipos y administradores. Establecen metas, definición de roles, evaluación de riesgos y producen un plan de equipo.
- Lanzamiento: Revisión de objetivos a perseguir asignación de equipos y de roles.

- Estrategia: Creación de un diseño conceptual para el producto. Se hacen estimaciones iniciales de esfuerzo y tamaño. Utiliza un plan de administración de la configuración.
- Requerimientos: Se analizan las necesidades de los clientes. Se especifican los requerimientos. Se diseñan un plan de prueba del sistema.
- Diseño: Se crea un diseño de alto nivel, se especifica el diseño y se desarrolla un plan de pruebas de integración.
- Implementación: se requiere la utilización de PSP para la implementación de módulos y unidades, además crea el diseño detallado de cada uno de los módulos y unidades.
- Pruebas: En él se construye y se integra el sistema, además se lleva a cabo las pruebas del sistema y se produce la documentación del usuario.
- Después de la Muerte (Postmortem): Se realizan los análisis de los resultados, además se escribe el reporte del ciclo.”

Estas fases ayudan a los equipos de trabajo a tener un buen desarrollo de software. Los objetivos que tiene el TSP según la página web Gutimar Soluciones (2012) son:

“* Formar equipos que sean capaces de planear y registrar su trabajo, establecer metas bien definidas y sean aptos para realimentar su propio trabajo mediante la medición del mismo.

* Brindar un punto de vista a los gerentes y líderes de proyecto acerca de cómo monitorear y como motivar a sus equipos de trabajo para sacar el máximo potencial del mismo.

* Establecer una guía para el mejoramiento en organizaciones maduras; así como acelerar la mejora continua de procesos.”

El aplicar el TSP puede traer consigo muchos beneficios tanto personales como dentro de una empresa, ya que el buen trabajo en equipo ayuda a cumplir tanto los objetivos personales como en equipo que se planteen.

1.6.4. PMBOK. A Guide to the Project Management Body of Knowledge / Guía de los fundamentos para la dirección de proyectos.

La Guía de los Fundamentos para la Dirección de Proyectos de su acrónimo PMBOK (en inglés) es un libro que establece pautas, normas y estándares para la gestión de proyectos.

En palabras de Retos en Supply Chain, un blog de EAE Business School el PMBOK:

“...establece un criterio de buenas prácticas relacionadas con la gestión, la administración y la dirección de proyectos mediante la implementación de técnicas y herramientas que permiten identificar un conjunto de 47 procesos, distribuidos a su turno en 5 macroprocesos generales.”

Así mismo también Wikipedia, dice sobre el PMBOK que:

“La Guía PMBOK identifica el subconjunto de fundamentos de gestión de proyectos que es *"generalmente reconocido"* como una *"buena práctica"*. Con *"generalmente reconocido"* se trata de referir a los conocimientos y prácticas aplicables a la mayoría de los proyectos, la mayor parte del tiempo; en la que hay un consenso sobre su utilidad e importancia; mientras que *"buena práctica"* implica que hay un acuerdo general para la aplicación de los conocimientos, habilidades, herramientas y técnicas que pueden aumentar las posibilidades de éxito a lo largo de muchos proyectos.”

Con lo anterior expuesto se puede decir que el PMBOK es una herramienta desarrollada por el PMI (Project Management Institute) que proporciona normas, prácticas y estándares relacionados con la gestión operativa, administrativa y directiva utilizadas para el correcto desarrollo de un proyecto, esto se debe a que se recopilan aspectos clave que pueden aumentar considerablemente, la posibilidad de éxito en los proyectos si se aplican correctamente.

Sobre los macroprocesos generales del PMBOK, Omar García (2015) a través de la página projectum.lat, nos los define como:

“Los procesos de la dirección de proyectos se agrupan en 5 categorías conocidas como Grupos de Procesos de la Dirección de Proyectos, los cuáles se definen a continuación.

1. El grupo de procesos de iniciación. Aquellos procesos realizados para definir un nuevo proyecto o una nueva fase de un proyecto ya existente, mediante la obtención de la autorización para comenzar dicho proyecto o fase.
2. El grupo de procesos de planificación. Aquellos procesos requeridos para establecer el alcance del proyecto, refinar los objetivos y definir el curso de acción necesario para alcanzar los objetivos para cuyo logro se emprendió el proyecto.
3. El grupo de procesos de ejecución. Aquellos procesos realizados para completar el trabajo definido en el plan para la dirección del proyecto a fin de cumplir con las especificaciones del mismo.
4. El grupo de procesos de seguimiento y control. Aquellos procesos requeridos para dar seguimiento, analizar y regular el progreso y el desempeño del proyecto, para identificar áreas en las que el plan requiera cambios y para iniciar los cambios correspondientes.
5. El grupo de procesos de cierre. Aquellos procesos realizados para finalizar todas las actividades a través de todos los grupos de procesos, a fin de cerrar formalmente el proyecto o una fase del mismo. “

De acuerdo a lo anterior mostrado cada uno de este grupo de procesos muestra un objetivo, el procedimiento que se lleva a cabo y lo que se busca obtener o como se busca terminar el proceso. A continuación, se muestra un resumen sobre los 49 procesos del PMBOK® 6, basado en la información presentada por Oscar Josafat (2016) en la página web medium.com, así como en la web de todopmp.com, así como también otras fuentes consultadas de la web:

Los procesos del PMBOK® 6 ordenados:

1. (Inicio) Desarrollar el acta de constitución del proyecto: Tiene como objetivo formalizar el inicio del proyecto, describiendo las necesidades, objetivos, así como fijar las fechas importantes del proyecto, así mismo establece el nivel de autoridad que debe tener el director a lo largo del proyecto.
2. Identificar a los interesados: Es el primer paso que debe realizar el administrador de proyectos una vez que es nombrado. Los interesados serán aquellas organizaciones y/o empresas cuyos intereses se pueden ver comprometidos tanto de manera positiva

o negativa. La importancia de identificarlos es para aumentar el apoyo de todos aquellos cuyos intereses se ven afectados de manera positiva, así como disminuir el impacto a los que se ven afectados de manera negativa.

3. (Planeación) Desarrollar el plan para la dirección del proyecto: Planea las acciones para coordinar, definir e integrar todos los planes de las distintas áreas de conocimiento.
4. Planificar el involucramiento de los interesados: Busca mediante una estrategia poder involucrar a los interesados en el proyecto, así como suavizar los posibles impactos negativos analizando sus expectativas y requisitos.
5. Planificar la gestión del alcance: Describe cómo va a controlar, definir y validar el alcance del proyecto, para que este tenga todas las características y funciones requeridas.
6. Recopilar los requisitos: Se gestionan los requisitos de los interesados, estos requisitos incluyen necesidades, deseos, expectativas cuantificadas y alcance del producto.
7. Definir el alcance: Mediante una descripción detallada del proyecto y del producto se busca generar el enunciado del alcance del proyecto. Este describe de manera detallada los productos del proyecto y el trabajo que se necesita para crearlos.
8. Crear la EDT (Estructura de descomposición del trabajo): Se divide el proyecto de forma jerárquica para facilitar su planificación.
9. Planificar la gestión del cronograma: Se definen políticas que sirven para elaborar y gestionar el cronograma, sus cambios, contingencias, y el cómo se actualizará este.
10. Definir las actividades: Las actividades son el resultado de descomponer los paquetes de trabajo en componentes más pequeños y en estas se define el cronograma, costos, riesgos y recursos requeridos en el plan de dirección.

11. Secuenciar las actividades: Se determinan las dependencias entre las actividades del proyecto. Puede llevarse a cabo de forma manual o utilizar software especializado en gestión de proyectos.
12. Planificar la gestión de los riesgos: Proceso donde se define como realizar una identificación de riesgos y cómo actuar sobre ellos.
13. Identificar los riesgos: Proceso donde se determinarán todos los riesgos que puedan afectar al proyecto, así también se documentan sus características.
14. Realizar el análisis cualitativo de riesgos: Se incluyen métodos para priorizar los riesgos identificados, así como identificar otras acciones, ya sea el análisis cuantitativo de riesgos, o la planificación de la respuesta a los riesgos. Con el fin de mejorar el proyecto de manera efectiva centrándose en los riesgos de alta prioridad.
15. Realizar el análisis cuantitativo de riesgos: Se analiza el efecto de los riesgos asignando una cuantificación numérica, para tomar decisiones en base a ésta, en caso de incertidumbre.
16. Planificar la respuesta a los riesgos: Se desarrollan las opciones y acciones para mejorar las oportunidades y reducir las amenazas en el proyecto. Asignándose a un “Propietario de la respuesta de riesgos”, para que este asuma la responsabilidad de cada respuesta a los riesgos.
17. Planificar la gestión de recursos: Se identifican y documentan los roles del proyecto, las responsabilidades, habilidades necesarias y las relaciones de comunicación. También se crea un plan para gestionar al personal, donde a su vez se definen las relaciones jerárquicas, fechas de adquisición y liberación de este.
18. Planificar la gestión de los costos: Se establecen la documentación, políticas, y procedimientos necesarios para la planificación, gestión y ejecución del gasto; así como el control de los costos del proyecto.

19. Estimar los costos: Se desarrolla una estimación aproximada de los recursos monetarios necesarios para cumplir con las actividades del proyecto. Estos costos deben incluir: personal, materiales, equipamiento, servicios e instalaciones, entre otros. Se deben incluir categorías especiales como son: el factor de inflación, el costo de financiamiento y el costo de contingencias.
20. Estimar los recursos de las actividades: Se estiman los recursos de las actividades, identificándose el tipo, características y cantidad de los recursos necesarios para el cumplimiento de las actividades, todo esto para estimar el costo y la duración del proyecto de una manera precisa.
21. Estimar la duración de las actividades: Se identifican la cantidad de períodos de trabajo necesarios para cumplir y finalizar las actividades individuales con los recursos estimados. Se utiliza la estimación por tres valores, que consiste en estimar la duración de una actividad usando estimaciones optimista, más probable y pesimista.
22. Desarrollar el cronograma: Se integran y analizan las actividades, las secuencias de actividades, duraciones, requisitos de recursos y restricciones de cronograma para poder crear un modelo de programación del proyecto. Se suele desarrollar de las siguientes formas.
 1. Cuando se desarrolla el cronograma por primera vez no se incluyen los retrasos, adelantos, dependencias, asumiendo recursos ilimitados, esto con el propósito de determinar una fecha de finalización pesimista.
 2. Cuando se desarrolla el cronograma por segunda vez, ya se incluyen retrasos, adelantos, dependencias y recursos limitados, esto para determinar una fecha de finalización optimista o más probable.
23. Determinar el presupuesto: Se suman los costos estimados de todas las actividades individuales o de los paquetes de trabajo. Con el objetivo de establecer una línea base de costos autorizada con la que se podrá controlar y monitorear el proyecto.

24. Planificar la gestión de la calidad: Se identifican y documentan las métricas y estándares para que el proyecto pueda demostrar el cumplimiento de los mismos. La calidad debe estar planificada, diseñada e incorporada antes de que comience la ejecución del proyecto. Se implementan auditorías para la prevención de errores y defectos, llevándose a cabo de manera oportuna, periódica y puntual. En estas se debe de:

- Recomendar mejoras tanto en los procesos como en las políticas de calidad de la empresa.
- Establecer métricas que midan la calidad.
- Revisar la calidad del entregable antes de finalizarlo.
- Evaluar qué impacto tiene en la calidad los cambios de alcance, tiempo, costo, recursos y riesgos.
- Dedicar tiempo para realizar mejoras en el aspecto de la calidad.
- Asegurarse de utilizar el control integrado de cambios.

25. Planificar la gestión de las comunicaciones: Se determina cuáles serán los requisitos de información y necesidades de los interesados y activos de la organización disponibles. Tiene que garantizarse que la generación, recopilación, distribución, almacenamiento, recuperación y disposición final de la información sean oportunos y adecuados.

26. Planificar la gestión de las adquisiciones: Se determinan cuales son los bienes y servicios que deben adquirirse fuera de la organización y cuáles de estos pueden ser provistos internamente por el equipo del proyecto. Se analizan los contratos en particular para ver cuál es más conveniente, también se prepara toda la documentación necesaria para realizar los pedidos de propuestas.

27. (Ejecución) Dirigir y gestionar el trabajo del proyecto: Se lleva a cabo lo desarrollado en el plan para la dirección del proyecto. Se implementan todos los cambios aprobados que son las acciones correctivas, preventivas y reparación de defectos. También se revisa de manera periódica el impacto que tienen los cambios sobre el proyecto, y se realizan las actividades para cumplir con los objetivos del proyecto.
28. Gestionar el conocimiento del proyecto: Con el conocimiento existente se busca crear un nuevo conocimiento, con el fin de contribuir en el aprendizaje organizacional, así como también contribuir en el aprendizaje organizacional.
29. Gestionar la participación de los interesados: Se trabaja y comunica con los interesados con el fin de satisfacer las expectativas/necesidades, abordando los incidentes que han ocurrido y fomentar la participación. Las actividades que principalmente se realizan son:
1. Se involucra a los interesados con el fin de obtener y/o confirmar su compromiso.
 2. Mediante negociación y comunicación se gestionan las expectativas.
 3. Se tratan las inquietudes que pueden representar incidentes, y se tratan futuros problemas que puedan plantear.
 4. Se resuelven y aclaran incidentes ya identificados.
30. Adquirir recursos: Se negocia el obtener los mejores recursos posibles; se contratan nuevos trabajadores, ya sea internos y/o externos; se busca conocer las ventajas y desventajas de los equipos virtuales. El no adquirir los recursos humanos necesarios para el proyecto impacta en los presupuestos, cronogramas, calidad, riesgos y satisfacción al cliente.
31. Desarrollar el equipo: Se mejoran las capacidades grupales e individuales de los miembros del equipo. Por lo que el equipo debe mejorar en competencias, trabajo en equipo, cohesión, confianza e interrelaciones. Se debe motivar constantemente al

equipo proporcionándole oportunidades y desafíos, brindando apoyo y retroalimentación.

32. Dirigir al equipo: Consiste en dar seguimiento del desempeño de los integrantes del equipo, así como su retroalimentación y la resolución de polémicas o conflictos.
33. Gestionar las comunicaciones: Se crea, recopila, almacena, distribuye y recupera la información del proyecto de acuerdo con el plan de gestión de las comunicaciones. Con el fin de proporcionar un flujo de comunicación efectiva y eficiente entre los interesados y el equipo del proyecto.
34. Efectuar las adquisiciones: El equipo recibe ofertas y propuestas, para aplicar criterios de selección definidos y seleccionar uno o más vendedores que sean aceptables y están calificados para el trabajo. Se deben obtener las respuestas de los proveedores, seleccionarlos y generar los contratos. Se puede llegar a tener un solo proveedor para bajar costos, pero el riesgo para el proyecto aumenta en dado caso que el proveedor no cumpla. Así mismo cuando se trabaja con más de un proveedor para una misma tarea, los riesgos se diversifican y si uno de los proveedores no cumple se reemplaza rápidamente por otro que si este cumpliendo.
35. Gestionar la calidad: Proceso que consiste en verificar que se implementen correctamente todas las normas y procesos definidos en el plan de calidad. Se llevan a cabo auditorías para revisar los requisitos de calidad y los resultados obtenidos a partir de estas medidas. Esto llevado a cabo por un departamento de aseguramiento de calidad o una organización similar.
36. Implementar la respuesta a los riesgos: Se identifican los riesgos y se planifica una respuesta a los mismos. Garantiza que las respuestas acordadas sean llevadas a cabo para abordar la exposición al riesgo, minimizando las amenazas. Este proceso es importante contemplarlo con absoluto rigor. Este proceso está presente a lo largo de todo el proyecto.

37. (Monitoreo) Monitorear y controlar el trabajo del proyecto: En este proceso se da seguimiento, revisión e informes del avance del proyecto, conforme a los objetivos de desempeño definidos. Se compara el desempeño real con el que estaba planeado. Se identifican los nuevos riesgos, también se verifican e implementan los cambios formalmente aprobados.
38. Realizar el control integrado de cambios: Se analizan todas las solicitudes de cambio y aprobaciones. Se gestionan los cambios a los entregables y se informa las decisiones correspondientes.
39. Monitorear el involucramiento de los interesados: Se monitorean las relaciones de todos los interesados en el proyecto, y a su vez buscan estrategias para poder involucrar a los *stakeholders*. (“... (del inglés stakeholder) hace referencia a una persona, organización o empresa que tiene interés en una empresa u organización dada.”. De Wikipedia, La enciclopedia libre)
40. Controlar el cronograma: Detecta desviaciones en el cronograma para establecer acciones preventivas y/o correctivas con el objetivo de minimizar el riesgo en el proyecto.
41. Controlar los costos: Se detectan desviaciones y se llevan a cabo acciones correctivas para minimizar el riesgo del proyecto; analizando la relación entre el trabajo real y el uso de los fondos del proyecto, para efectuar cambios a estos gastos.
42. Monitorear las comunicaciones: Provee de un flujo óptimo de información entre los interesados y otros colaboradores con el equipo del proyecto, todo esto para que la comunicación sea de forma periódica a la audiencia adecuada en el momento adecuado.
43. Monitorear los riesgos: Se busca optimizar la eficacia de respuestas a todas las amenazas del proyecto. La herramienta más importante de este proceso es la auditoría de riesgos.

44. Controlar la calidad: Se asegura el cumplimiento de las normas establecidas, se verifica que los entregables del proyecto estén dentro del límite de calidad preestablecidos. También se pueden hacer cambios que sean necesarios para el proyecto.
45. Controlar los recursos: Se asegura que los recursos físicos asignados al proyecto estén disponibles tal y como fueron planeados, llevando a cabo un trabajo de monitoreo y una posible corrección si hay desviación del plan inicial. Este proceso se realizará durante toda la vida del proyecto.
46. Validar el alcance: Se formaliza la aceptación de los entregables revisándolos con el cliente y/o patrocinador para que se asegure que se ha completado satisfactoriamente.
47. Controlar el alcance: Se mide el grado de cumplimiento conseguido con respecto al alcance del proyecto, además supervisa cualquier solicitud de cambio que afecte a la línea base del alcance, para evitar la llamada “corrupción del alcance”
48. Controlar las adquisiciones: Se administran las relaciones y desempeño de los proveedores, verificando que se cumplan los términos y condiciones de los contratos estipulados. Se lleva a cabo por los administradores de contratos, ya que son los únicos que pueden rechazar o autorizar una solicitud para cambiar el contrato.
49. Cerrar el proyecto o fase: También se le llama cierre administrativo, y es en este paso donde se evalúan los resultados obtenidos para poder ya sea cerrar el proyecto o pasar a la siguiente fase.

Con estos 49 pasos podemos identificar las 5 etapas mencionadas con anterioridad, así como podemos deducir a cuál etapa corresponde cada paso. Retomando las etapas que son:

Inicio: Donde el propósito general es definir una nueva tarea, o una nueva fase de ejecución y con esto obtener los permisos necesarios para llevarlo a cabo.

Planificación: Se definen y concretan las metas, así como definir las estrategias más adecuadas para alcanzarlas.

Ejecución: Se llevan a cabo las actividades especificadas, de acuerdo con las estrategias adoptadas, para lograr las metas establecidas.

Gestión y control: Tienen como fin dar el seguimiento y evaluación del desempeño del proyecto.

Cierre: Se muestra la aceptación y satisfacción con los resultados obtenidos.

1.7. Metodologías para el control de calidad del software.

Mariana Pérez (2021), menciona lo siguiente sobre la definición de la palabra metodología:

“El término metodología se define como el grupo de mecanismos o procedimientos racionales, empleados para el logro de un objetivo, o serie de objetivos que dirige una investigación científica. Este término se encuentra vinculado directamente con la ciencia, sin embargo, la metodología puede presentarse en otras áreas como la educativa, en donde se encuentra la metodología didáctica o la jurídica en el derecho. Son muchos los contextos en donde la palabra metodología puede ser utilizada”

María Estela Raffino (2021) en la web de *concepto.de* menciona que:

“La metodología es la disciplina que estudia el conjunto de técnicas o métodos que se usan en las investigaciones científicas para alcanzar los objetivos planteados. Es una pieza fundamental para el estudio de las ciencias.”

La Wikipedia define metodología de la siguiente manera:

“La metodología (del griego μέθοδος de μετά μετά 'más allá, después, con', ὁδός odós 'camino' y λόγος logos 'razón, estudio'), hace referencia al conjunto de procedimientos racionales utilizados para alcanzar el objetivo o la gama de objetivos que rige una investigación científica, una exposición doctrinal o tareas que requieran habilidades, conocimientos o cuidados específicos. Con frecuencia puede definirse la metodología como el estudio o elección de un método pertinente o adecuadamente aplicable a determinado objeto.

No debe llamarse metodología a cualquier procedimiento, pues se trata de un concepto que en la gran mayoría de los casos resulta demasiado amplio, siendo preferible usar el vocablo método ...”

Con lo anterior se puede conceptualizar por metodología al conjunto de métodos y técnicas

que se llevan a cabo de forma sistemática para lograr un fin. Y esto es aplicable para el desarrollo y control de calidad del software. Actualmente son dos tipos de metodologías que abarcan el desarrollo y control de calidad del software, estas son la metodología tradicional, y la metodología ágil Y cada una de estas metodologías tienen modelos que los representan. En las siguientes páginas se hará mención detallada de ello.

1.7.1. Metodologías tradicionales

Ronny Demera (2021) en el sitio web de *tech.tribalyte.eu* menciona lo siguiente sobre las metodologías tradicionales:

“Las metodologías tradicionales, como su nombre nos indica, son las que se han usado toda la vida. Buscan imponer disciplina al proceso de desarrollo software y de esa forma volverlo predecible y por ello eficiente. De hecho, estas metodologías tienen un enfoque predictivo, donde se sigue un proceso secuencial en una sola dirección y sin marcha atrás. La estimación/captura de requisitos se realiza una única vez (exacto, una vez solo) al principio del proyecto y es precisamente por eso que nuestra estimación tendrá mucha importancia ya que de ella dependen todos los recursos que emplearemos en el proyecto. Si queremos adoptar una metodología tradicional, el desarrollo de un proyecto debe empezar siempre con un riguroso proceso de captura de requisitos, análisis y diseño.”

Por otra parte, Damorelos (2019) en la web de *scio.com.mx* menciona lo siguiente:

“La metodología tradicional también es conocida como desarrollo en cascada o waterfall en inglés. Este framework es de forma lineal, lo que significa que cada inicio de una etapa está condicionado a la finalización de la etapa anterior.”

Resumiendo lo anterior la metodología tradicional son aquellas metodologías que son lineales, buscan ser eficientes gracias a ser predecibles y automáticas.

1.7.2. Metodologías ágiles.

En palabras de la *Wikipedia* (2019) menciona lo siguiente:

“El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado mediante la colaboración de equipos autoorganizados y multidisciplinarios, inmersos en un proceso compartido de toma de decisiones a corto plazo. [...]”

Los métodos ágiles enfatizan las comunicaciones cara a cara en vez de la documentación. La mayoría de los equipos ágiles están localizados en una simple oficina abierta, a veces llamadas "plataformas de lanzamiento" (bullpen en inglés). La oficina debe incluir revisores, escritores de documentación y ayuda, diseñadores de iteración y directores de proyecto. Los métodos ágiles también enfatizan que el software funcional es la primera medida del progreso. Combinado con la preferencia por las comunicaciones cara a cara, generalmente los métodos ágiles son criticados y tratados como "indisciplinados" por la falta de documentación técnica."

Así mismo *Vanessa Rossello Villan (2019)* a través de la web *iebschool.com* menciona que:

"...las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.

En esencia, las empresas que apuestan por esta metodología consiguen gestionar sus proyectos de forma flexible, autónoma y eficaz reduciendo los costes e incrementando su productividad. Con este Curso en modelos organizativos ágiles 100% online aprenderás los aspectos básicos para gestionar las metodologías ágiles dentro de tu empresa."

Yanina Muradas (2018) a través de la web *openwebinars.net* menciona sobre metodologías ágiles lo siguiente:

"Podemos definir las metodologías ágiles como un conjunto tareas y procedimientos dirigidos a la gestión de proyectos. Son aquellos métodos de desarrollo en los cuales tanto las necesidades como las soluciones a estas evolucionan con el pasar del tiempo, a través del trabajo en equipo de grupos multidisciplinarios que se caracterizan por tener las siguientes cualidades:

- Desarrollo evolutivo y flexible.
- Autonomía de los equipos.
- Planificación.
- Comunicación."

La web de *redhat.com* (s.f.) menciona lo siguiente:

"El enfoque ágil para el desarrollo de software busca distribuir de forma permanente sistemas de software en funcionamiento diseñados con iteraciones rápidas.

Sin embargo, la frase "metodología ágil" es engañosa porque implica que el enfoque ágil es la única forma de abordar el desarrollo de software. La metodología ágil no hace referencia a una serie de indicaciones sobre qué hacer exactamente durante el desarrollo de software. Se trata más bien de una forma de pensar en la colaboración y los flujos de trabajo, y define un conjunto de valores que guían nuestras decisiones con respecto a lo que hacemos y a la manera en que lo hacemos.

En concreto, las metodologías ágiles de desarrollo de software buscan proporcionar en poco tiempo piezas pequeñas de sistemas de software en funcionamiento para mejorar la satisfacción del cliente. Estas metodologías utilizan enfoques flexibles y el trabajo en equipo para ofrecer mejoras constantes. Por lo general, el desarrollo ágil de software implica que pequeños equipos autoorganizados de desarrolladores de software y representantes empresariales se reúnan regularmente en persona durante el ciclo de vida del desarrollo de software. La metodología ágil

favorece un enfoque sencillo de la documentación de software, y acepta los cambios que puedan surgir en las diferentes etapas del ciclo de vida, en lugar de resistirse a ellos.”

Por otro lado, la web de *heflo.com* (s.f.) lo define de la siguiente forma:

“Metodología ágil es una de varias metodologías de desarrollo de software basadas en el desarrollo iterativo e incremental, en contraposición a las metodologías tradicionales de desarrollo de software lineal o cascada. Una metodología ágil proporciona un marco para apoyar el diseño, desarrollo y pruebas de soluciones de software a lo largo de su ciclo de vida.

Los métodos ágiles (p. Ej., Scrum) fomentan respuestas rápidas y flexibles al cambio mediante la promoción de la planificación adaptativa, la identificación de requisitos colaborativos y la racionalización entre el equipo interfuncional autoorganizado, así como el desarrollo gradual de soluciones. Muchos esfuerzos de desarrollo de software comerciales modernos siguen este tipo de enfoque.”

De acuerdo con lo anterior citado sobre las metodologías ágiles, podemos conceptualizar que estas consisten en un conjunto de métodos que aportan flexibilidad a la hora de desarrollar software, esto para lograr una mejor comunicación entre los miembros del equipo y lograr un mejor producto a menor tiempo y coste.

Estas metodologías ágiles, se basan en los valores y principios del documento del Manifiesto Ágil firmado por 17 expertos en programación: Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland y Dave Thomas redactado en el año 2001 con el fin de cambiar la forma de desarrollar software. Valores y principios los cuales retomando de *Wikipedia* (2020) son los siguientes:

Valores:

1. Individuos e interacciones sobre procesos y herramientas.
2. Software funcionando sobre documentación exhaustiva.
3. Colaboración con el cliente sobre negociación contractual.
4. Respuesta ante el cambio sobre seguir un plan.

Principios:

1. Satisfacer al cliente a través de la entrega temprana y continua de software con valor es una prioridad.
2. Aceptar que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan este cambio para proporcionar ventaja competitiva al

cliente.

3. Entregar software funcional frecuentemente, entre dos semanas y dos meses, teniendo preferencia al periodo de tiempo más corto posible.
4. Los responsables del negocio y los desarrolladores trabajan juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Se debe de dar el entorno y el apoyo que necesitan, así como confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es conversando cara a cara.
7. El software funcionando es la medida principal de progreso.
8. Los procesos ágiles promueven el desarrollo sostenido. Los promotores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
10. La simplicidad, o maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen cuando los equipos están auto-organizados.
12. A intervalos regulares, el equipo debe reflexionar sobre como ser más efectivo, para que ajuste y perfeccione su comportamiento en consecuencia.

Capítulo 2.

2.1 Calidad de un sistema de software.

Sabemos que la calidad en el software es diferente a la calidad que tiene un producto, actualmente un software es pieza clave en una empresa y las actividades que tiene, además de que mejora de forma significativa la gestión interna y externa por lo que hoy en día la mayoría de las empresas dependen del funcionamiento que tenga el software.

Por lo que existen distintos tipos en la calidad de software:

- **Calidad de diseño:** Son las características que los diseñadores dan para un componente, incluye el diseño del sistema, especificaciones y los requisitos.
- **Calidad de concordancia:** Es el rango de las características de diseño que se van aplicando mientras el software se encuentra en desarrollo.
- **Calidad técnica:** Está enfocada en el producto final que recibe un cliente.
- **Calidad funcional:** Se enfoca en cómo está siendo entregado el servicio durante el proceso. Si el cliente está de acuerdo con un modelo con ciertos requisitos, su base son las especificaciones, de igual forma se describe como la capacidad que se tiene para el uso de una parte del software o que es lo que la distingue de su competencia en el mercado.
- **Calidad estructural:** Es la forma que cumple con los requisitos no funcionales que son de apoyo a los requisitos funcionales, el mantenimiento o lo grande que es el software y la

forma en que se ha producido y si fue de manera correcta. Esta se evalúa por medio del análisis de la organización del código fuente, tecnología empleada y nivel de sistema.

2.1.1 Costo en la calidad de software.

El costo que se realizará al desarrollar un software es algo muy importante que se tiene al momento de hacer un análisis del proyecto a realizar, ya que este análisis de costo se definen la cantidad y el costo de los recursos como lo son conocimientos, tiempo y dinero.

Por lo que los estudios de costo de calidad se realizan para obtener una línea base y así determinar con exactitud si se puede reducir y lograr un presupuesto aceptable.

Los costos de calidad se pueden dividir en dos según *Valdéz (2014)*:

- “1) Costos asociados con prevención; estos costos incluyen la planificación de la calidad, revisiones técnicas formales, equipo de pruebas y entrenamiento.

- 2) Evaluación y fallas; estos costos incluyen actividades que permiten comprender mejor la condición del producto a través de cada proceso. Algunos ejemplos de costos de valuación incluyen ni inspección en el proceso y procesos, calibración y mantenimiento de equipo además de las pruebas correspondientes. Los costos de fallas son aquellos que desaparecerán si no hubiese defectos antes de enviar el producto a los clientes. Estos costos se subdividen en costos de fallas internas y externas.”

2.1.2 Sistema de control de calidad en el software

De acuerdo con la web *EcuRed (2013)* afirma lo siguiente sobre el llevar acabo un sistema de calidad dentro de una empresa:

“Las pruebas y las revisiones pueden variar dependiendo de la empresa, aquí es importante que se tengan documentadas y estén adaptados a los procesos de la compañía. Esto debe de estar en revisión a lo largo del ciclo de vida del software.

Existe una serie de actividades para el control de calidad que tendrá el software como lo son:

- Prueba del software
- Aplicación de una metodología
- Aplicación de las técnicas de desarrollo
- Ajuste a los estándares de implementación
- Control de información
- Administración de informes en el control de calidad

En cuanto a lo abordado con anterioridad, se llevan a cabo las pruebas y revisiones usadas a lo largo del desarrollo del software, para garantizar que cumple con las características o requisitos que se le han estipulado anteriormente y asegurar su funcionamiento.”

En cuanto a lo abordado con anterioridad, se llevan a cabo las pruebas y revisiones usadas a lo largo del desarrollo del software, para garantizar que cumple con las características o requisitos que se le han estipulado anteriormente y asegurar su funcionamiento.

2.1.3 Garantía de la calidad del software (SQA).

Como plantea la *Finding Passion for Quality (2015)* acerca de la SQA en la calidad que debe tener un producto, menciona lo siguiente:

“La SQA (Software Quality Assurance) es la supervisión que se aplica a lo largo del proceso y los métodos que aseguran su calidad, se realiza bajo las intervenciones de la gerencia de calidad la cual está llevando a cabo el desarrollo de software.

Objetivos de la SQA:

1. Planea las actividades para asegurar la calidad

2. Comprobar de manera objetiva que las actividades estén de acuerdo a los estándares de la SQA y procedimientos
3. Entrega los resultados de las auditorias informando a la gerencia.

La SQA (Software Quality Assurance) engloba:

- Un enfoque de gestión de calidad.
- Tecnología de Ingeniería de Software efectiva (métodos y herramientas).
- Revisiones técnicas formales que se aplican durante el proceso del software.
- Una estrategia de prueba multiescalada.
- Un control de la documentación del software y de los cambios realizados.
- Un procedimiento que asegure un ajuste a los estándares de desarrollo de software.
- Mecanismos de medición y de generación de informes.

Ventajas de usar la SQA:

- Disminuir los costos: Se realizan actividades de testeo que disminuyen las probabilidades de que exista alguna falla.
- Eficiencia: Se disminuyen los riesgos de extender el plazo establecido.
- Confianza: Ya que el software es testeado con la calidad requerida, esto garantiza que el sistema se esté desarrollando de forma segura.
- Previene conflicto: Las personas internas o externas no deberán tener problemas al momento de usarlo.
- Incrementa el nivel de satisfacción en el cliente.”

2.1.4 Metodología SQA

Ernesto (s.f.) señala lo siguiente:

“La SQA es una especialidad compleja y abundante en metodologías, por lo que es necesario la especialización de sus profesionales. De ahí, que el liderazgo de SQA deba ser asumido por uno o más ingenieros de calidad, lo que se conoce como grupo de SQA.

Tipos de pruebas:

- Validación: Esta prueba entran datos para despues realizar una comparación con los datos de salida, una vez realizada la comparación de los parámetros de un sistema anterior con los del nuevo sistema para determinar si es buena opción el sistema a desarrollar.
- Tensión: Esta prueba es realizada para llevar al software a su máxima capacidad en un periodo de tiempo y considerar si es adecuado a los niveles de carga.
- Utilidad: Con esta prueba se consigue que los usuarios que no conocen el nuevo software lo prueben por un tiempo y ofrezcan su opinión si la interfaz es entendible, puedan dar su opinión de cómo mejorar, al igual de los posibles bugs o errores que pueda tener y de esta forma repararlos en un futuro.”

Fases de desarrollo

En las fases de desarrollo, *Vargas (2010)* menciona que se pueden encontrar las siguientes:

“1.- Inicial. En esta fase los métodos aún no están establecidos. Por este modo, los proyectos se realizan de manera ilógica o incoherente y con muy poco profesionalismo. Según la unidad certificadora del CMM (Capability Maturity Model), el SEI (Software Engineering Institute) De los Estados Unidos de América, Existen muchos equipos de desarrollo de software en el mundo que operan con esta fase.

2.- Ejecutar: En esta fase se implementan algunos métodos de ingeniería en los proyectos.

3.- Definir: Los métodos, actividades y procesos que están vinculados con la ingeniería o la administración de proyectos se empiezan a estandarizar, documentar y ser construidos en base de un marco integrado para la compañía.

4.- Administración: La empresa comienza a operar bajo controles estadísticos de procesos.

5.- Optimiza: Las empresas comienzan a mejorar de manera continua, al igual que comienzan a enfocarse en técnicas de prevención de defectos, procesos y cambios de tecnología. Según el SEI (Software Engineering Institute) actualmente menos del 0.1% de las compañías alrededor del mundo se encuentran en esta fase de madurez.”

En la siguiente imagen se muestra un esquema sobre las fases del desarrollo en la ingeniería de software, con los puntos anteriormente mostrados.

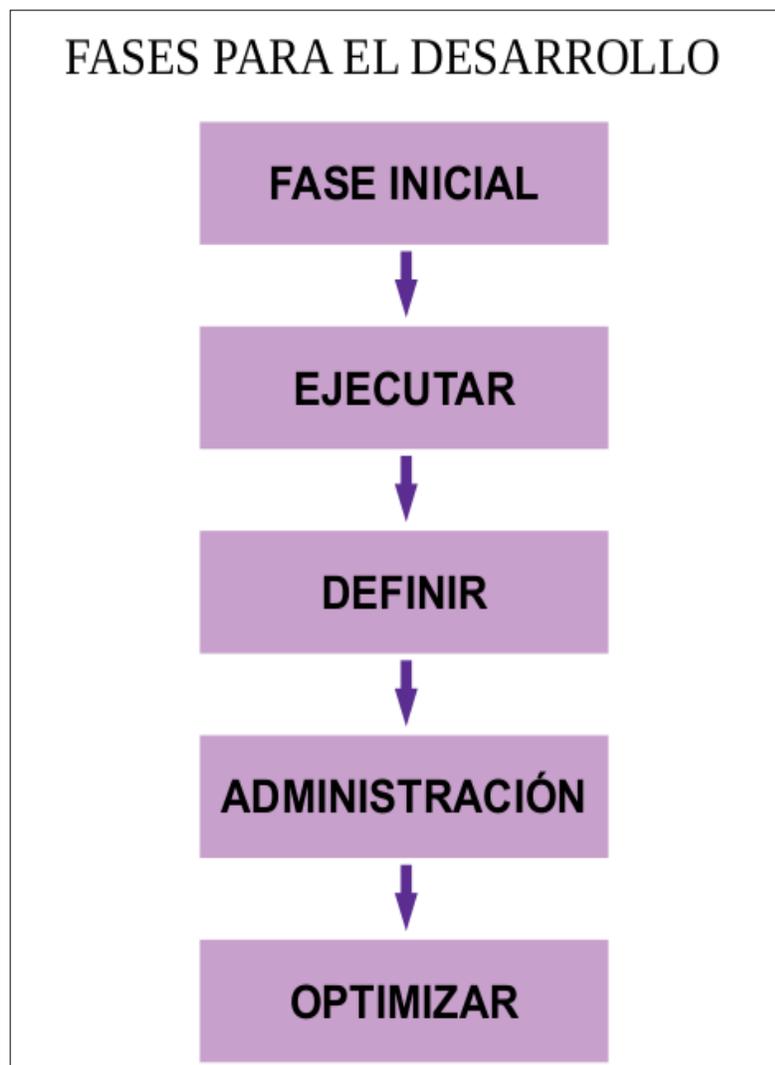


Imagen 2.1 Fases del desarrollo.

Imagen de autoría propia.

Capítulo 3.

Normas ISO en la calidad de un Software

3.1 ¿Qué son las ISO?

La ISO (International Organization for Standardization) es el organismo que se encarga de crear estándares para poder asegurar la calidad, al igual que la seguridad y eficiencia de un producto o servicio.

En la actualidad 164 países la tienen presente y existen más de 22,000 estándares de calidad que se rigen bajo estas normas en todas las industrias, como lo son las de tecnología, agricultura, salud, etc.

La ISO fue establecida en 1926 como la Federación Internacional de Asociaciones de Estandarización Nacionales, durante la segunda guerra mundial fue suspendida temporalmente, tiempo después las Naciones Unidas aconsejaron su refundación, por lo que en 1946, los representantes de 25 países estableció una reunión para la coordinación de estándares por medio de una unificación internacional, por lo que propusieron las bases de lo que hoy en día se terminará conociendo como ISO, fundada así oficialmente el 23 de febrero de 1947.

A principios del año 1980, la ISO nombró a una serie de comités técnicos para que comenzaran a desarrollar unas normas comunes que fueran aceptadas. El resultado de esto fue el compendio de las normas ISO 9000, se dio a conocer en 1986

3.1.1 Ventajas de las ISO

Ya que las normas ISO fueron creadas para orientar a las empresas con el objetivo de disminuir los costos y ser más eficientes, estas normas dan importantes beneficios como lo son:

3.1.2 Familias de las normas ISO

Las ISO se clasifican en series o familias con una nomenclatura en particular. Las categorías más importantes son

Serie ISO 9000: 2008

Se encarga de los estándares de calidad en los productos o servicios de las empresas, independientemente de su tamaño.

Serie ISO 14000: 2008

Son las encargadas de que las compañías organizan sus actividades dentro de unos estándares para respetar el medio ambiente, dando respuesta a la concientización y exigencia de la sociedad.

Serie ISO 27001: 2013

La web isotools.org, afirma lo siguiente acerca de esta ISO:

“ISO 27001 es una norma internacional que permite el aseguramiento, la confidencialidad e integridad de los datos y de la información, así como de los sistemas que la procesan.

El estándar ISO 27001:2013 para los Sistemas Gestión de la Seguridad de la Información (SGSI) permite a las organizaciones la evaluación del riesgo y la aplicación de los controles necesarios para mitigarlos o eliminarlos.

La aplicación de ISO-27001 significa una diferenciación respecto al resto, que mejora la competitividad y la imagen de una organización.

La Gestión de la Seguridad de la Información se complementa con las buenas prácticas o controles establecidos en la norma ISO 27002.

Estructura de la norma ISO 27001

1. Objeto y campo de aplicación: La norma comienza aportando unas orientaciones sobre el uso, finalidad y modo de aplicación de este estándar.

2. Referencias Normativas: Recomienda la consulta de ciertos documentos indispensables para la aplicación de ISO27001.

3. Términos y Definiciones: Describe la terminología aplicable a este estándar.

4. Contexto de la Organización: Este es el primer requisito de la norma, el cual recoge indicaciones sobre el conocimiento de la organización y su contexto, la comprensión de las necesidades y expectativas de las partes interesadas y la determinación del alcance del SGSI.

5. Liderazgo: Este apartado destaca la necesidad de que todos los empleados de la organización han de contribuir al establecimiento de la norma. Para ello la alta dirección ha de demostrar su liderazgo y compromiso, ha de elaborar una política de seguridad que conozca toda la organización y ha de asignar roles, responsabilidades y autoridades dentro de la misma.

6. Planificación: Esta es una sección que pone de manifiesto la importancia de la determinación de riesgos y oportunidades a la hora de planificar un Sistema de Gestión de Seguridad de la Información, así como de establecer objetivos de Seguridad de la Información y el modo de lograrlos.

7. Soporte: En esta cláusula la norma señala que para el buen funcionamiento del SGSI la organización debe contar con los recursos, competencias, conciencia, comunicación e información documentada pertinente en cada caso.

8. Operación: Para cumplir con los requisitos de Seguridad de la Información, esta parte de la norma indica que se debe planificar, implementar y controlar los procesos de la organización, hacer una valoración de los riesgos de la Seguridad de la Información y un tratamiento de ellos.

9. Evaluación del Desempeño: En este punto se establece la necesidad y forma de llevar a cabo el seguimiento, la medición, el análisis, la evaluación, la auditoría interna y la revisión por la dirección

del Sistema de Gestión de Seguridad de la Información, para asegurar que funciona según lo planificado.

10. Mejora: Por último, en la sección décima vamos a encontrar las obligaciones que tendrá una organización cuando encuentre una no conformidad y la importancia de mejorar continuamente la conveniencia, adecuación y eficacia del SGSI.”

Serie ISO 26000: 2008

De acuerdo a (Global STD Certification, 2019) se menciona lo siguiente:

“ISO 26000 es una norma internacional que ofrece una guía para integrar la responsabilidad social en todo tipo de organizaciones, con el fin de ayudar a las organizaciones a contribuir al desarrollo sostenible. No es una norma del sistema de gestión y no es certificable. La responsabilidad social está definida bajo esta norma como el compromiso de una organización ante los impactos que sus decisiones y actividades ocasionan en la sociedad y el medio ambiente mediante un comportamiento ético y transparente que contribuya al desarrollo sostenible incluyendo salud y bienestar de la sociedad.

La ISO 2600 nos establece 7 materias que son elementales para que la empresa u organización tenga que operar:

- Gobernanza organizacional
- Derechos humanos
- Prácticas laborales
- Medio ambiente
- Prácticas justas de operación
- Asuntos de consumidores
- Participación activa y desarrollo de la comunidad

ISO 26000 es una gran herramienta para las empresas, las ayuda a pasar de las buenas intenciones a las buenas acciones, y entre sus beneficios se destacan: mayor competitividad, mejor reputación

a nivel social, atraer y retener colaboradores, clientes y usuarios, mantener la motivación, compromiso y productividad.

Todos formamos parte de esta cadena de responsabilidad social, dueños, directivos, colaboradores y sociedad; por lo que el cuidado de nuestra comunidad y medio ambiente depende de todos. Las políticas y prácticas serán entonces, la carta de presentación ante las nuevas generaciones que verán con buenos ojos las acciones que se implementan.”

En conclusión, a lo anterior expuesto, podemos afirmar que esta familia ayuda a que la empresa tenga un comportamiento ético como parte de su administración.

3.2 ISO 9004-2: 2008

“En esta norma según Portillo (2015) en su libro *Gestión de la calidad de productos editoriales multimedia. ARGN0110* nos afirma que:

Es un documento en el que especifica cómo tiene que comportarse el software en los soportes de los usuarios, en dicho documento se recogen los siguientes criterios:

- Responsabilidad de gestión.
- Examinar, medir y pruebas.
- Sistema de calidad.
- Inspección y estado de pruebas.
- Revisión de contrato.
- Acción correctiva.
- Control de diseño.
- Control del producto no aceptado por las directrices de calidad.
- Control documental.
- Tratamiento, almacenamiento, packaging y entrega del producto.

- Compra del mismo.
- Producto que se proporciona al comprador.
- Registros de calidad.
- Estadísticas.”

3.2.1. ISO 9000-3: 2008

Se dedica al proceso de desarrollo con la calidad del software. Su finalidad es proveer las especificaciones a la implementación y el mantenimiento del desarrollo del software.

Sus beneficios son los siguientes:

- Se obtiene una documentación de los sistemas.
- Aumenta la productividad y la eficiencia del software.
- Satisfacción al cliente.
- Reducción de auditorías de los clientes en cuestión de calidad
- Optimiza el tiempo de desarrollo.

Algunas de las secciones más importantes de la ISO 9000-3 son:

• **Control de documentos y datos:** Se implementarán los métodos para control de los datos y los documentos, al igual que la documentación obsoleta deberá ser eliminada.

• **Inspección interna de la calidad:** Se debe llevar a cabo un sistema de auditorías y sus conclusiones deben estar documentadas.

• **Gestión de la dirección:** La empresa deberá implementar sus objetivos de acuerdo a la calidad. La dirección de todo el personal deberá ser claramente definida.

- **Control de calidad:** Se deberá tener documentación que incluye:
 - (1) La revisión de los contratos del personal,
 - (2) Instrucciones para los métodos del sistema de calidad
 - (3) Cómo llevar a cabo la aplicación de estos métodos.

- **Control de producto no conforme:** Los productos que no cumplan con las especificaciones no serán instalados o desarrollados.

- **Métodos estadísticos:** Se deberá tener documentadas las estadísticas que lleven a cabo los estados de las actividades.

- **Capacitación:** La empresa debe implementar los procesos de capacitación para el personal y ser certificado con base en el área.

3.2.2 ISO 12207

Gandarillas (2017) Plantea lo siguiente acerca de la norma ISO 12207:

“Es el estándar para los procesos de ciclo de vida del software de la organización ISO. Los procesos se clasifican en tres tipos: procesos principales, procesos de soporte y procesos de la organización. Los procesos de soporte y de organización deben existir independientemente de la organización y del proyecto ejecutado. Los procesos principales se instancian de acuerdo con la situación particular.

Procesos principales:

- **Adquisición:** En este proceso son definidas las tareas a realizar donde, la organización adquiere un sistema, producto de software o servicio de software.

- **Suministro:** Se definen las actividades que tendrá el proveedor, organización que entrega un sistema o servicio de software al cliente.

- **Desarrollo:** Engloba las actividades que realizará el desarrollador del software.

- **Operación:** Contiene las distintas actividades a realizar del operador en la organización.

- Mantenimiento: Abarca las tareas de la persona que será responsable del mantenimiento, de esta forma el producto se mantendrá actualizado y en operación.

Procesos de soporte.

- Proceso de documentación: Es el registro de indicaciones del ciclo de vida que tendrá el software.
- Proceso de administración de la configuración: Engloba a la administración de la estructuración del software.
- Proceso de aseguramiento de la calidad: Se encarga de asegurar que los procesos del software se ajusten a los planes establecidos.
- Proceso de verificación: Se verifica que la calidad del proyecto sea la adecuada.
- Proceso de validación: Se hacen válidos los procesos del software.
- Proceso de revisión conjunta: Engloba a las tareas para evaluar el producto.
- Proceso de auditoría: Contiene las fechas en que será auditado el software.
- Proceso de resolución de problemas: Se definen y analizan los posibles problemas que podría tener el sistema.”

En definitiva, encontramos que en la actualidad los softwares son uno de los mayores ingresos económicos alrededor del mundo, ya que se ofrece una gran oportunidad y diversas fuentes de negocio para países que están en desarrollo, por lo que el software se debe estar actualizando constantemente. Es por ello que fue creada la Norma ISO 12207, esta norma es la encargada de definir el ciclo de vida que tendrá el software, desde los procesos, las actividades que realizará y las configuraciones que tendrá.

El principal objetivo de esta norma es dar una base para la gente involucrada en el desarrollo haga uso de un lenguaje común

A continuación, se muestra un esquema sobre la ISO 12207.



*Imagen 3.1 ISO 12207
Imagen de autoría propia.*

3.2.3 ISO/IEC 9126

Sobre la ISO 9126 Medina Prieto (2017) indica que:

“Es un estándar internacional para la evaluación de la calidad del software. Está reemplazado por el proyecto SQuaRE, ISO 25000:2005, el cual sigue los mismos conceptos

El estándar está dividido en cuatro partes las cuales dirigen, realidad, métricas externas, métricas internas y calidad en las métricas de uso y expendido. El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y subcaracterísticas como demuestra en la imagen:



Imagen 3.2 Modelo de ISO 9126 (ISOTools Excellence, 2015).

- Funcionalidad - Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas.

Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas.

- Adecuación - Atributos del software relacionados con la presencia y aptitud de un conjunto de funciones para tareas especificadas.
- Exactitud - Atributos del software relacionados con la disposición de resultados o efectos correctos o acordados.
- Interoperabilidad - Atributos del software que se relacionan con su habilidad para la interacción con sistemas especificados.
- Seguridad - Atributos del software relacionados con su habilidad para prevenir acceso no autorizado ya sea accidental o deliberado, a programas y datos.
- Cumplimiento funcional.
- Fiabilidad - Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido.
- Madurez - Atributos del software que se relacionan con la frecuencia de falla por fallas en el software.
- Recuperabilidad - Atributos del software que se relacionan con la capacidad para restablecer su nivel de desempeño y recuperar los datos directamente afectados en caso de falla y en el tiempo y esfuerzo relacionado para ello.
- Tolerancia a fallos - Atributos del software que se relacionan con su habilidad para mantener un nivel especificado de desempeño en casos de fallas de software o de una infracción a su interfaz especificada.
- Cumplimiento de Fiabilidad - La capacidad del producto software para adherirse a normas, convenciones o legislación relacionadas con la fiabilidad.
- Usabilidad - Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.
- Aprendizaje- Atributos del software que se relacionan al esfuerzo de los usuarios para reconocer el concepto lógico y sus aplicaciones.
- Comprensión - Atributos del software que se relacionan al esfuerzo de los usuarios para reconocer el concepto lógico y sus aplicaciones.

- Operatividad - Atributos del software que se relacionan con el esfuerzo de los usuarios para la operación y control del software.
- Atractividad
- Eficiencia - Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas.
- Comportamiento en el tiempo - Atributos del software que se relacionan con los tiempos de respuesta y procesamiento y en las tasas de rendimientos en desempeñar su función.
- Comportamiento de recursos - Usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función bajo condiciones determinadas.
- Mantenibilidad - Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.
- Estabilidad - Atributos del software relacionados con el riesgo de efectos inesperados por modificaciones.
- Facilidad de análisis - Atributos del software relacionados con el esfuerzo necesario para el diagnóstico de deficiencias o causas de fallos, o identificaciones de partes a modificar.
- Facilidad de cambio - Atributos del software relacionados con el esfuerzo necesario para la modificación, corrección de falla, o cambio de ambiente.
- Facilidad de pruebas - Atributos del software relacionados con el esfuerzo necesario para validar el software modificado.
- Portabilidad - Conjunto de atributos relacionados con la capacidad de un sistema de software para ser transferido y adaptado desde una plataforma a otra.
- Capacidad de instalación - Atributos del software relacionados con el esfuerzo necesario para instalar el software en un ambiente especificado.
- Capacidad de reemplazamiento - Atributos del software relacionados con la oportunidad y esfuerzo de usar el software en lugar de otro software especificado en el ambiente de dicho software especificado.

- Calidad en uso - Conjunto de atributos relacionados con la aceptación por parte del usuario final y Seguridad.
- Eficacia - Atributos relacionados con la eficacia del software cuando el usuario final realiza los procesos.
- Productividad - Atributos relacionados con el rendimiento en las tareas cotidianas realizadas por el usuario final.
- Seguridad - Atributos para medir los niveles de riesgo.
- Satisfacción - Atributos relacionados con la satisfacción de uso del software.

La subcaracterística Conformidad no está listada arriba ya que se aplica a todas las características. Ejemplos son conformidad a la legislación referente a usabilidad y fiabilidad. Cada subcaracterística (como adaptabilidad) está dividida en atributos. Un atributo es una entidad la cual puede ser verificada o medida en el producto software. Los atributos no están definidos en el estándar, ya que varían entre diferentes productos software.”

De acuerdo a lo anterior, es posible decir que la ISO 9126 fue creada para evaluar la calidad que tendrá un software, especificando las características que este pudiera tener.

Capítulo 4.

Desarrollo de la página web.

En este capítulo consideramos que es necesario e importante dar una introducción a lo que son las herramientas necesarias para desarrollar la página web, todo esto con la finalidad de que se pueda comprender para qué sirve cada programa y lenguaje empleado; así como también consideramos necesario explicar palabras clave que se utilizarán a lo largo del desarrollo de la página web.

4.1. Concepto de “Pagina Web”.

Hay varios conceptos que es necesario saber para comprender que es una página web; *María Estela Raffino (2020)* en la web de *concepto.de* menciona lo siguiente:

“Se conoce como página Web, página electrónica o página digital a un documento digital de carácter multimediático (es decir, capaz de incluir audio, video, texto y sus combinaciones), adaptado a los estándares de la World Wide Web (WWW) y a la que se puede acceder a través de un navegador Web y una conexión activa a Internet. Se trata del formato básico de contenidos en la red.”

Así mismo la web *conceptodefinicion.de (2019)* menciona lo siguiente:

“Una Página Web es conocida como un documento de tipo electrónico, el cual contiene información digital, la cual puede venir dada por datos visuales y/o sonoros, o una mezcla de ambos, a través de textos, imágenes, gráficos, audio o vídeos y otros tantos materiales dinámicos o estáticos. Toda esta información se ha configurado para adaptarse a la red informática mundial, también conocida como World Wide Web.”

Mientras que *Julián Pérez Porto y María Merino (2009)* en la web de *Definicion.de* dice:

“Se conoce como página web al documento que forma parte de un sitio web y que suele contar con enlaces (también conocidos como hipervínculos o links) para facilitar la navegación entre los contenidos. [...]

Las páginas web están desarrolladas con lenguajes de marcado como el HTML, que pueden ser interpretados por los navegadores. De esta forma, las páginas pueden presentar información en distintos formatos (texto, imágenes, sonidos, videos, animaciones), estar asociadas a datos de estilo o contar con aplicaciones interactivas.

Entre las múltiples características que tiene una página web y que sirven para identificarla se encuentran las siguientes: cuenta con información textual y también con material de tipo audiovisual, está dotada de un diseño atractivo, está optimizada y ejerce como la tarjeta de presentación de una empresa, una persona o un profesional concreto.”

Así mismo la *Wikipedia 2020* dice que:

“Una página web, o página electrónica, página digital, o ciber página es un documento o información electrónica capaz de contener texto, sonido, vídeo, programas, enlaces, imágenes y muchas otras cosas, adaptada para la llamada World Wide Web (WWW) y que puede ser accedida mediante un navegador web. Esta información se encuentra generalmente en formato HTML o XHTML, y puede proporcionar acceso a otras páginas web mediante enlaces de hipertexto. Frecuentemente también incluyen otros recursos como pueden ser hojas de estilo en cascada, scripts, imágenes digitales, entre otros.

Las páginas web pueden estar almacenadas en un equipo local o en un servidor web remoto. El servidor web puede restringir el acceso únicamente a redes privadas, por ejemplo, en una intranet corporativa, o puede publicar las páginas en la World Wide Web. El acceso a las páginas web es realizado mediante una transferencia desde servidores, utilizando el protocolo de transferencia de hipertexto (HTTP).”

Con respecto a lo anterior se puede decir que una página web es un documento digital el cual puede contener texto, imágenes, audios, video, links, entre otras cosas. Y es utilizada con distintos fines, ya sea desde el entretenimiento hasta ser una tarjeta de presentación de servicios, entre otros fines.

4.2. Software utilizado.

Para la realización de este proyecto se utilizó distinto software del cual se expondrá a continuación una descripción del mismo

4.2.1. HTML

Sobre que es HTML la web de *developer.mozilla.org (2020)* dice que:

“HTML, que significa Lenguaje de Marcado de Hipertextos (HyperText Markup Language), es la pieza más básica para la construcción de la web y se usa para definir el sentido y estructura del contenido en una página web. Otras tecnologías además de HTML son usadas generalmente para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript).

"Hipertexto" se refiere a los enlaces que conectan las páginas web entre sí, ya sea dentro de un mismo sitio web o entre diferentes sitios web. los vínculos son un aspecto fundamental de la web. Al subir contenido a Internet y vincularlo a páginas creadas por otras personas, te haces participante activo en la red mundial (World Wide Web)."

La web de *desarrolloweb.com (s.f.)* menciona lo siguiente:

“HTML es un lenguaje de marcación que sirve para definir el contenido de las páginas web. Se compone en base a etiquetas, también llamadas marcas o tags, con las cuales conseguimos expresar las partes de un documento, cabecera, cuerpo, encabezados, párrafos, etc. En definitiva, el contenido de una página web.”

María Jesús Lamarca Lapuente en la web *hipertexto.info (s.f.)* menciona lo siguiente:

“El lenguaje de marcas de hipertexto, HTML o (HyperText Markup Language) se basa en el metalenguaje SGML (Standard Generalized Markup Language) y es el formato de los documentos de la World Wide Web. El World Wide Web Consortium (W3C) es la organización que desarrolla los estándares para normalizar el desarrollo y la expansión de la Web y la que publica las especificaciones relativas al lenguaje HTML.

HTML fue concebido como un lenguaje para el intercambio de documentos científicos y técnicos adaptado para su uso por no especialistas en tratamiento de documentos. HTML resolvió el problema de la complejidad de SGML sirviéndose de un reducido conjunto de etiquetas estructurales y semánticas apropiadas para la realización de documentos relativamente simples. Pero, además de simplificar la estructura de los documentos, HTML soportaba el hipertexto.”

Julián Pérez Porto y Ana Gardey en la web de definicion.de (2008) mencionan que:

“HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto.”

Así mismo *Juan José Pino Reyes (s.f.) en la web de devcode.la afirma que:*

“HTML no es un lenguaje de programación, esto debes tenerlo muy en claro desde el principio, HTML es un lenguaje de marcado de hipertexto o “HyperText Markup Language” por el desarrollo de sus iniciales en inglés, básicamente este lenguaje se escribe en su totalidad con elementos, estos elementos están constituidos por etiquetas, contenido y atributos [...]”

Conceptualizando lo anterior expuesto podemos decir que HTML del HyperText Markup Language o Lenguaje de Marcas de Hipertexto es un lenguaje de marcado y no de programación, que se utiliza para construir la estructura principal de una página web.

4.2.2. CSS

La *Wikipedia (2020)* con respecto a lo que es CSS menciona que:

“CSS (siglas en inglés de Cascading Style Sheets), en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. Te puede ayudar a crear tu propio sitio web. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles (como Firefox OS).

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.”

La web de *Arume (2019)* menciona lo siguiente:

“El CSS (hojas de estilo en cascada) es un lenguaje que define la apariencia de un documento escrito en un lenguaje de marcado (por ejemplo, HTML).

Así, a los elementos de la página web creados con HTML se les dará la apariencia que se desee utilizando CSS: colores, espacios entre elementos, tipos de letra, ... separando de esta forma la estructura de la presentación.

Esta separación entre la estructura y la presentación es muy importante, ya que permite que sólo cambiando los CSS se modifique completamente el aspecto de una página web. Esto posibilita, entre otras cosas, que los usuarios puedan usar hojas de estilo personalizadas (como hojas de estilo de alto contraste o de accesibilidad).”

Asimismo, la web de *desarrolloweb.com (s.f.)* menciona sobre CSS lo siguiente:

“CSS es el segundo lenguaje más básico y esencial para crear páginas web. El primero sería HTML, con el que se define el contenido de la página. El segundo CSS, con el que se define la parte de la presentación, es decir, cómo deben mostrarse los elementos de la página, su posición, forma, espaciados, colores y, en resumen, toda la parte estética.

CSS es un lenguaje que consiste en una serie de elementos mediante los cuales se declaran los estilos, básicamente éstos son los más importantes:

- Selectores, mediante los cuales podemos especificar qué elementos de la página nos estamos refiriendo
- Atributos de estilo para definir qué cosas queremos estilizar sobre los selectores indicados

- Una serie de valores, que indican qué estilo se debe aplicar a cada atributo sobre cada selector. Los valores se expresan con unidades CSS, que sirven para cuantificar los valores (píxeles, puntos...)”

Por lo que podemos afirmar con lo anterior citado que CSS es un lenguaje de diseño utilizado para dar apariencia y estilo a las páginas web de acuerdo a los gustos del usuario o del cliente.

4.2.3. PHP

La web *php.net* (s.f.) define PHP de la siguiente forma:

“PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. “

Wikipedia (2021) menciona sobre PHP lo siguiente:

“PHP es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web. Fue creado inicialmente por el programador danés-canadiense Rasmus Lerdorf en 1994. En la actualidad, la implementación de referencia de PHP es producida por The PHP Group. PHP originalmente significaba Personal Home Page (Página personal), pero ahora significa el inicialismo recursivo PHP: Hypertext Preprocessor.

El código PHP suele ser procesado en un servidor web por un intérprete PHP implementado como un módulo, un Daemon o como un ejecutable de interfaz de entrada común (CGI). En un servidor web, el resultado del código PHP interpretado y ejecutado —que puede ser cualquier tipo de datos, como el HTML generado o datos de imágenes binarias— formaría la totalidad o parte de una respuesta HTTP. Existen diversos sistemas de plantillas, sistemas de gestión de contenidos y frameworks que pueden emplearse para organizar o facilitar la generación de esa respuesta. Por otra parte, PHP puede utilizarse para muchas tareas de programación fuera del contexto de la web, como aplicaciones gráficas autónomas y el control de drones. También se puede interpretar y ejecutar un código PHP cualquiera a través de una interfaz de línea de comandos (CLI).

El intérprete estándar de PHP, impulsado por Motor Zend, es un software libre publicado bajo Licencia PHP. PHP ha sido ampliamente portado y puede ser desplegado en la mayoría de los servidores web en casi todos los sistemas operativos y plataformas, de forma gratuita.”

Por otro lado, la definición que la web de *aprenderaprogramar.com* (s.f.) da es la siguiente:

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas.

PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc.

Por lo que podemos conceptualizar a PHP como un lenguaje de programación que se utiliza ampliamente en el desarrollo de páginas web dinámicas (páginas cuyo contenido está en constante cambio), siendo uno de los lenguajes más utilizados para el desarrollo web.

4.2.4. Visual Studio Code

La web de *mclibre.org* (2020) menciona lo siguiente:

Visual Studio Code es un editor de programación multiplataforma desarrollado por Microsoft. Es un proyecto de software libre que se distribuye bajo la licencia MIT, aunque los ejecutables se distribuyen bajo una licencia gratuita no libre.

La Wikipedia dice que:

“Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias.

Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

Visual Studio Code se basa en Electron, un framework que se utiliza para implementar Chromium y Node.js como aplicaciones para escritorio, que se ejecuta en el motor de diseño Blink. Aunque utiliza el framework Electron, el software no usa Atom y en su lugar emplea el mismo componente editor (Monaco) utilizado en Visual Studio Team Services (anteriormente llamado Visual Studio Online)."

Resumiendo lo anteriormente citado podemos decir que VSC o Visual Studio Code es un editor de código que puede ser utilizado para diferentes lenguajes de programación o de diseño (HTML & CSS).

4.2.5. JavaScript

JavaScript (JS) es un lenguaje simple, traducción secuencial o lenguaje de programación con funciones de primer nivel. Mejor conocido como scripting de sitio web, se utiliza en muchos entornos que no son de navegación, como Adobe Acrobat. Apache Node.js y CouchDB. JS es un lenguaje de programación basado en productos de un solo subproceso, multiproceso que admite la optimización, implementación y programación de la visualización de objetos (como programas de trabajo). Esta sección es específica del lenguaje JavaScript en sí y no está destinada a ninguna página web u otro entorno de host en particular. Para conocer las API específicas del sitio web, consulte Web DOM y API. El JavaScript estándar es ECMAScript. Desde el 2012, todos los navegadores modernos admiten ECMAScript 5.1. Los investigadores de mayor edad apoyan al menos ECMAScript 3. El 17 de junio de 2015, ECMA International lanzó la sexta versión principal de ECMAScript. Oficialmente llamado ECMAScript 2015 y originalmente llamado ECMAScript 6 o ES6. Desde entonces, la versión ECMAScript está en una versión anual. Este documento hace referencia a la nueva plantilla de vista previa que actualmente es ECMAScript 2020. No confunda JavaScript con el lenguaje de programación Java. Tanto "Java" como "JavaScript" lenguajes de programación registrados por Oracle en los Estados Unidos y otros países. Sin embargo, los dos lenguajes de programación difieren en estructura, significado y facilidad de uso.

4.2.5.1. Librería JQuery

Wikipedia menciona al respecto sobre JQuery:

“jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. De acuerdo a un análisis de la Web (realizado en 2017) JQuery es la biblioteca de JavaScript más utilizada, por un amplio margen.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.”

Laura Chuburu (s.f.) en su web de *laurachuburu.com.ar* menciona lo siguiente sobre JQuery:

“jQuery es una librería de JavaScript (JavaScript es un lenguaje de programación muy usado en desarrollo web). Esta librería de código abierto, simplifica la tarea de programar en JavaScript y permite agregar interactividad a un sitio web sin tener conocimientos del lenguaje.

Basados en esta librería, existe una infinita cantidad de plugins (gratis y pagos) creados por desarrolladores de todo el mundo. Estos plugins resuelven situaciones concretas dentro del maquetado de un sitio, por ejemplo: un menú responsive, una galería de fotos, un carrusel de imágenes, un slider, un header que cambia de tamaño, el desplazamiento del scroll al hacer clic en un botón (anclas HTML), la transición entre páginas y miles de efectos más.”

Con lo anterior mencionado podemos afirmar que JQuery es una librería perteneciente a JavaScript, que tiene como objetivo simplificar las tareas de programación con JavaScript, para que estas sean más entendibles para usuarios que no son expertos en JavaScript o están iniciando en esto.

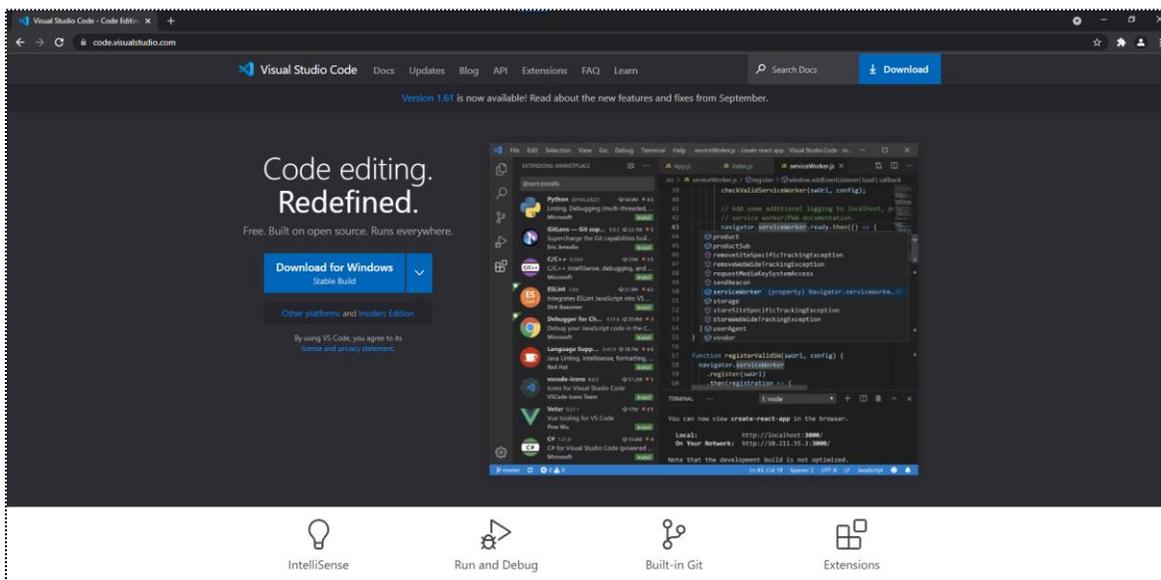
4.3. Instalación del software

En esta sección se mostrarán los procedimientos de instalación del software necesario tanto en el sistema operativo Windows 10, así como en Ubuntu Budgie en su versión 18.4 LTS

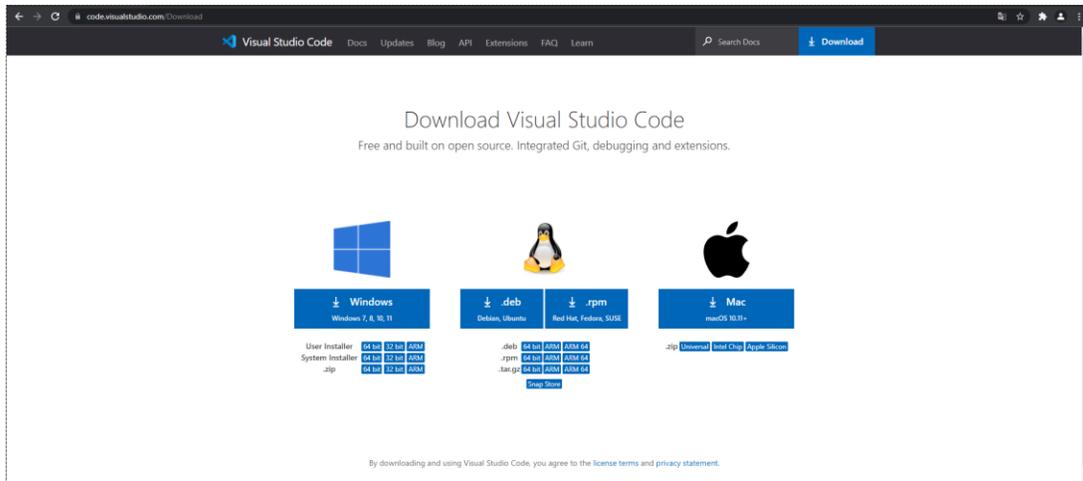
4.3.1. Instalación del software en Windows 10

Empezaremos mostrando la instalación de Visual Studio Code.

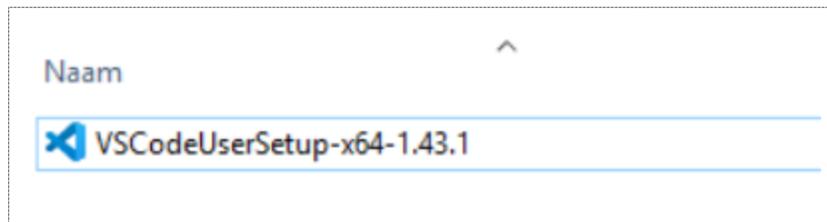
Paso 1: Ingresa a la página <https://code.visualstudio.com> y haz clic en el botón 'Download' que se muestra del lado superior derecho de la página.



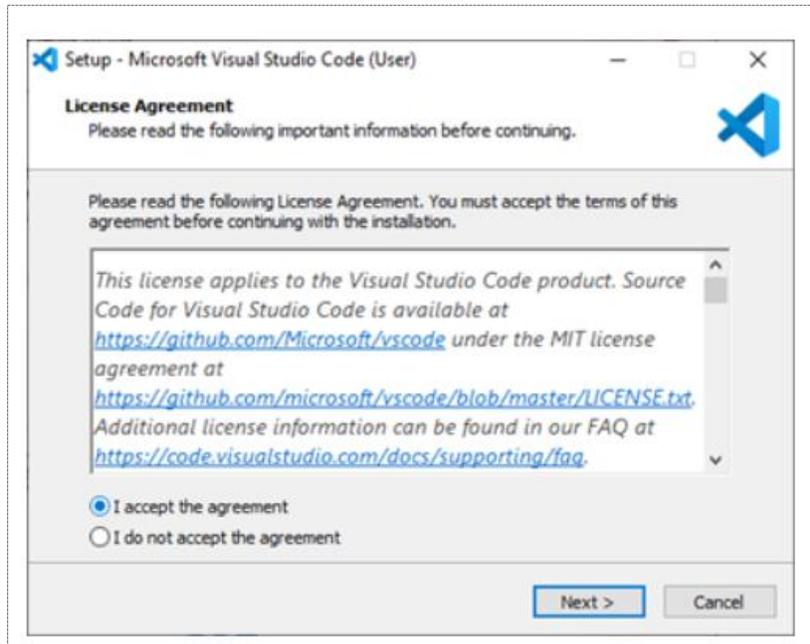
Dirigirá a la siguiente página, como se muestra a continuación. Seleccionamos el sistema operativo de Windows, que es con el que trabajaremos, una vez elegido comenzará la descarga.



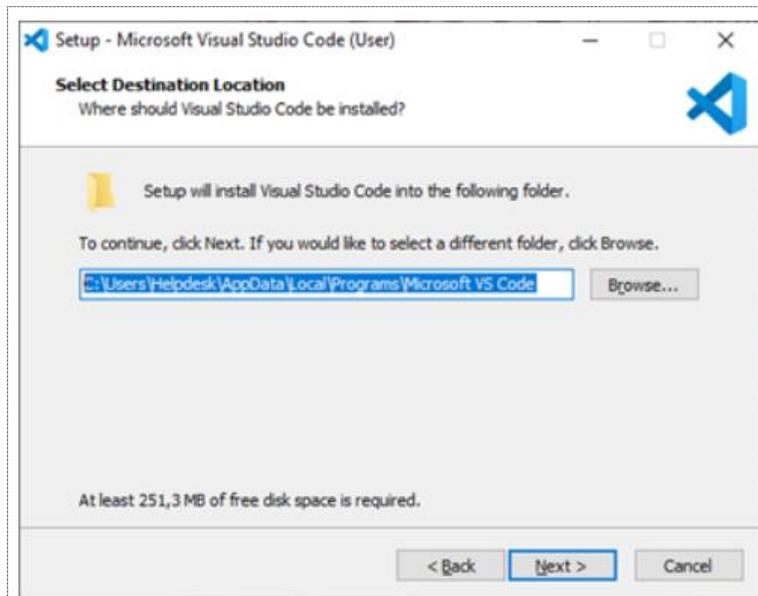
Paso 2: Una vez terminada la descarga, deberemos ir a la ruta en donde nuestro archivo fue almacenado, mostrando un ejecutable similar al de la imagen inferior; ejecutaremos el archivo para iniciar la instalación.



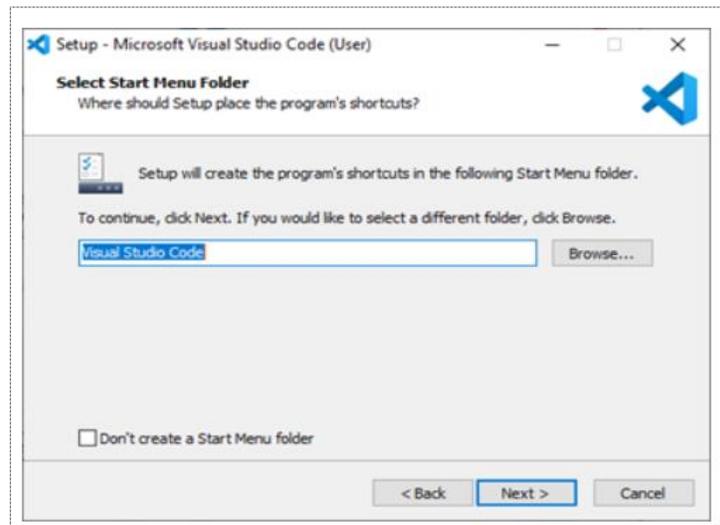
Paso 3: Mostrara un acuerdo de licencia, deberemos aceptarlo para continuar. Una vez marcada la casilla en donde aceptamos términos y condiciones, procedemos a hacer clic en Next para continuar con la instalación del programa. Justo como se muestra en la siguiente imagen.



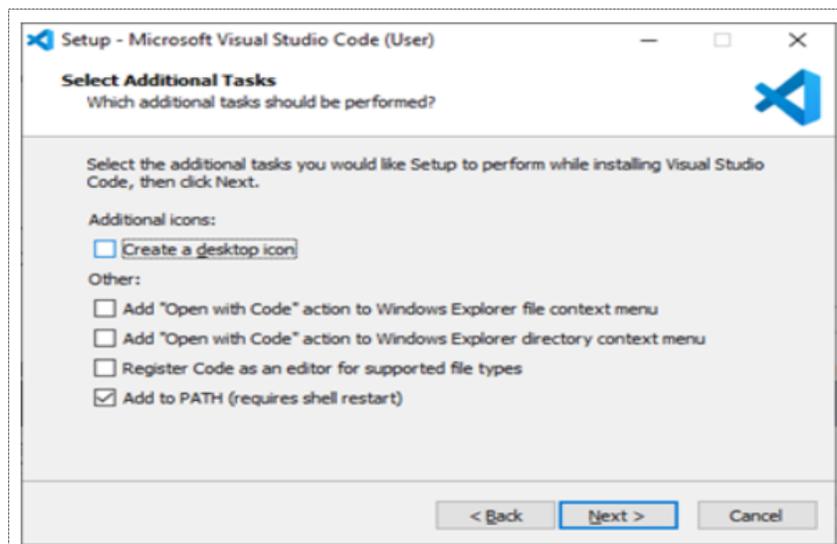
Paso 4: Se mostrará una ventana como la de la siguiente imagen en donde podremos seleccionar la ruta en nuestra computadora en la cual el programa se instalará. Utilizaremos la ruta que por defecto marca el instalador y daremos clic en Next para continuar.



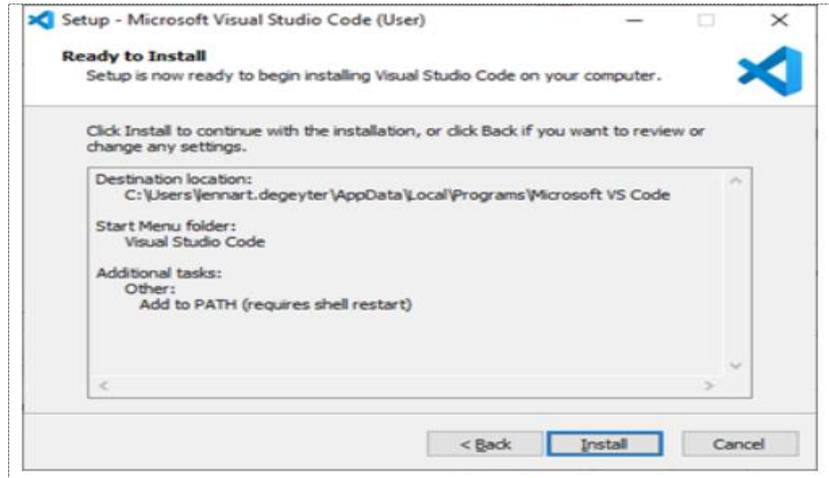
Paso 5: Mostrará una ventana como la que se muestra abajo, en donde se puede cambiar el nombre con el que nuestro acceso directo será creado, mantendremos el nombre predeterminado y daremos clic en Next para continuar.



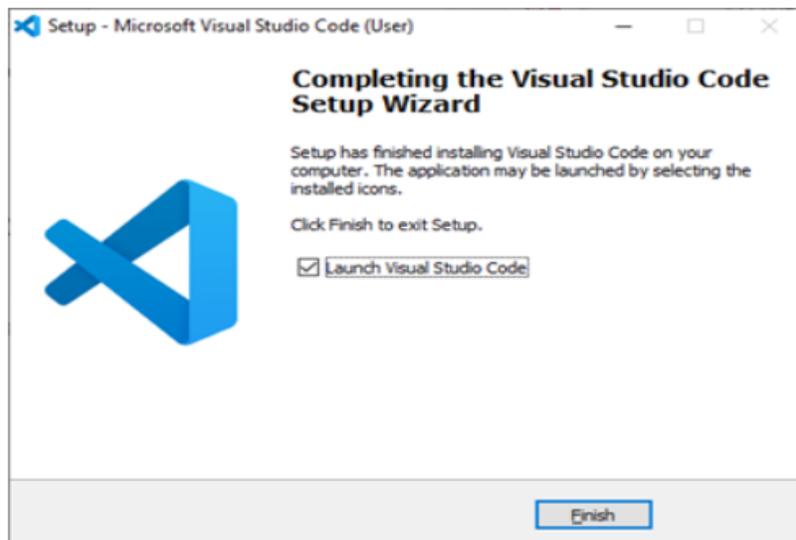
Paso 6: Aparecerá la ventana de seleccionar tareas de la instalación, como se muestra en la imagen. Seleccionaremos las tareas adicionales que creamos pertinentes. (Esto es a consideración del usuario). Una vez seleccionadas clic en Next para continuar.



Paso 7: Se mostrará la ventana en donde dice que estamos listos para instalar, y daremos clic en Install para iniciar la instalación de Visual Studio Code.



Paso 8: Con esto el programa estará instalado y preparado para usar. Seleccionaremos la casilla que dice "Launch Visual Studio Code" para que el programa inicie una vez dado clic en Finish. Como se muestra en la siguiente imagen.



Extensiones para Visual Studio Code

Estas extensiones o también llamadas plugins ayudaran para optimizar el tiempo en el desarrollo de software

1. Bracket Pair Colorized 2

Esta extensión permite identificar los corchetes a juego con los colores. El usuario puede definir qué tokens hacer coincidir y qué colores usar. Como se muestra a continuación:

```
22     public updateDocument(document: TextDocument) {
23         const documentDecoration = this.getDocumentDecorations(document);
24         if (documentDecoration) {
25             documentDecoration.triggerUpdateDecorations();
26         }
27     }
```

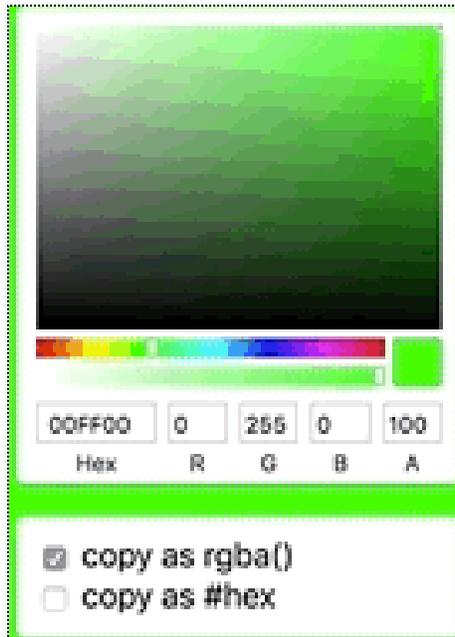
2. Color Highlight

Esta extensión aplica estilos a los colores en el código CSS / WEB que se encuentran en el documento donde se trabaja. Como se muestra en la siguiente:

```
background-color: #C594C5;
border: 1px solid #5FB3B3;
border-radius: 50%;
border: 1px solid #EB606B;
border: 1px solid #6699CC;
border: 1px solid #F2777A;
border: 1px solid #F99157;
```

3. Color Pick

La extensión Color Pick nos permitirá dar elegir un color en RGB o #hex para aplicar a nuestra página web. Como se muestra a continuación:



4. JavaScript Code Snippets

Esta extensión contiene abreviaturas útiles para JavaScript con la que podemos ser más rápidos al escribir código. Por ejemplo:

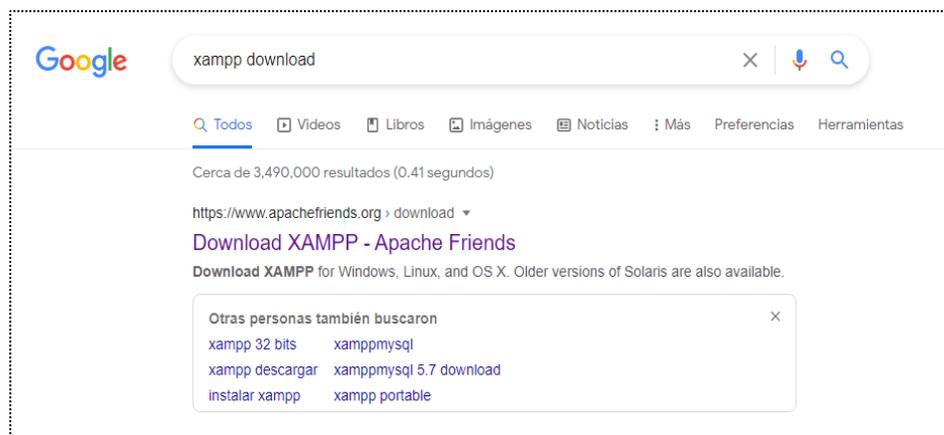
<code>imp</code>	imports entire module <code>import fs from 'fs';</code>
<code>imn</code>	imports entire module without module name <code>import 'animate.css'</code>
<code>imd</code>	imports only a portion of the module using destructuring <code>import {rename} from 'fs';</code>
<code>ime</code>	imports everything as alias from the module <code>import * as localAlias from 'fs';</code>
<code>ima</code>	imports only a portion of the module as alias <code>import { rename as localRename } from 'fs';</code>
<code>rqr</code>	require package <code>require('');</code>
<code>req</code>	require package to const <code>const packageName = require('packageName');</code>
<code>mde</code>	default module.exports <code>module.exports = {};</code>
<code>env</code>	exports name variable <code>export const nameVariable = localVariable;</code>
<code>enf</code>	exports name function <code>export const log = (parameter) => { console.log(parameter);};</code>
<code>edf</code>	exports default function <code>export default function fileName (parameter){ console.log(parameter);};</code>
<code>ecl</code>	exports default class <code>export default class Calculator { };</code>
<code>ece</code>	exports default class by extending a base one <code>export default class Calculator extends BaseClass { };</code>

4.3.2. Instalación del software en Ubuntu Budgie 18.04 LTS

A continuación, se mostrará el proceso para instalar el software necesario para trabajar el proyecto en Ubuntu Budgie 18.04 LTS.

Empezaremos mostrando la instalación de XAMPP. Para instalarlo debemos seguir los siguientes pasos:

1. Nos dirigimos a nuestro navegador preferido y en el buscador escribimos: XAMPP download. Como se muestra a continuación.



2. Ingresamos a la primera opción que dice “Dowload XAMPP – Apache Friends” y la página que mostrará es la siguiente

Apache Friends

Descargar Complementos Alojamiento Comunidad Acerca de

Buscar.. Buscar ES

Descargar

XAMPP es una distribución de Apache fácil de instalar que contiene MariaDB, PHP y Perl. Simplemente descarga y ejecuta el instalador. ¡Es así de fácil!

XAMPP para Windows 7.3.27, 7.4.16 & 8.0.3

Versión	Suma de comprobación	Tamaño
7.3.27 / PHP 7.3.27	¿Qué está incluido? md5 sha1	Descargar (64 bit) 154 Mb
7.4.16 / PHP 7.4.16	¿Qué está incluido? md5 sha1	Descargar (64 bit) 156 Mb
8.0.3 / PHP 8.0.3	¿Qué está incluido? md5 sha1	Descargar (64 bit) 156 Mb

Requisitos Complementos Más Descargas »

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

XAMPP para Linux 7.3.27, 7.4.16 & 8.0.3

Versión	Suma de comprobación	Tamaño
---------	----------------------	--------

Documentación/FAQs

No hay un manual para XAMPP. Escribimos la documentación en forma de preguntas frecuentes (FAQs). ¿Tienes una pregunta que no está respondida? Prueba los Foros o Stack Overflow.

- Linux Preguntas frecuentes
- Windows Preguntas frecuentes
- OS X Preguntas frecuentes
- OS X XAMPP-VM Preguntas frecuentes

Complementos

Bitnami proporciona herramientas gratuitas para instalar Drupal, Joomla!, WordPress y muchas otras aplicaciones populares en XAMPP. Visita Bitnami XAMPP o accede a la sección de complementos de XAMPP.

Y bajaremos hasta la parte de XAMPP para Linux que se muestra a continuación:

XAMPP para Linux 7.3.27, 7.4.16 & 8.0.3

Versión	Suma de comprobación	Tamaño
7.3.27 / PHP 7.3.27 ¿Qué está incluido?.	md5 sha1	Descargar (64 bit) 152 Mb
7.4.16 / PHP 7.4.16 ¿Qué está incluido?.	md5 sha1	Descargar (64 bit) 151 Mb
8.0.3 / PHP 8.0.3 ¿Qué está incluido?.	md5 sha1	Descargar (64 bit) 152 Mb

[Requisitos](#) [Complementos](#) [Más Descargas »](#)

3. Seleccionaremos la última versión que es la 8.03 / PHP 8.0.3 de 64 bits, cabe mencionar que la pagina detectara en automático la arquitectura del sistema operativo. Una vez que hagamos clic en el botón azul que dice “Descargar (64 bit)” se comenzará a descargar el programa y nos redirigirá a la siguiente página:

Apache Friends

[Descargar](#)
[Complementos](#)
[Alojamiento](#)
[Comunidad](#)
[Acerca de](#)

ES

¡Genial!

Tu descarga debería empezar automáticamente. Si no es así, haz clic aquí.

Leyendo

Asegúrate de leer estas instrucciones de instalación y preguntas frecuentes:

- [Linux Preguntas frecuentes](#)
- [Windows Preguntas frecuentes](#)
- [OS X Preguntas frecuentes](#)
- [OS X XAMPP-VM Preguntas frecuentes](#)

Puedes encontrar ayuda adicional en nuestros foros o en Stack Overflow.

Comparte XAMPP con tus amigos

Acabo de descargar #XAMPP de @ApacheFriends
<https://www.apachefriends.org> #opensource.

[Twitter](#)
[Facebook](#)
[Google+](#)

Complementos

Bitnami proporciona herramientas gratuitas para instalar Drupal, Joomla!, WordPress y muchas otras aplicaciones populares en XAMPP. Visita Bitnami XAMPP o accede a la sección de complementos de XAMPP.

Comunidad

XAMPP tiene más de 10 años – hay una gran comunidad detrás del proyecto. Puedes participar uniéndote a nuestros Foros, suscribiéndote a nuestra Lista de correo, uniéndote a nuestro Facebook, o siguiendo nuestras hazañas en Twitter.

Lista de correo

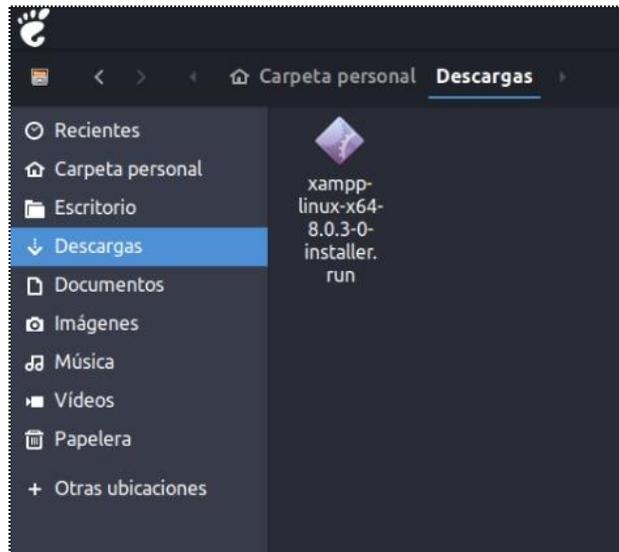
Email

Nombre

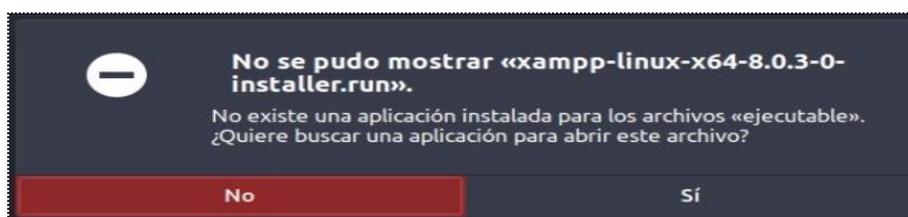
Apellidos (Opcional)

Por favor vea nuestras Políticas de Privacidad para aprender más acerca de cómo usamos su información personal.

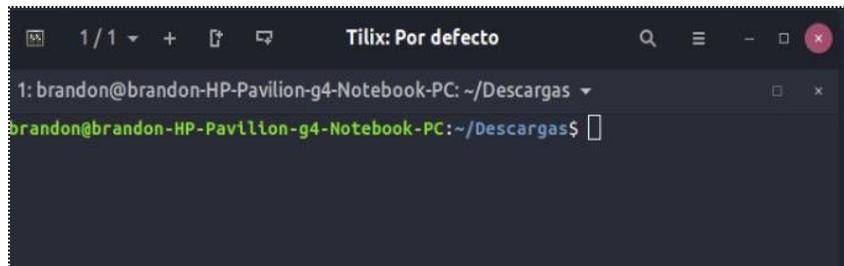
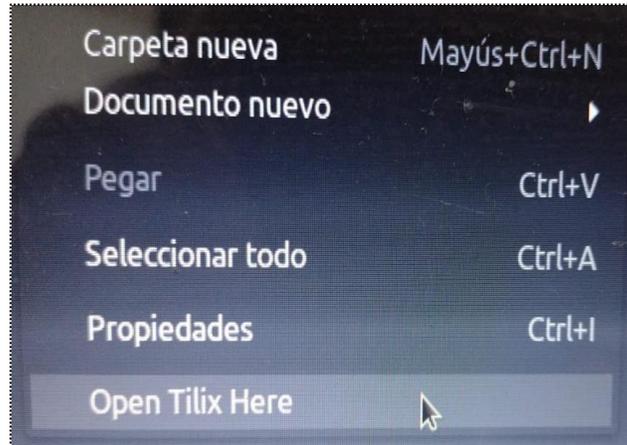
- Una vez concluida la descarga, minimizamos el navegador y nos dirigimos a nuestra carpeta de “Descargas” para visualizar el archivo descargado. Visualizándose de la siguiente manera:



- Una vez visualizado el archivo descargado, si se intenta ejecutar el archivo directamente mostrará el siguiente error. En caso de que se haya mostrado el error le daremos a la opción de “No”.



- Para instalarlo correctamente procederemos primero a presionar clic derecho (sin seleccionar nada) dentro de la carpeta de descargas y le daremos clic a la opción que dice: “Open Tilix Here”. Esto nos abrirá la consola de operaciones. Como se muestra a continuación:



7. Una vez abierta la consola de operaciones ingresamos el comando "ls -l" esto para ver el archivo que descargamos anteriormente.



8. Una vez que se visualizan correctamente el próximo paso es generar los permisos para realizar la instalación. Esto se hará con el comando: "sudo chmod 755" seguido del nombre del archivo que es: "xampp-linux-x64-8.0.3-0-installer.run". Ya puesto el nombre presionaremos la tecla enter y nos solicitará nuestra contraseña de usuario. (Esta contraseña es la contraseña personal que nos solicita cuando instalamos Ubuntu por primera vez en nuestro equipo, por obvias razones no se revelará la contraseña del equipo)

Se verá de la siguiente forma:

```
Tilix: Por defecto
1: brandon@brandon-HP-Pavilion-g4-Notebook-PC: ~/Descargas
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~/Descargas$ ls -l
total 156576
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~/Descargas$ sudo chmod 755 xampp-linux-x64-8.0.3-0-installer.run
[sudo] contraseña para brandon: 
```

9. Una vez colocada la contraseña, pulsaremos la tecla enter nuevamente. Y escribiremos el comando `ls -l` y pulsaremos enter. Cuando pulsemos enter nos mostrará el nombre del archivo en verde, esto significa que ya tiene permiso de ejecución. Ya con el permiso de ejecución escribiremos el siguiente comando y daremos enter: `sudo ./xampp-linux-x64-8.0.3-0-installer.run` y presionamos enter. Y empezara a cargar la ventana de instalación

```
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~/Descargas$ ls -l
total 156576
-rwxr-xr-x 1 brandon brandon 160327491 mar 14 19:56 xampp-linux-x64-8.0.3-0-installer.run
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~/Descargas$ sudo ./xampp-linux-x64-8.0.3-0-installer.run
```

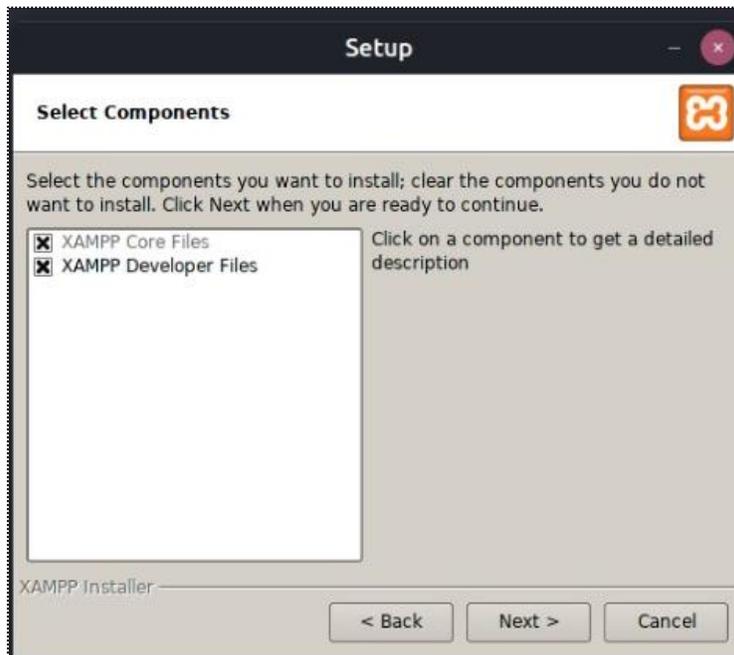
10. Mostrará la siguiente pantalla:



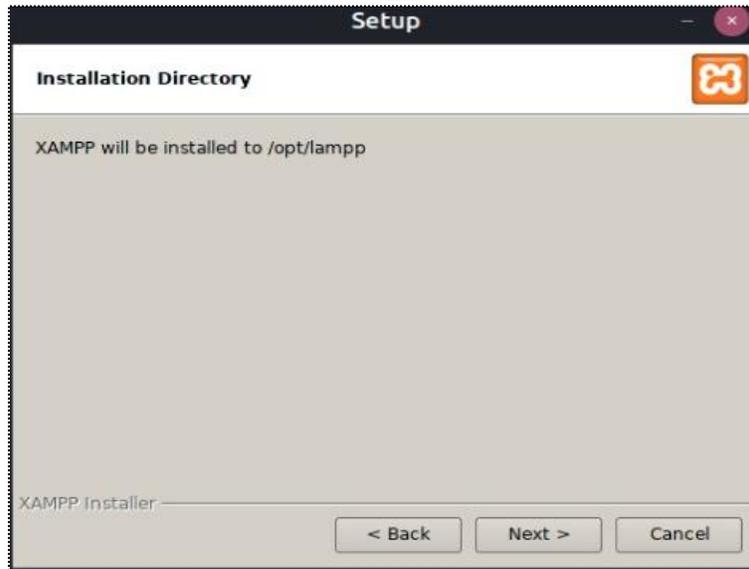
11. Una vez que paso la ventana de carga nos mostrará la siguiente, daremos clic en el botón “Next”



12. Nos mostrará la siguiente ventana donde nos pide seleccionar los componentes que deseamos instalar, marcaremos ambas casillas y daremos clic en el botón de “Next”



13. Mostrará la siguiente ventana en la que menciona cuál será el directorio donde quede nuestro programa, daremos clic en el botón de “Next”



14. Mostrará la pantalla siguiente, le daremos clic en el botón de “Next”



15. Mostrará una ventana que dirá que ya está listo para instalarse el programa, daremos clic en el botón "Next" para continuar.



16. Enseguida comenzará la instalación y mostrará las siguientes pantallas, solo es cuestión de esperar a que termine la carga.

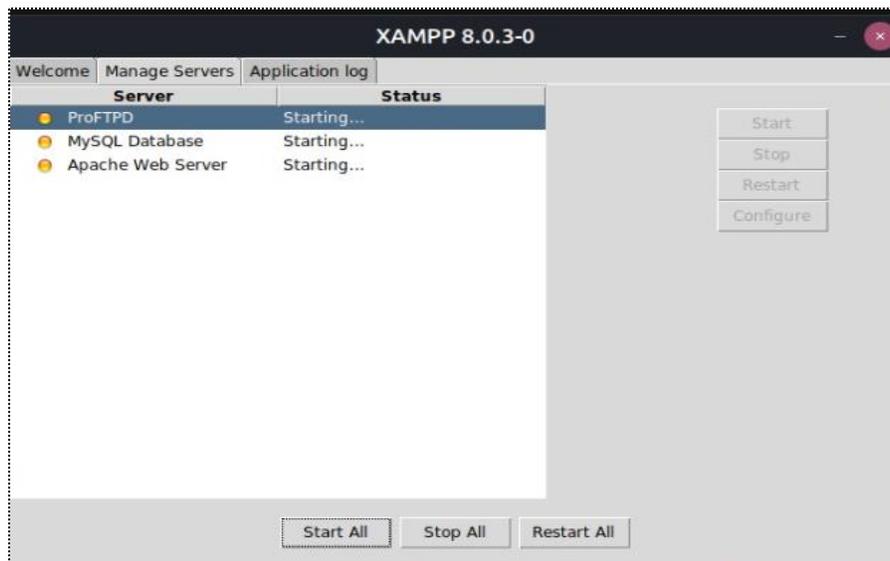
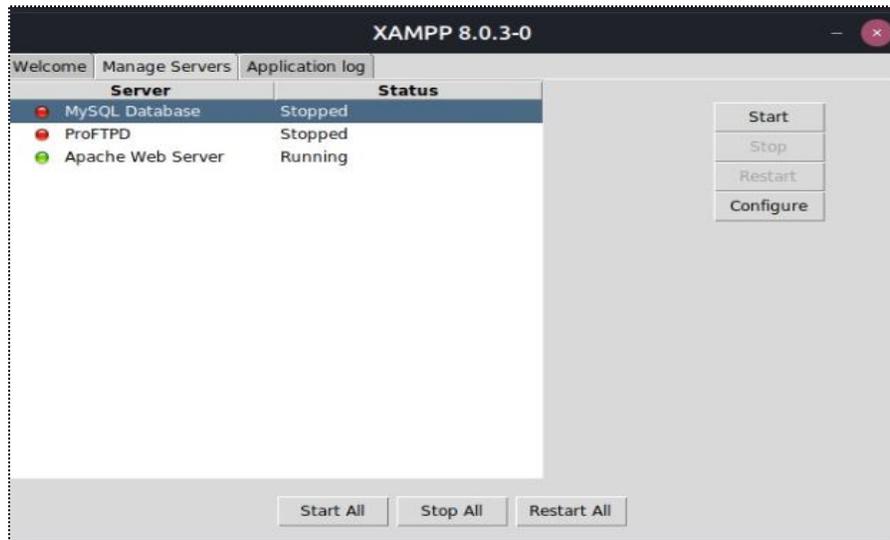


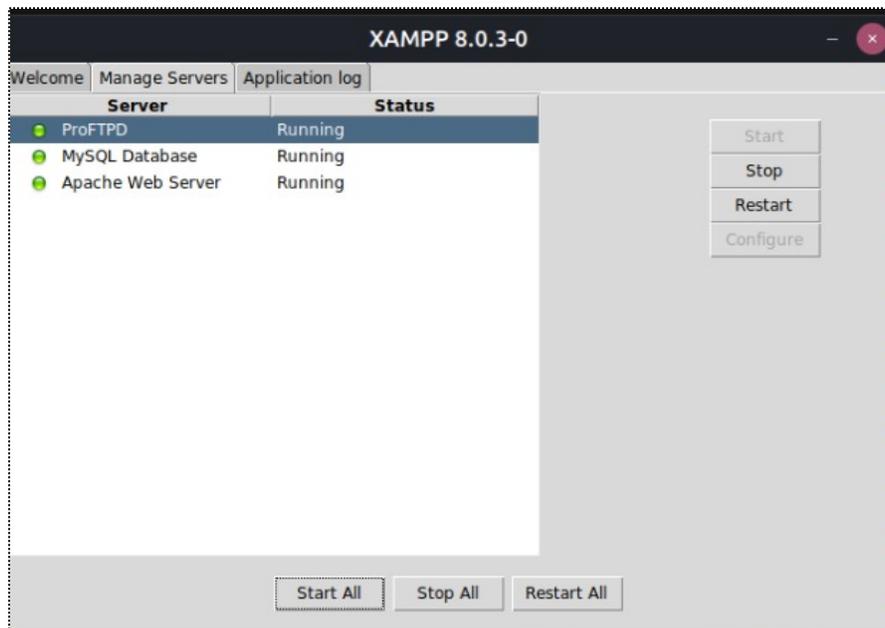


17. Una vez concluida la carga de la instalación mostrará la siguiente pantalla de finalización

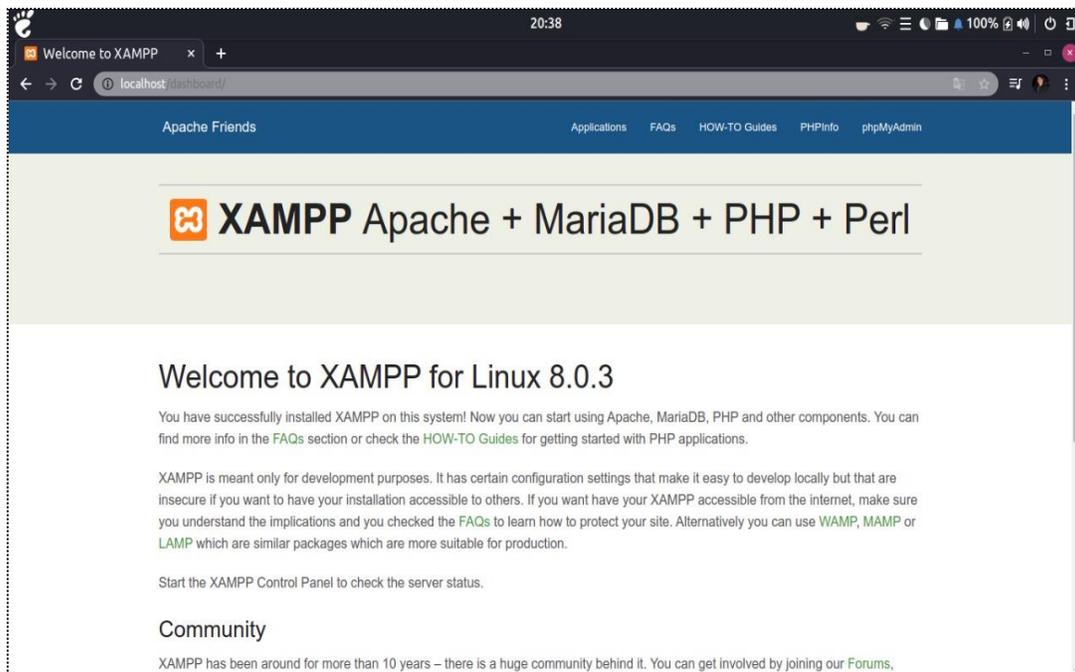


18. Seguido de esto veremos qué servicios están activos y nos vamos a la pestaña que dice “Manage Servers”, como vemos nuestros servicios están detenidos, por lo que daremos clic al botón de “Start All” para iniciar todos los servicios. Una vez cargados los servicios estos se pondrán en color verde indicando que ya están activos.

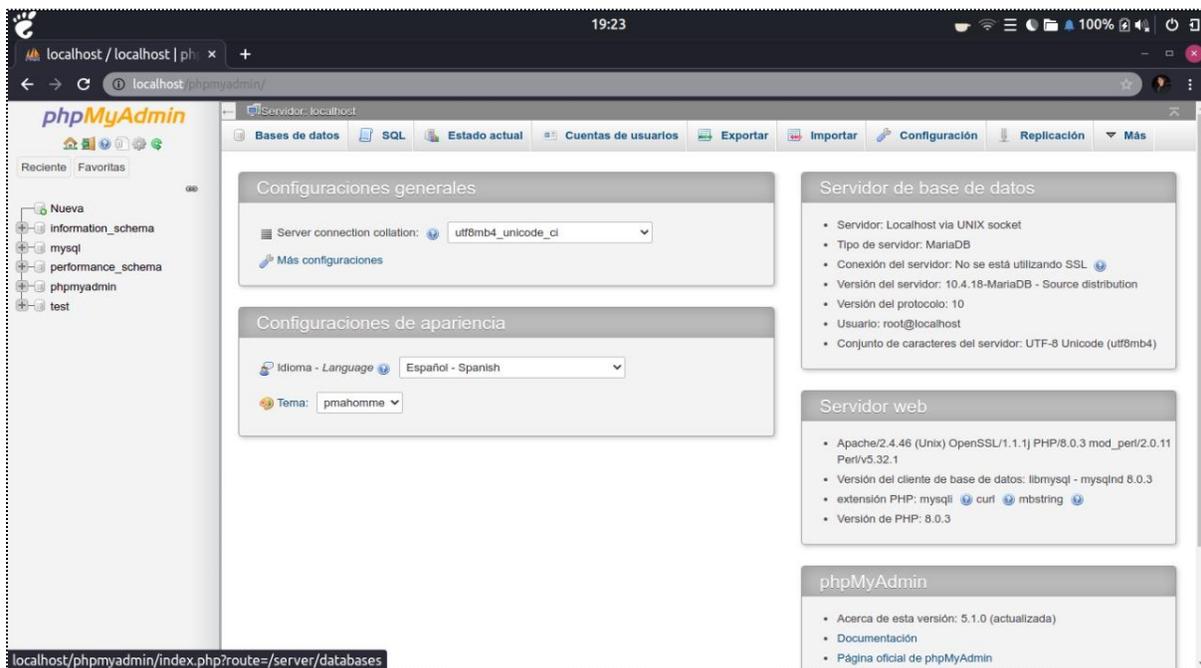




19. Una vez iniciados los servicios, comprobaremos que se iniciaron correctamente, esto lo haremos en nuestro navegador escribiendo en la url *“localhost/dashboard”* deberá de abrir la siguiente página, esta es la página de bienvenida de XAMPP



20. Para acceder a “phpMyAdmin” deberemos ya sea escribir en la url “localhost/phpmyadmin”, o darle al botón “phpMyAdmin” que muestra la pantalla de bienvenida. Esta nos mostrara la página principal de phpMyAdmin que es donde realizaremos la creación de la base de datos.



4.4. Desarrollo de la página web en HTML

En este apartado se explicará de forma detallada como se estructura la página web y las clases que se crearan para que trabajaremos en este proyecto

4.4.1. Estructura básica del HTML

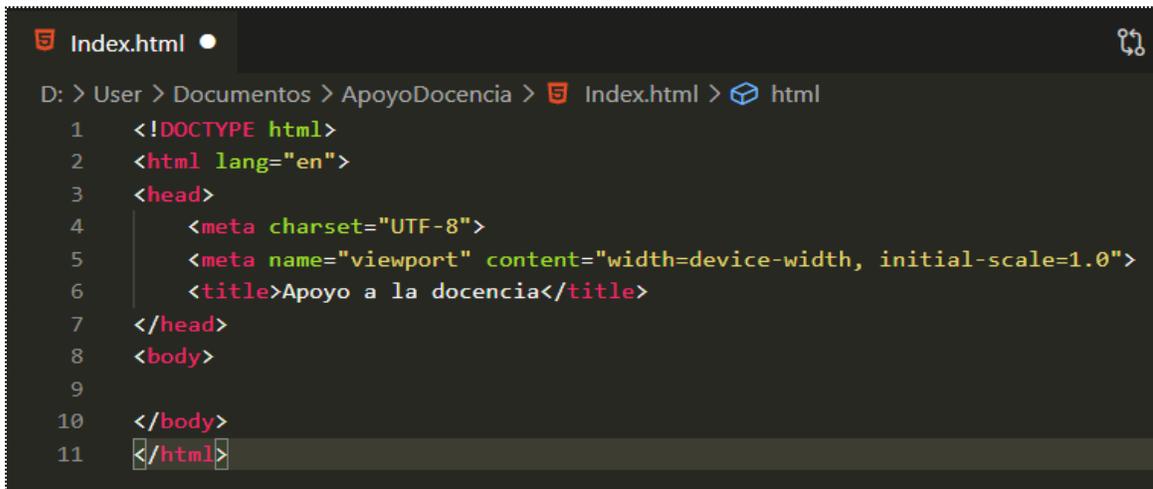
Cada página contiene al inicio un <html>, contiene información sobre la versión de HTML. La etiqueta que contiene lang <html Lang = “en”> se utiliza para indicar el idioma del contenido de un elemento.

A continuación, viene la cabecera, delimitada por <head > y </head> esta etiqueta provee información general (metadatos) acerca del documento, incluyendo su título y enlaces a scripts y hojas de estilos.

Dentro del head tenemos la etiqueta <meta> encabeza un documento HTML y suministra información codificada a navegadores y motores de búsqueda sobre una página web. Los

metadatos son invisibles para el usuario y se encargan de añadir información para facilitar el análisis de los archivos HTML y la gestión del contenido de una página web.

La etiqueta `<title></title>` en la sección de *Head* es usada para identificar contenidos de un documento, por lo que se aconseja poner el título que refleje con claridad el contenido de la página web, este título puede contener diferentes caracteres, pero no código. Como se muestra en la siguiente imagen:



```
Index.html
D: > User > Documentos > ApoyoDocencia > Index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Apoyo a la docencia</title>
7  </head>
8  <body>
9
10 </body>
11 </html>
```

Dentro de las etiquetas `<body> </body>` se escribe el contenido de la página, además que puede contener los siguientes atributos:

- **Background** = archivo. Establece como fondo de la página la imagen especificada en el archivo.
- **Bgcolor** = color. Establece un color de fondo para la página.
- **Text** = color. Establece el color del texto
- **Link** = color. Establece el color del texto de los vínculos no visitados
- **Vlink** = color. Establece el color del texto que de los vínculos visitados
- **Alink** = color. Especifica el color del texto que de los vínculos cuando son seleccionados por el usuario.

4.4.2. Creación del archivo index.html.

Este archivo contiene la estructura de la página inicial de la plataforma, dentro de las etiquetas `<body>` `</body>`, escribimos una etiqueta `<div>` `</div>` lo que hace esta etiqueta es crear un bloque dentro de la página web, con la clase `contenedor` (la palabra `class` servirá para darle estilos con `css` más adelante) en esta etiqueta ponemos el contenido de la página para tener un mejor orden dentro de esta etiqueta escribimos el título que tendrá.

Fragmento del código:

```
<div class="contenedor">
  <h3 class="titulo" id="titulo">Enseñanza de la calidad en ingeniería de software </h3>
</div>
```

Se visualizará en el navegador de la siguiente manera:

Enseñanza de la calidad en ingeniería de software

En el `<div>` `contenedor` escribimos un nuevo `<div>` con la `class` `login`, dentro colocamos las etiquetas `<a>` `` que harán referencia al Inicio de la página y a otra página que se enlaza al registro del usuario (esta página se creará más adelante).

Fragmento del código:

```
<div class="login">
  <a href="index.html" >Inicio</a>
  <a href="Registro.html">Registrarse</a>
</div>
```

Se visualizará en el navegador de la siguiente manera:

Enseñanza de la calidad en ingeniería de software

[Inicio](#) [Registrarse](#)

Creamos un nuevo `<div></div>` con la clase `acceso`, esta `<div>` de acceso será para el control de una pequeña base de datos que contiene el nombre de usuario y su contraseña, además que contiene el formulario del nombre de usuario que deberá llenar para ingresar a la página, contiene un `<label>`, esta etiqueta está asociada a un control de formulario.

La etiqueta `<input>` se usa para crear controles en un formulario de la web, que envía datos de lo escribe el usuario. Esta etiqueta varía considerablemente dependiendo del valor de su atributo mediante `type`.

A continuación, tenemos otro `<div></div>` con la `class` `contraseña` dentro de esta tenemos `<form>` este el pequeño formulario que deberá llenar el usuario para escribir su contraseña, contiene un `label` de la contraseña con su respectivo `input` para que el usuario escriba su contraseña.

Fragmento del código:

```

<!-- INGRESO -->
<div class="acceso">
  <form>
    <div class="usuario">
      <!-- Nombre de usuario -->
      <label for="nombre-usuario">Nombre de usuario</label>
      <br>
      <input type="text" placeholder="Nombre de usuario">
    </div>
  </form>
  <!-- Contraseña -->
  <form>
    <div class="contrasena">
      <label for="contraseña">Contraseña</label>
      <br>
      <input type="password" placeholder="Ingresa tu contraseña">
    </div>
  </form>
</div>

```

Se visualizará en el navegador de la siguiente manera:

Nombre de usuario

Contraseña

Ahora en caso de que el usuario no recuerde la contraseña tenemos una nueva etiqueta `<div></div>` con la *class* `recupcontra` dentro tiene otra etiqueta `<a> ` que nos dirige a una nueva página donde el usuario podrá recuperar su contraseña.

La etiqueta `<div></div>` con la clase `"btn-ingresar"` contiene un input de tipo botón que el usuario dará clic para ingresar al sistema si su nombre de usuario y la contraseña son correctas.

Fragmento del código:

```
<div class="recupcontra">
  <a href="RecuperarContraseña.html">Recuperar contraseña</a>
</div>

<div class="btn-ingresar">
  <input type="submit" value="Ingresar">
</div>
```

Se visualizará en el navegador de la siguiente manera:



Para finalizar tenemos la etiqueta `<footer></footer>` esta etiqueta representa al pie de una sección o documento, aquí habitualmente se escribe información de quien realizo la página, firmas, información de licencias etc. Lo que tenemos es un copyright para proteger los derechos exclusivos de la página y de esta forma no se use para uso comercial o la divulgación no autorizada.

Fragmento del código:

```
<footer>
  <p>Copyright &copy; Sistema de apoyo a la docencia</p>
</footer>
```

Se visualizará en el navegador de la siguiente manera:

Copyright © Sistema de apoyo a la docencia

Código completo HTML del archivo *index.html*:

```

index.html x
D: > User > Documentos > ApoyoDocencia > SitWeb > index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <meta http-equiv="X-UA-Compatible" content="ie=edge">
8    <title>Apoyo a la docencia</title>
9    <!-- <link rel="stylesheet" href="css/stylesindex.css">
10   <link href="https://fonts.googleapis.com/css?family=Poppins&display=swap" rel="stylesheet"> -->
11
12   <!-- IMPORTANDO LIBRERIA JQUERY -->
13   <script src="js/jquery-3.4.1.min.js"></script>
14   <!-- MAIN.JS -->
15   <script src="js/main.js"></script>
16 </head>
17 <body>
18   <div class="contenedor">
19     <h3 class="titulo" id="titulo">Enseñanza de la calidad en ingeniería de software </h3>
20
21     <div class="login">
22       <a href="index.html" >Inicio</a>
23       <a href="Registro.html">Registrarse</a>
24     </div>
25     <!-- INGRESO -->
26     <div class="acceso">
27       <form>
28         <div class="usuario">
29           <!-- Nombre de usuario -->
30           <label for="nombre-usuario">Nombre de usuario</label>
31           <br>
32           <input type="text" placeholder="Nombre de usuario">
33         </div>
34       </form>
35       <!-- Contraseña -->
36       <form>
37         <div class="contrasena">
38           <label for="contraseña">Contraseña</label>
39           <br>
40           <input type="password" placeholder="Ingresa tu contraseña">
41         </div>
42       </form>
43       <div class="recupcontra">
44         <a href="RecuperarContraseña.html">Recuperar contraseña</a>
45       </div>
46
47       <div class="btn-ingresar">
48         <input type="submit" value="Ingresar">
49       </div>
50     </div>
51     <!-- FinIngreso -->
52   </div>
53   <Footer>
54     <p>Copyright &copy; Sistema de apoyo a la docencia</p>
55   </Footer>
56 </div>
57 </body>
58
59 </html>

```

Vista en el navegador de la página Apoyo a la docencia (sin CSS):

Enseñanza de la calidad en ingeniería de software

[Inicio](#) [Registrarse](#)

Nombre de usuario

Contraseña

[Recuperar contraseña](#)

Copyright © Sistema de apoyo a la docencia

4.4.3. Estructura HTML para el registro de usuario

Creamos nuevo archivo para el registro con extensión HTML, posteriormente añadimos la hoja CSS que tendrá su estilo (este archivo se creará más adelante), el tipo de fuente y un archivo kit Font awesome de la página oficial de fontkit lo que hace este archivo es añadir iconos de forma más rápida.

Fragmento del código:

```

Registro.html
D: > User > Documentos > ApoyoDocencia > SitWeb > Registro.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <meta http-equiv="X-UA-Compatible" content="ie=edge">
7 <title>Registro</title>
8
9 <!-- IMPORTANDO ARCHIVO CSS -->
10 <link rel="stylesheet" href="css/stylesRegistro.css">
11 <!-- IMPORTANDO FUENTES DE GOOGLE -->
12 <link href="https://fonts.googleapis.com/css?family=Poppins&display=swap" rel="stylesheet">
13 <!-- IMPORTANDO KIT FONTAWESOME ES PARA LOS ICONOS -->
14 <script src="https://kit.fontawesome.com/94eae82570.js" crossorigin="anonymous"></script>
15
16 </head>

```

Dentro de la etiqueta `<body>` `</body>` en `<header>` `</header>` creamos una `<div>``</div>` de clase contenedor que tendrá unas imágenes y un título.

Fragmento del código:

```
<header>  
  <div class="contenedor">  
      
      
    <h3 class="registro">Registro</h3>  
  </div>  
</header>
```

Se visualizará en el navegador de la siguiente manera:



Cabe mencionar que la segunda imagen agregada aún no se ve ya que está encimada en la primera imagen, esto se corregirá posteriormente en el archivo CSS para separar estas imágenes.

Se añade otra `<div>``</div>` con la clase formulario, esta `<div>` contendrá los campos que el usuario llenará para poder registrarse, así que agregamos la clase `<form>``</form>` con el *id* `form`, (un *id* funciona similar a una clase para dar un estilo CSS solo que este se representa

como un identificador único así que no puede haber otro con este nombre), dentro de nuestro form agregamos una nueva `<div></div>` con la clase `columns` ya que esta estará dividida en dos columnas (estas se modifican en el CSS) una para el nombre del usuario y otra para su apellido además que contienen una etiqueta `<i>` que contiene un icono y la etiqueta `<input>` para que el usuario pueda añadir lo antes mencionado

Fragmento del código:

```
<div class="formulario">
  <form action="" id="form">
    <div class="columns">
      <div class="contenedor-campo">
        <p>Nombre: </p>
        <!-- ICONO DE USUARIO -->
        <i class="fas fa-user"></i>
        <input type="text" class="campo">
      </div>
      <div class="contenedor-campo" id="apellido">
        <p>Apellido: </p>
        <!-- ICONO DE USUARIO -->
        <i class="fas fa-user"></i>
        <input type="text" class="campo">
      </div>
    </div>
  </form>
</div>
```

Se visualizará en el navegador de la siguiente manera:



Registro

Nombre:

Apellido:

Agregamos una nueva `<div>` con la clase contenedor de campo, al igual que para el nombre y el apellido añadimos una para la contraseña, el correo electrónico, la carrera y el número de cuenta.

Fragmento de código:

```
<div class="contenedor-campo">
  <p>Contraseña: </p>
  <!-- ICONO DE USUARIO -->
  <i class="fas fa-key"></i>
  <input type="password" class="campo">
</div>

<div class="contenedor-campo">
  <p>Correo electrónico: </p>
  <!-- ICONO DE EMAIL -->
  <i class="fas fa-envelope"></i>
  <input type="text" class="campo">
</div>

<div class="contenedor-campo">
  <p>Carrera: </p>
  <!-- ICONO DE CARRERA -->
  <i class="fas fa-graduation-cap"></i>
  <input type="text" class="campo">
</div>

<div class="contenedor-campo">
  <p>Num. Cuenta: </p>
  <!-- ICONO DE NUM DE CUENTA -->
  <i class="fas fa-id-card"></i>
  <input type="text" class="campo">
</div>
```

Se visualizará en el navegador de la siguiente manera:



Registro

Nombre:

Apellido:

Contraseña:

Correo electrónico:

Carrera:

Num. Cuenta:

Para finalizar agregamos un `<div></div>` con la clase botón que contiene un input de tipo botón para enviar la información a una base de datos. Otro `<div></div>` con la clase cuenta registrada que regresa a la página principal al usuario, si es que ya cuenta con una cuenta.

Fragmento de código:

```

<!-- BOTON PARA ENVIAR -->
<div class="boton">
  <input type="button" class="button" value="Enviar">
</div>
<!-- TEXTO DE CUENTA REGISTRADA -->
<div class="cuentaRegistrada">
  <p>Si ya cuentas con una cuenta da click <a href="index.html">aqui</a></p>
</div>

```

Se visualizará en el navegador de la siguiente manera:



Registro

Nombre:

Apellido:

Contraseña:

Correo electrónico:

Carrera:

Num. Cuenta:

Si ya cuentas con una cuenta da click [aquí](#)

4.5. Desarrollo de la página web en CSS

A los elementos de la página web creados con HTML se les dará la apariencia que se desee utilizando CSS: colores, espacios entre elementos, tipos de letra, entre otros. Separando de esta forma la estructura de la presentación.

4.5.1. Creación del archivo CSS para index.html

Se crea un archivo con la extensión CSS, este llevará por nombre *stylesindex.css*, posteriormente lo invocamos al HTML dentro de las etiquetas de `<head></head>` y debajo del `<title> </title>` ya que HTML comienza a leer el código de arriba hacia abajo.

Fragmento de código:

```

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Apoyo a la docencia</title>
  <link rel="stylesheet" href="css/stylesindex.css">
</head>

```

Se inicia con la etiqueta `body`.

Se invoca la etiqueta `<body>` y entre corchetes se escribe:

Background: Es el fondo que tendrá la página, en este caso pasamos una imagen.

Background-size: Asigna el tamaño de este fondo

Margin/Padding: Son los márgenes de la página.

Background-repeat: En este caso se utiliza el *"no-repeat"* para que el fondo no se repita

Fragmento de código:

```

stylesindex.css X
D: > User > Documentos > ApoyoDocencia > SitWeb > css > stylesindex.css
1  body{
2    background: url(../Img/Portada.jpg), no-repeat center top;
3
4    background-size: cover;
5    margin: 0;
6    padding: 0;
7    background-repeat: no-repeat ;
8  }

```

Se visualizará en el navegador de la siguiente manera:



En esta imagen se puede observar que esta ajustada al tamaño completo de la pantalla.

Posteriormente pasamos darle estilo a la clase contenedor, el título de la página y el login, dentro de la clase contenedor alineamos el texto al centro con una fuente de tamaño de 10píxeles, un margen de 0 y una fuente, esta fuente la llamamos al HTML debajo de nuestro archivo CSS.

Fragmento del código:

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Apoyo a la docencia</title>
  <link rel="stylesheet" href="css/stylesindex.css">
  <link href="https://fonts.googleapis.com/css?family=Poppins&display=swap" rel="stylesheet">
```

Ahora dentro de la clase título solo le añadimos un color y un tamaño de fuente, en la clase login agregamos un margen hacia arriba.

Fragmento del código:

```
.contenedor{
  text-align: center;
  font-size:10 px;
  padding: 0;
  margin: 0;
  font-family: 'Poppins', sans-serif;
}

.titulo{
  color: #fdfdff;
  font-size: 20px;
}

.login{
  margin-top: 200px;
}
```

Se puede observar que los textos ya están centrados y la fuente cambio, al igual de un cambio de color en el título de la página.

Posteriormente se crea un estilo a nuestra etiqueta <a> por lo que agregamos un margen hacia arriba de 200px, alineado a la izquierda con un color #ffff, un margen en general de 50px y quitamos el subrayado al texto, después se crea en la etiqueta <a> un *hover* que hace un efecto de selección al link de esta forma el usuario podrá saber que puede dar clic en él.

Para nuestra etiqueta <form> solo la centramos con un margen en general de 30px

Fragmento del código:

```
a{
  margin-top: 200px;
  text-align: left;
  color: #ffffff;
  margin: 50px;
  text-decoration: none;
}
a:hover{
  text-decoration:underline;
}
form{
  margin: 30px;
  text-align: center;
}
```

Se visualizará en el navegador de la siguiente manera:



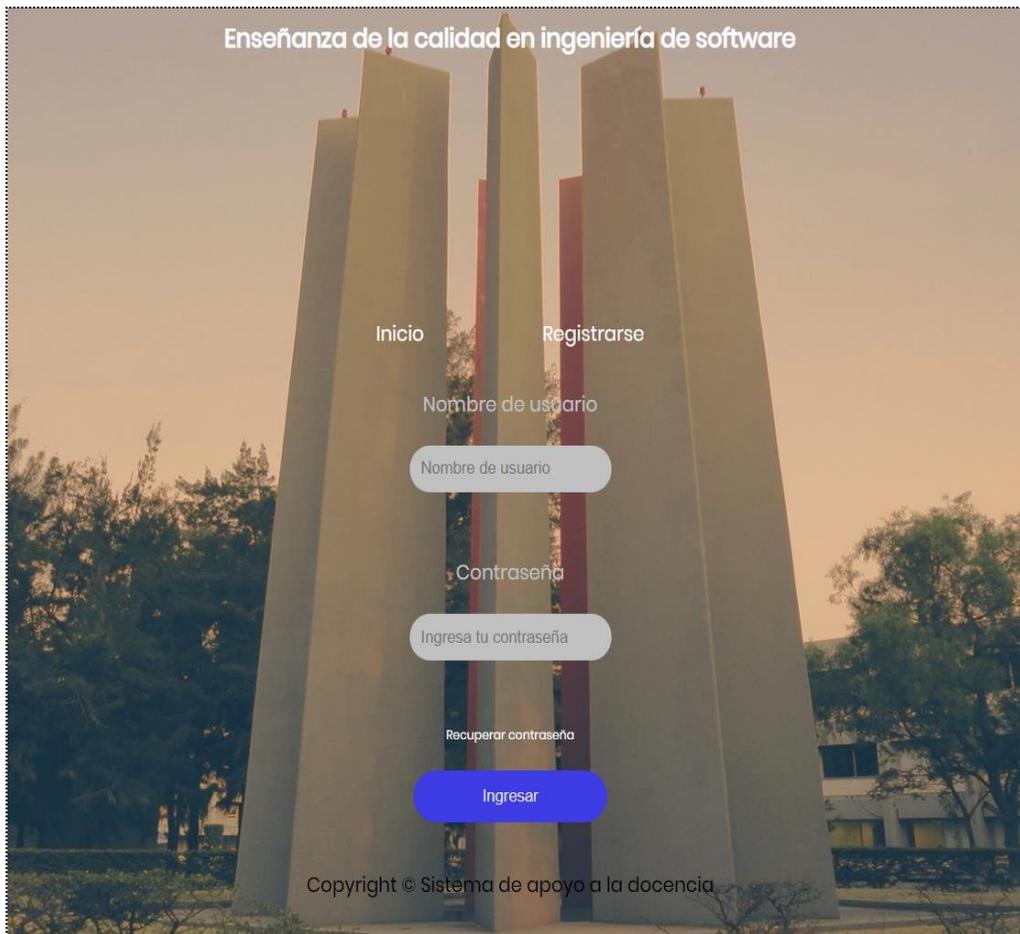
Se da estilo a la clase de acceso con el color de fondo `#c1c1c1` y un texto alineado a la izquierda, la etiqueta `<input>` la posicionamos al centro y un borde circular agregamos un margen de 20px y un color de letra `#0000`, al input con la clase de nuestro botón agregamos un color `#3d3de4` y un color de letra `e6e1e1`, al igual que en la etiqueta de los links agregamos un hover para que el usuario al colocar el cursor en este botón cambie el color de letra a `#3d3d3d` y su fondo a `#e6e1e1` con una transición de 3 segundos. A la clase de `recupcontra` solo le cambiamos el tamaño de fuente.

Fragmento de código:

```
.acceso form label{
  color: #c1c1c1;
}
.acceso label{
  text-align: left;
}

input {
  float:center;
  border-radius: 20px;
  border: none;
  padding: 10px;
  margin: 20px;
  background: #c1c1c1;
  color: #000000;
}
.btn-ingresar input{
  padding:12px 60px 12px 60px;
  background: #3d3de4;
  color: #e6e1e1;
}
.btn-ingresar input:hover{
  padding:12px 60px 12px 60px;
  background: #e6e1e1;
  color: #3d3d3d;
  cursor: pointer;
  -webkit-transition: all .3s ease;
  -o-transition: all .3s ease;
  transition: all .6s ease;
}
.recupcontra{
  font-size: 10px;
}
```

Se visualizará en el navegador de la siguiente manera:

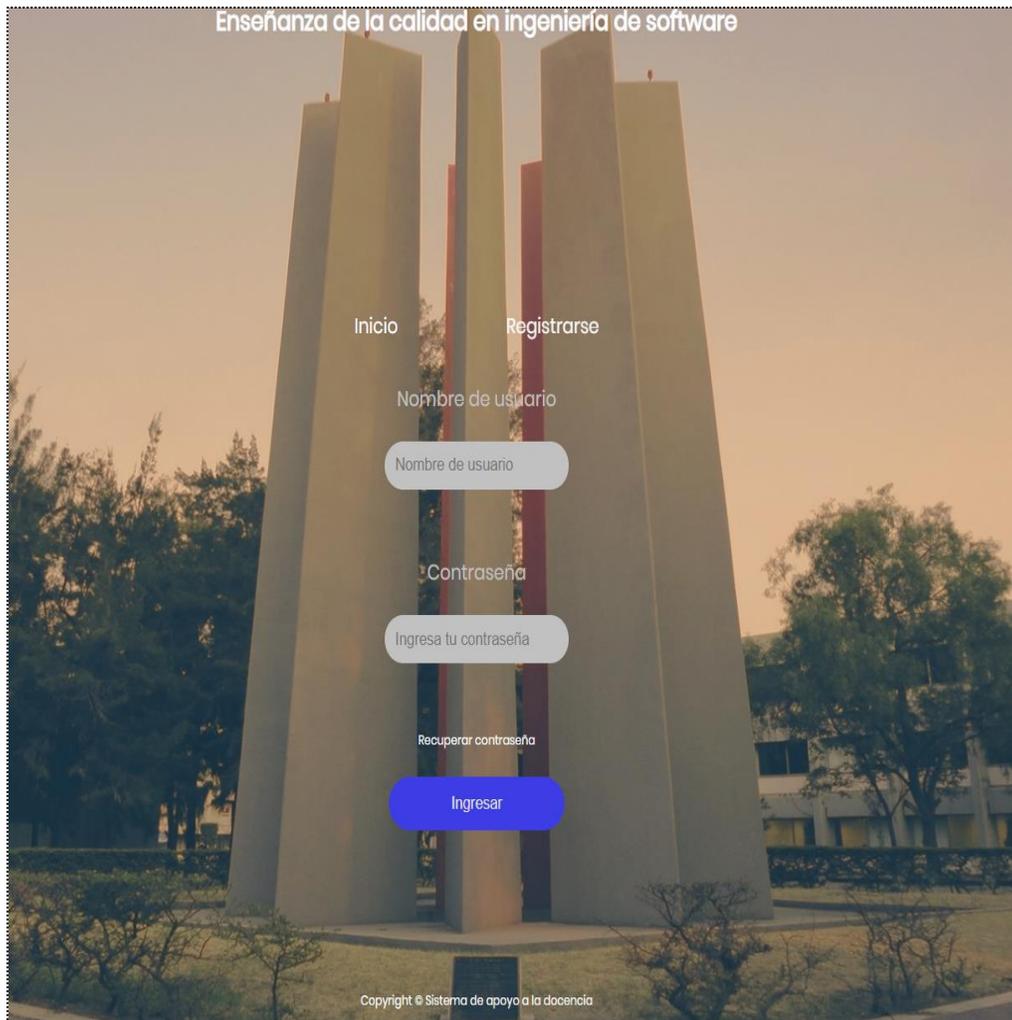


Para la etiqueta `<footer>` le añadimos un margen hacia arriba de `100px`, un tamaño de fuente de `10px`, el color `#fff`, en la fuente de familia `poppins` y el texto alineado al centro

Fragmento del código:

```
footer{
  margin-top: 100px;
  font-size: 10px;
  color: #ffffff;
  font-family: 'Poppins', sans-serif;
  text-align: center;
}
```

Se visualizará en el navegador de la siguiente manera:



4.5.2. Creación del archivo CSS para la página de registro

Dentro de nuestro archivo agregamos un estilo al `<body></body>` con un color de fondo `#081883` un *padding* (establece el espacio de relleno requerido por todos los lados de un elemento.) de 50 px a la izquierda y derecha y una fuente de tipo Poppins, las imágenes que están dentro de la clase en el HTML (*unamlogo* y *feslogo*) quedan alineadas a la izquierda y derecha, la clase registro agregamos un *padding* hacia arriba de 85px y un tamaño de fuente de 40px con un color *aliceblue* alineado al centro, mientras que a la clase formulario agregamos un fondo blanco un *padding* de 20px un tamaño de 650px y un borde (este dará un estilo de

luz) con un color en *rgb 255,255,255,0.616* y a la etiqueta *<form>* un *padding* de *60px* a la izquierda y derecha.

Fragmento del código:

```
body {
  background-color: #081883;
  padding: 0 50px 0px 50px;
  font-family: 'Poppins', sans-serif;
}
.unamLogo {
  float: left;
}
.fesLogo {
  float: right;
}
.registro {
  padding-top: 85px;
  font-size: 40px;
  color: aliceblue;
  text-align: center;
}

.formulario {
  background-color: #f3f3f3;

  padding: 20px;
  margin: 0 auto;
  width: 650px;
  border-radius: 15px;
  -webkit-box-shadow: 0px 0px 30px 5px rgba(255, 255, 255, 0.616);
  -moz-box-shadow: 0px 0px 30px 5px rgba(255, 255, 255, 0.616);
  box-shadow: 0px 0px 30px 5px rgba(255, 255, 255, 0.616);
}

form{
  padding: 0 60px 0px 60px;
}
```

Se visualizará en el navegador de la siguiente manera:

The image shows a registration form titled "Registro" on a dark blue background. The form is centered and contains the following fields and elements:

- Nombre:** A text input field with a person icon to its left.
- Apellido:** A text input field with a person icon to its left.
- Contraseña:** A text input field with a key icon to its left.
- Correo electrónico:** A text input field with an envelope icon to its left.
- Carrera:** A text input field with a graduation cap icon to its left.
- Num. Cuenta:** A text input field with a document icon to its left.
- Enviar:** A button located below the "Num. Cuenta" field.
- Footer:** A link that says "Si ya cuentas con una cuenta da click [aquí](#)".

The background features the UNAM logo on the left and the UNAM Aragón logo on the right.

Para la clase *cuentaRegistrada* solo añadimos un color con un texto alineado al centro, al contenedor-campo agregamos un borde de 1px y un padding de 10px además de un margen a la derecha y un color, a la clase campo agregamos un borde de 5px alineado verticalmente hacia arriba y un color de fondo, la clase campo

Solo contiene un borde de 50x, un color un padding de 40px y un tamaño.

Fragmento del código:

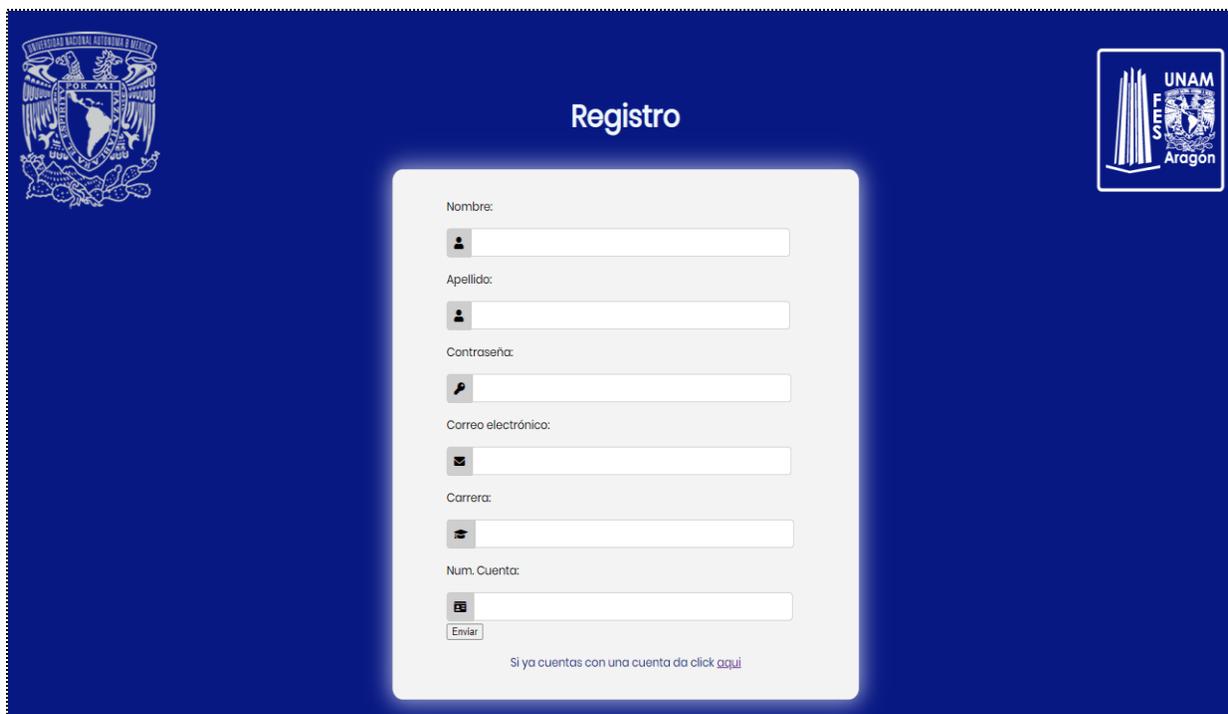
```

.cuentaRegistrada{
  text-align: center;
  color: #081883;
}

.contenedor-campo i {
  border: solid 1px #ccc;
  padding: 10px;
  margin-right: -4px;
  vertical-align: top;
  background-color: #cecece;
  border-radius: 5px 0px 0px 5px;
}

.campo {
  border-radius: 0px 5px 5px 0px;
  border: solid 1px #ccc;
  padding: 10px;
  width: 450px;
}
    
```

Se visualizará en el navegador de la siguiente manera:

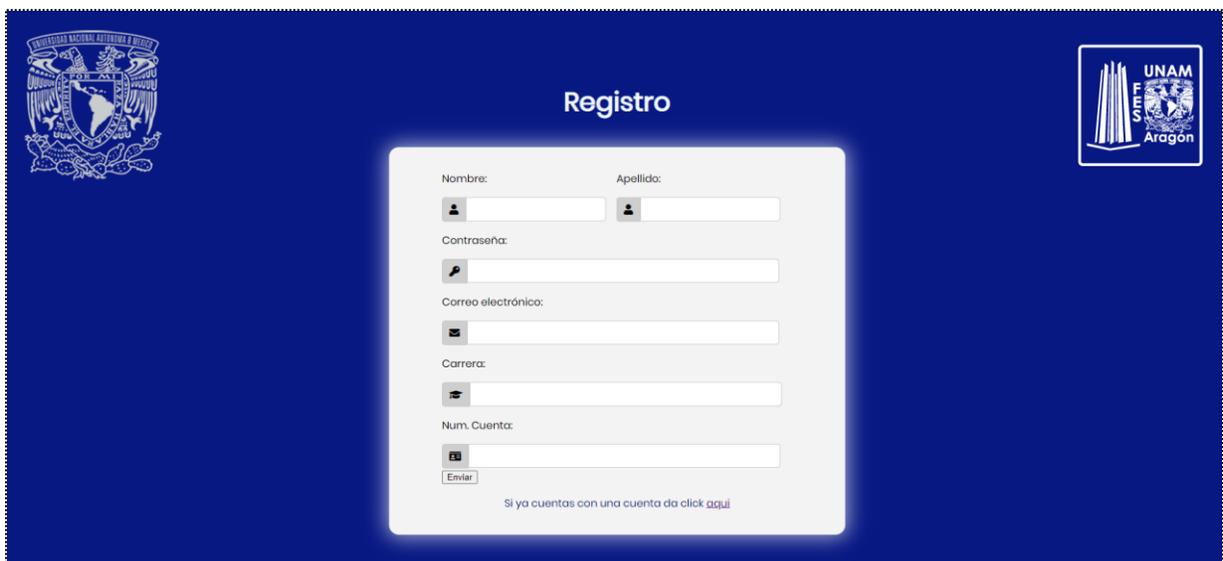


Para separar el nombre y el apellido, de la clase `columna` apuntando a la etiqueta que lo contiene en este caso `<div>` añadimos un `display` que alinea los elementos alineados verticalmente hacia arriba con un tamaño del 50% y un margen a la derecha, a la clase `columna` y campo modificamos su tamaño a 190px

Fragmento del código:

```
.columns > div {  
  display: inline-block;  
  vertical-align: top;  
  width: 50%;  
  margin-right: -5px;  
}  
.columns .campo {  
  width: 190px;  
}
```

Se visualizará en el navegador de la siguiente manera:



The screenshot shows a registration form on a blue background. The form is titled "Registro" and is centered. It contains the following fields and elements:

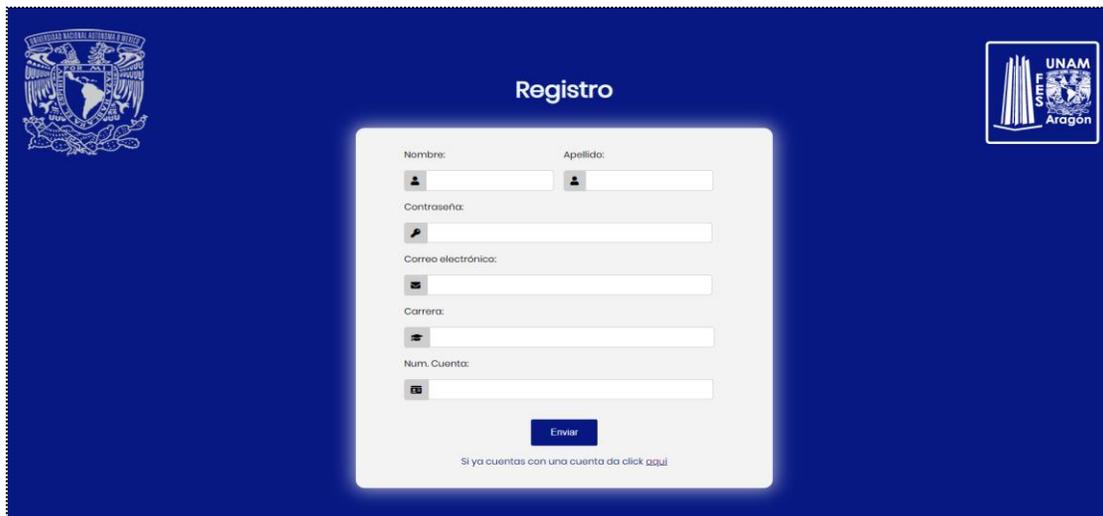
- UNAM logo in the top left corner.
- UNAM logo in the top right corner.
- Form fields: "Nombre:" and "Apellido:" (Name and Surname), "Contraseña:" (Password), "Correo electrónico:" (Email), "Carrera:" (Career), and "Num. Cuenta:" (Account Number).
- An "Enviar" (Send) button.
- A link: "Si ya cuentas con una cuenta da click [aquí](#)" (If you already have an account, click [here](#)).

Posteriormente agregamos el estilo al botón, a la clase botón la alineamos al centro y un margen, la clase button añadimos un color de fondo, añadimos un borde a las esquinas y un color a la fuente, con un padding de 15px hacia arriba y abajo y 32 a los lados alineado al centro y un tamaño de fuente de 16px, y a la clase button al darle clic agregamos un padding y cambiamos su color de fuente y una animación de .3 segundos.

Fragmento del código:

```
.boton{
  text-align: center;
  margin-top: 35px;
}
.button{
  background-color: #081883;
  border: none;
  border-radius: 3px;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
}
.boton input:hover{
  padding: 12px 45px 12px 45px;
  background: #e6e1e1;
  color: #3d3d3d;
  cursor: pointer;
  -webkit-transition: all .3s ease;
  -o-transition: all .3s ease;
  transition: all .3s ease;
}
```

Se visualizará en el navegador de la siguiente manera:



The image shows a registration form titled "Registro" on a dark blue background. The form is centered and contains the following fields:

- Nombre: [input field]
- Apellido: [input field]
- Contraseña: [input field with a password icon]
- Correo electrónico: [input field with an email icon]
- Carrera: [input field with a dropdown arrow]
- Num. Cuenta: [input field with a dropdown arrow]

Below the fields is a blue button labeled "Enviar". At the bottom of the form, there is a link: "Si ya cuentas con una cuenta da click [aquí](#)".

The background features the UNAM logo on the left and the UNAM Aragón logo on the right.

4.5.3. Haciendo el sitio web responsivo (Media queries)

Las "Medias Queries" o en español "consultas de medios" son útiles cuando se desea modificar una página web al tamaño de pantalla de un dispositivo ya sea un móvil, una pantalla o una medida de pantalla específica.

Para eso se usa @media screen and (max-width: la medida de la pantalla a la que se quiere ajustar).

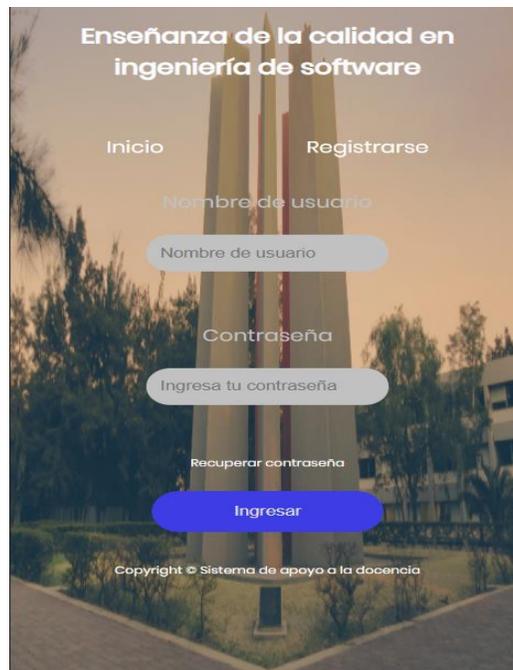
Daremos un tamaño de pantalla de 375px.

Cambiaremos el tamaño a la etiqueta <body> añadiendo un tamaño de fondo de la medida del dispositivo 375px y 667px, la clase login ajustamos el margin-top a 50px y el footer

Fragmento del código:

```
@media screen and (max-width:375px){
  body{
    background: url(../Img/Portada.jpg), no-repeat center top;
    background-size: cover;
    margin: 0;
    padding: 0;
    background-repeat:no-repeat ;
    background-size: 375px 667px ;
  }
  .login{
    margin-top: 50px;
  }
  footer{
    margin-top: 0px;
    font-size: 10px;
    color: #ffffff;
    font-family: 'Poppins', sans-serif;
    text-align: center;
  }
  footer p{
    margin-bottom: 0px;
  }
}
```

Se visualizará en el navegador de la siguiente manera (desde un móvil con resolución específica):



4.6. Desarrollando los efectos de la plataforma con JQuery.

Agregamos un archivo que podemos descargar desde la página oficial de JQuery y un archivo con extensión JS (main.js), estos archivos los cargamos al HTML como se hizo anteriormente con los archivos CSS.

Fragmento del código:

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Apoyo a la docencia</title>
  <link rel="stylesheet" href="css/stylesindex.css">

  <link href="https://fonts.googleapis.com/css?family=Poppins&display=swap" rel="stylesheet">

  <!--IMPORTANDO LIBRERIAJQUERY -->
  <script src="js/jquery-3.4.1.min.js"></script>
  <!--MAIN.JS -->
  <script src="js/main.js"></script>
</head>
```

Posteriormente se escribe el código en el archivo que se creó.

JQuery inicia con:

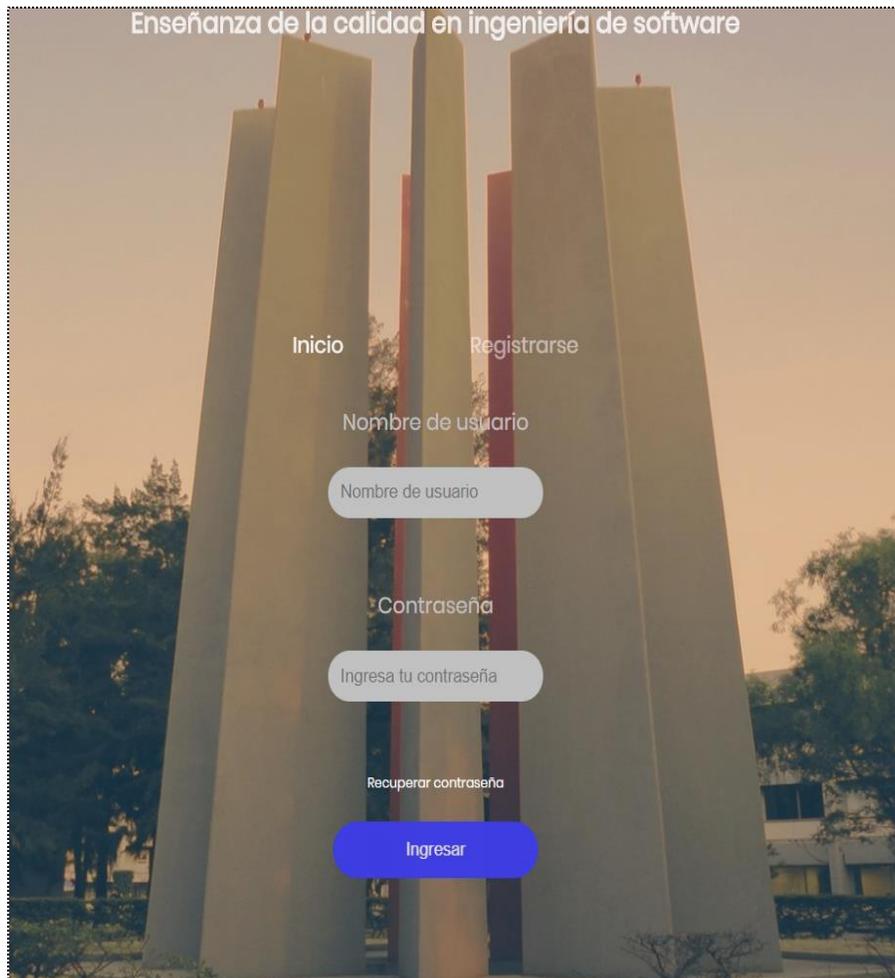
`$(function () {dentro de los corchetes se escribe el código para dar efectos a las clases puestas en HTML});`

Aquí le damos un efecto de animación a la clase título, login y al botón de ingresar, donde disminuimos su opacidad a 0, posteriormente a la misma clase le añadimos `animate` para que aumente gradualmente su opacidad

Fragmento del código:

```
main.js x
D: > User > Documentos > ApoyoDocencia > SitWeb > js > main.js > ...
1  $(function () {
2
3      //Efecto Titulo
4      $('.titulo').each(function(index, element){
5          $(this).css({
6              opacity: 0
7          });
8          $(this).animate({
9              opacity :1
10         },2500 + (index * 1500)); //con el index el tiempo que tarda en hacer la animacion
11     });
12     //Efecto Menu
13     $('.login a').each(function(index, element){
14         $(this).css({
15             opacity: 0
16         });
17         $(this).animate({
18             opacity :1
19         },2000 + (index * 1500)); //con el index el tiempo que tarda en hacer la animacion
20     });
21     //Efecto Menu
22     $('.btn-ingresar input').each(function(index, element){
23         $(this).css({
24             opacity: 0
25         });
26         $(this).animate({
27             opacity :1
28         },2000 + (index * 1500)); //con el index el tiempo que tarda en hacer la animacion
29     });
30
31 });
```

Se visualizará en el navegador de la siguiente manera:



Podemos ver este efecto añadido a las clases mencionadas

4.7. Media Queries de la página de registro

Se empezará modificando el tamaño a las imágenes dentro de las clases *unamlogo* y *feslogo*, a la clase registro se modificará el *padding* hacia arriba y el tamaño de fuente mientras que a la clase formulario solo se cambia el tamaño y el borde de las esquinas a la etiqueta `<form>` se reduce el tamaño del *padding* de la etiqueta `<form>`.

Fragmento del código:

```

@media screen and (max-width:375px){
  .unamLogo {
    float: left;
    width: 60px;
  }
  .fesLogo {
    float: right;
    width: 60px;
  }
  .registro{
    padding-top: 55px;
    font-size: 25px;
    color: ■aliceblue;
    text-align: center;
  }

  .formulario {
    background-color: ■#f3f3f3;
    padding: 0px;
    margin:auto;
    width: 270px;
    border-radius: 15px;
    -webkit-box-shadow: 0px 0px 30px 5px ■rgba(255, 255, 255, 0.616);
    -moz-box-shadow: 0px 0px 30px 5px ■rgba(255, 255, 255, 0.616);
    box-shadow: 0px 0px 30px 5px ■rgba(255, 255, 255, 0.616);
  }
  form{
    padding: 0 20px 0 20px;
  }
}

```

La clase contenedor. campo, columns se redujo el *padding* y el tamaño, al botón se le agrego un *margin* hacia arriba y abajo.

Fragmento del código:

```
.contenedor-campo i {
  border: solid 1px #ccc;
  padding: 4px;
  margin-right: -4px;
  vertical-align: top;
  background-color: #e0e0e0;
  border-radius: 5px 0px 0px 5px;
}

.campo {
  border-radius: 0px 5px 5px 0px;
  border: solid 1px #ccc;
  padding: 4px;
  width: 180px;
}

.columns > div {
  display: inline-block;
  vertical-align: top;
  width: 100%;
}

.columns .campo {
  width: 180px;
}

.button{
  text-align: center;
  margin-top: -15px;
  margin-bottom: 20px;
}

.cuentaRegistrada{
  padding-bottom: 10px;
}
```

Se visualizará en el navegador de la siguiente manera (desde un móvil con resolución específica):

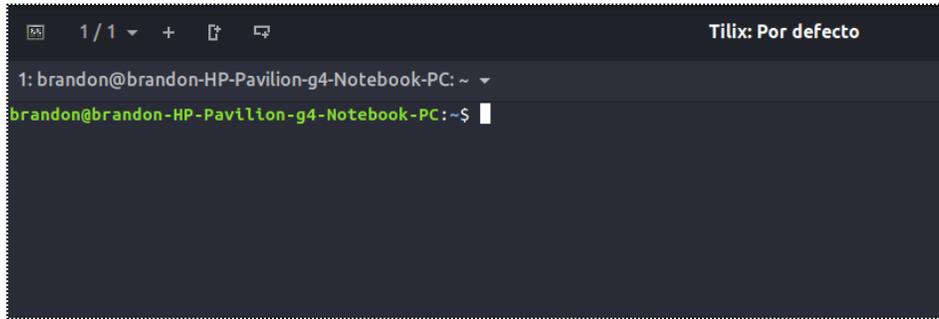


The image shows a mobile registration form with a dark blue background. At the top left is a crest, and at the top right is a logo for 'UNAJA'. The title 'Registro' is centered at the top. The form contains the following fields: 'Nombre:' with a person icon, 'Apellido:' with a person icon, 'Contraseña:' with a key icon, 'Correo electrónico:' with an envelope icon, 'Carrera:' with a graduation cap icon, and 'Num. Cuenta:' with a document icon. Below the fields is a blue 'Enviar' button. At the bottom, there is a link: 'Si ya cuentas con una cuenta da click [aquí](#)'.

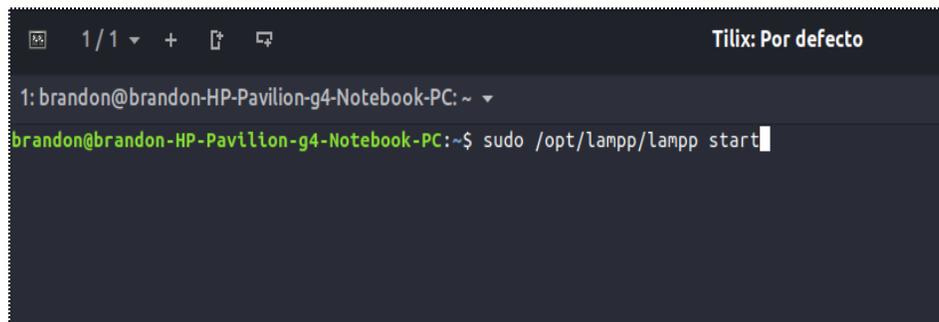
4.8. Desarrollo de la base de datos en XAMPP con MySQL

A continuación, se mostrará el proceso para crear la base de datos con XAMPP y MySQL. Cabe mencionar que este proceso fue realizado en el sistema operativo Ubuntu Budgie 18.04.5 LTS que lo único que cambia con respecto a Windows 10 es la forma en la que se debe de iniciar XAMPP, por lo que se pueden seguir los mismos pasos para realizar en Windows 10.

1. Comenzaremos inicializando XAMPP en Ubuntu, para esto debemos inicializar Tilix (Consola de comandos de Ubuntu) y una vez inicializado escribiremos el comando: "sudo /opt/lampp/lampp start"

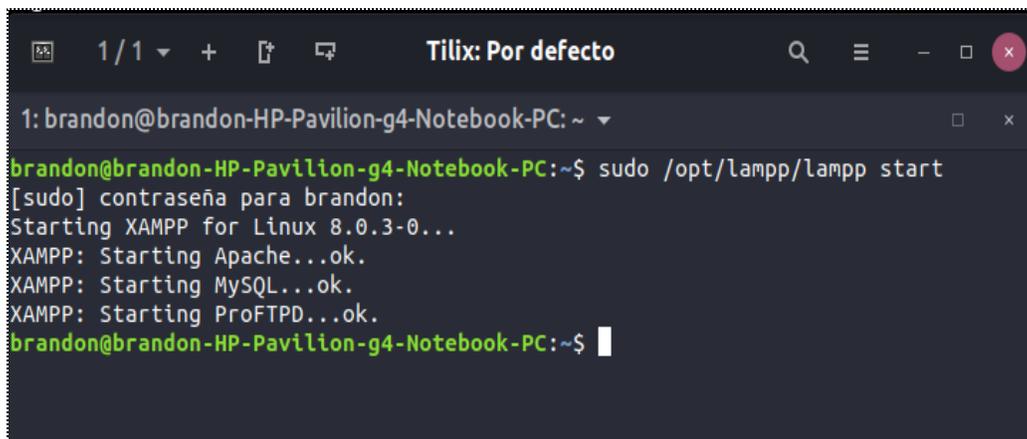


```
1: brandon@brandon-HP-Pavilion-g4-Notebook-PC: ~  
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~$
```



```
1: brandon@brandon-HP-Pavilion-g4-Notebook-PC: ~  
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~$ sudo /opt/lampp/lampp start
```

Seguido de esto nos pedirá la contraseña de nuestro usuario personal, la cual ingresamos, una vez ingresada aparecerá una ventana similar diciendo que los componentes se iniciaron correctamente.



```
1: brandon@brandon-HP-Pavilion-g4-Notebook-PC: ~  
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~$ sudo /opt/lampp/lampp start  
[sudo] contraseña para brandon:  
Starting XAMPP for Linux 8.0.3-0...  
XAMPP: Starting Apache...ok.  
XAMPP: Starting MySQL...ok.  
XAMPP: Starting ProFTPD...ok.  
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~$
```

Si por el contrario aparece una ventana similar mencionando “netstat: orden no encontrada” significa que XAMPP no encuentra el comando NETSTAT.

```

18:12
Tilix: Por defecto
1: brandon@brandon-HP-Pavilion-g4-Notebook-PC: ~ ▾
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~$ sudo /opt/lampp/lampp start
[sudo] contraseña para brandon:
Starting XAMPP for Linux 8.0.3-0...
XAMPP: Starting Apache.../opt/lampp/share/xampp/xampplib: línea 22: netstat: orden no encontrada
/opt/lampp/share/xampp/xampplib: línea 22: netstat: orden no encontrada
ok.
XAMPP: Starting MySQL.../opt/lampp/share/xampp/xampplib: línea 22: netstat: orden no encontrada
ok.
XAMPP: Starting ProFTPD.../opt/lampp/share/xampp/xampplib: línea 22: netstat: orden no encontrada
ok.
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~$ █

```

Por lo que deberemos instalar NET-TOOLS para solucionar este problema. Esto se hace abriendo una consola de Tilix y escribiendo el siguiente comando: “sudo apt install net-tools”.

```

brandon@brandon-HP-Pavilion-g4-Notebook-PC:~$ sudo apt install net-tools
[sudo] contraseña para brandon:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
linux-hwe-5.4-headers-5.4.0-42 linux-hwe-5.4-headers-5.4.0-54
linux-hwe-5.4-headers-5.4.0-56 linux-hwe-5.4-headers-5.4.0-58
linux-hwe-5.4-headers-5.4.0-59 linux-hwe-5.4-headers-5.4.0-60
linux-hwe-5.4-headers-5.4.0-62 linux-hwe-5.4-headers-5.4.0-64
linux-hwe-5.4-headers-5.4.0-65 linux-hwe-5.4-headers-5.4.0-66
linux-hwe-5.4-headers-5.4.0-67 python3-click python3-colorama
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes NUEVOS:
net-tools
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 54 no actualizados.
Se necesita descargar 194 kB de archivos.
Se utilizarán 803 kB de espacio de disco adicional después de esta operación.
Des:1 http://mx.archive.ubuntu.com/ubuntu bionic/main amd64 net-tools amd64 1.60+git20161116.90da8a0-1ubuntu1 [194 kB]
Descargados 194 kB en 1s (323 kB/s)
Seleccionando el paquete net-tools previamente no seleccionado.
(Leyendo la base de datos ... 488330 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../net-tools_1.60+git20161116.90da8a0-1ubuntu1_amd64.deb ...
Desempaquetando net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Configurando net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~$ █

```

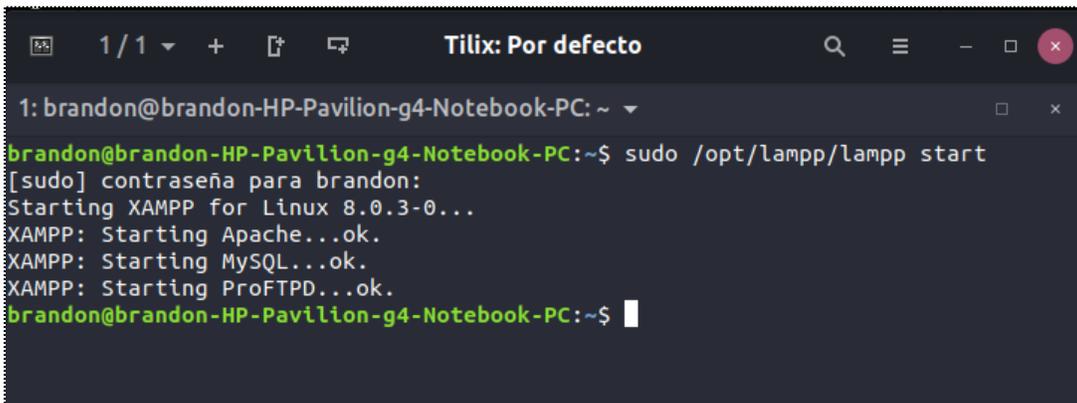
Una vez realizado lo anterior, debemos detener XAMPP para volverlo a inicializar, esto es con el comando de “sudo /opt/lampp/lampp stop”

```

brandon@brandon-HP-Pavilion-g4-Notebook-PC:~$ sudo /opt/lampp/lampp stop
[sudo] contraseña para brandon:
Stopping XAMPP for Linux 8.0.3-0...
XAMPP: Stopping Apache...ok.
XAMPP: Stopping MySQL...ok.
XAMPP: Stopping ProFTPD...ok.
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~$ █

```

Volvemos a iniciar XAMPP y debe visualizarse de la siguiente forma

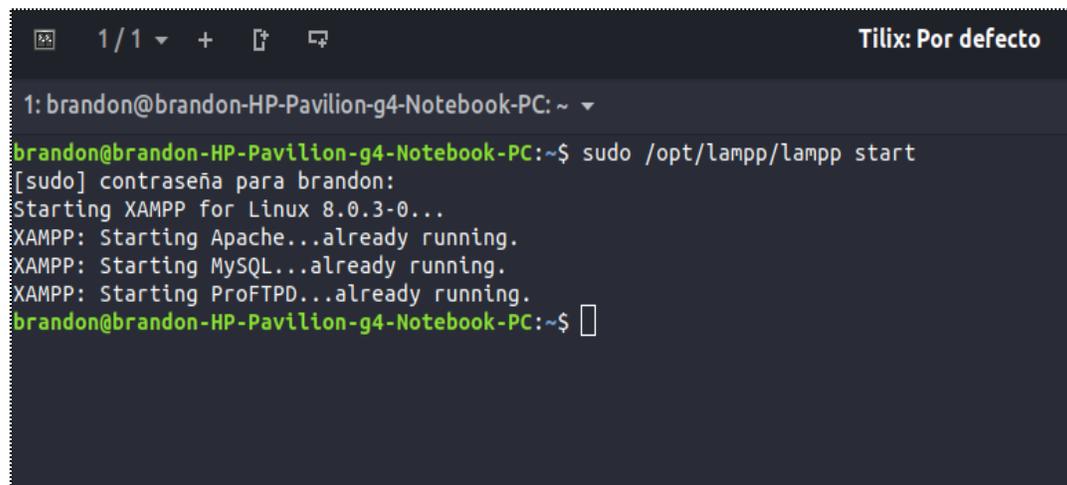


```

1: brandon@brandon-HP-Pavilion-g4-Notebook-PC: ~ ▾
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~$ sudo /opt/lampp/lampp start
[sudo] contraseña para brandon:
Starting XAMPP for Linux 8.0.3-0...
XAMPP: Starting Apache...ok.
XAMPP: Starting MySQL...ok.
XAMPP: Starting ProFTPD...ok.
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~$ █

```

Si XAMPP ya se encontraba inicializado mostrará una la siguiente ventana.



```

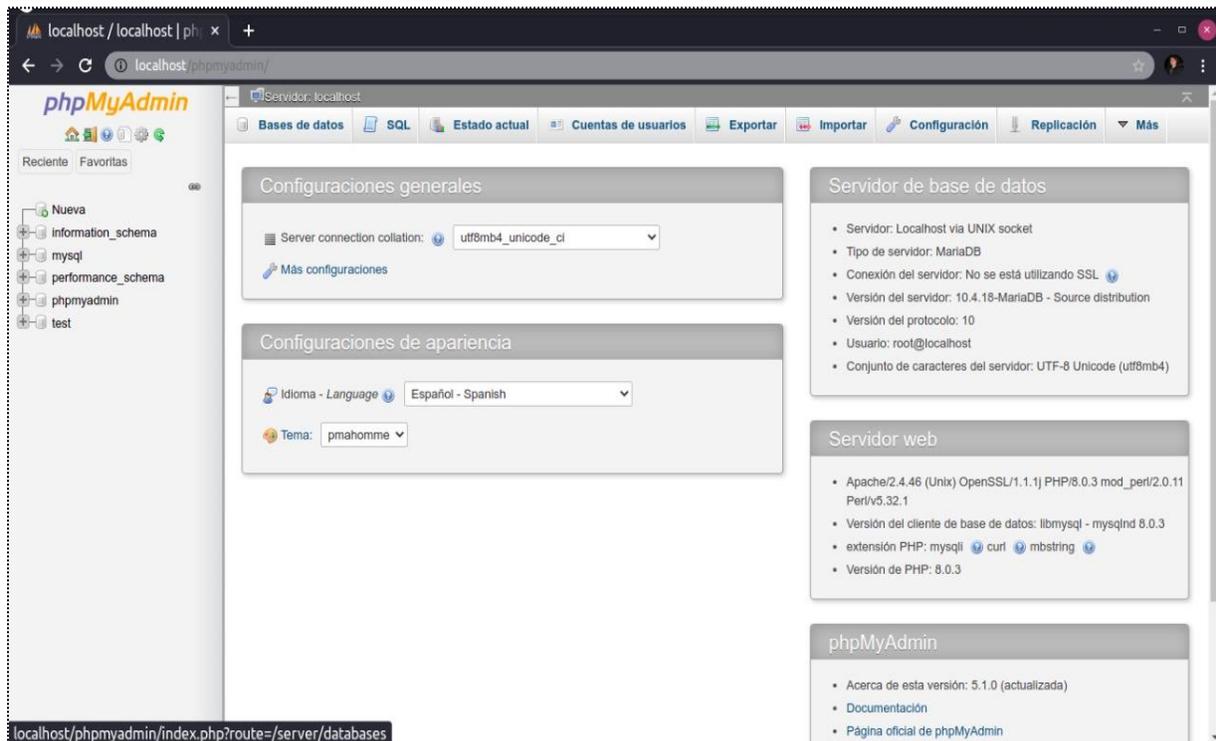
1: brandon@brandon-HP-Pavilion-g4-Notebook-PC: ~ ▾
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~$ sudo /opt/lampp/lampp start
[sudo] contraseña para brandon:
Starting XAMPP for Linux 8.0.3-0...
XAMPP: Starting Apache...already running.
XAMPP: Starting MySQL...already running.
XAMPP: Starting ProFTPD...already running.
brandon@brandon-HP-Pavilion-g4-Notebook-PC:~$ █

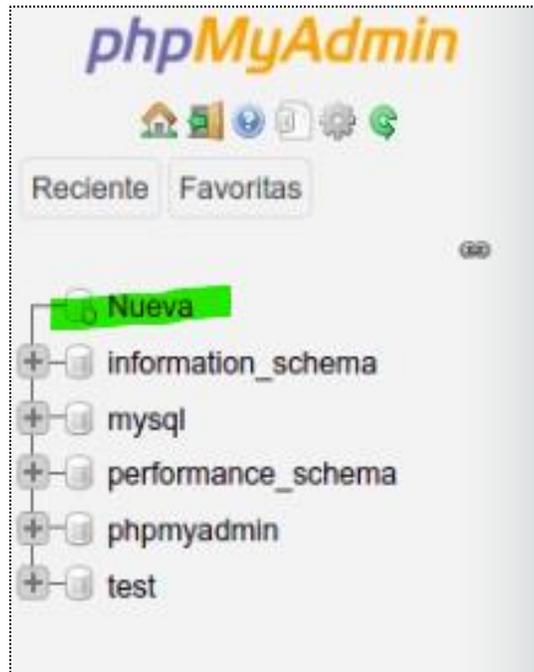
```

2. Una vez iniciado XAMPP en nuestro navegador accederemos a la página principal de XAMPP escribiendo en la url "*localhost/dashboard*" mostrando la siguiente página. Una vez en la ventana principal de XAMPP nos dirigiremos a "phpMyAdmin" situado del lado inferior derecho, mostrándonos la ventana principal de phpMyAdmin



3. El siguiente paso será crear la base de datos para nuestro proyecto, por lo que ya situados en php MyAdmin nos dirigiremos a la opción que dice “Nueva”





4. Mostrará la siguiente ventana y procederemos a nombrar nuestra base de datos, el nombre que usaremos para esta base de datos será “php_registro_usuarios” y en cotejamiento seleccionaremos “utf8_unicode_ci”.



Bases de datos

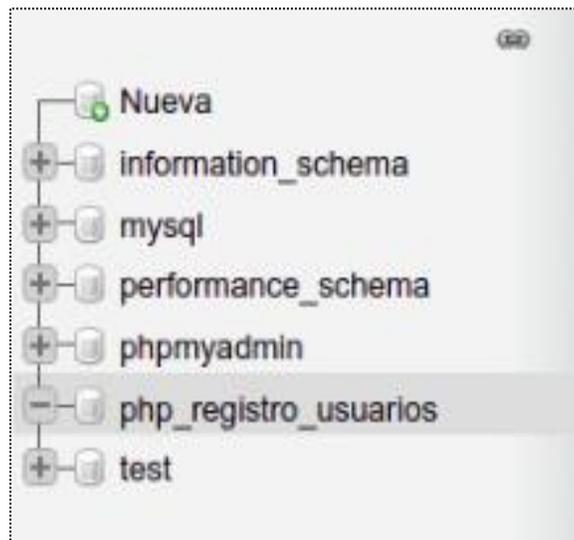
Crear base de datos 

php_registro_usuarios utf8_unicode_ci **Crear**

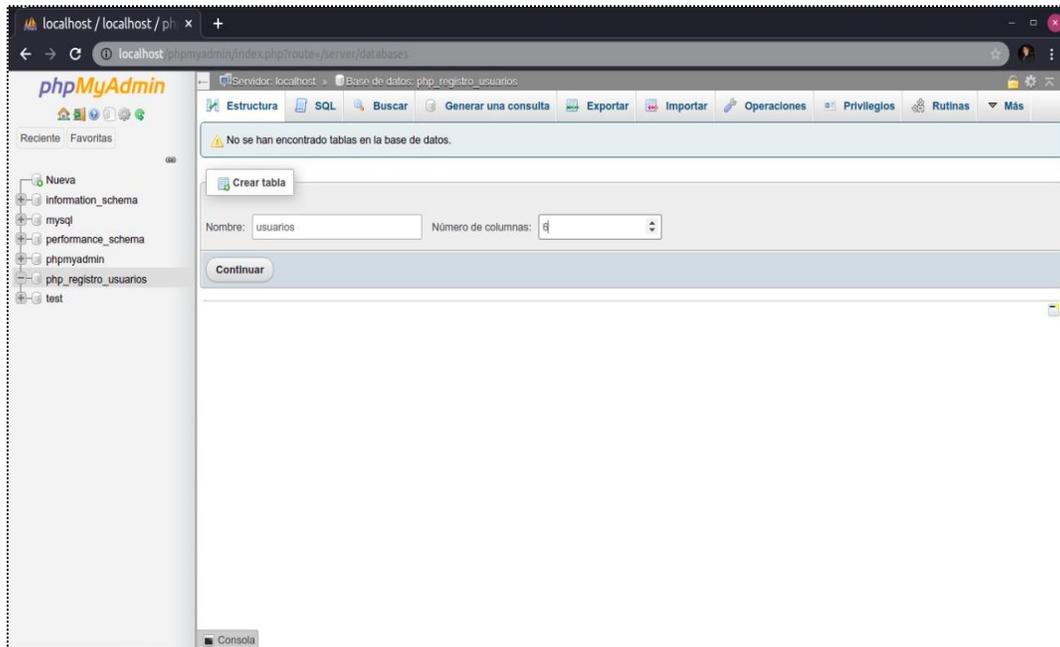
Base de datos	Cotejamiento	Acción
<input type="checkbox"/> information_schema	utf8_general_ci	 Seleccionar privilegios
<input type="checkbox"/> mysql	utf8mb4_general_ci	 Seleccionar privilegios
<input type="checkbox"/> performance_schema	utf8_general_ci	 Seleccionar privilegios
<input type="checkbox"/> phpmyadmin	utf8_bin	 Seleccionar privilegios
<input type="checkbox"/> test	utf8mb4_general_ci	 Seleccionar privilegios

Total: 5

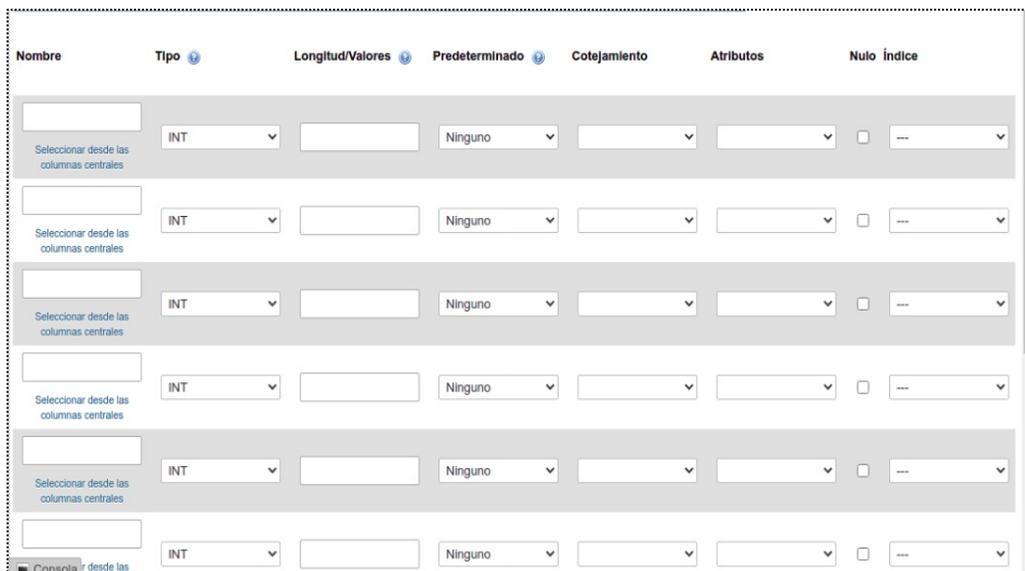
Mostrará un mensaje diciendo que la base de datos ha sido creada correctamente y veremos que está ya aparece de nuestro lado izquierdo.



5. Con la base de datos creada, el siguiente paso es crear la tabla en donde los datos serán almacenados, para esto seleccionaremos la tabla que acabamos de crear "php_registro_usuarios", daremos clic, nos mostrará la siguiente ventana y mostrará la opción que dice "Crear tabla". Nombraremos a la tabla con el nombre de usuarios y le crearemos 7 columnas



Aparecerá la siguiente ventana, aquí nombraremos los campos de la siguiente manera: “id”, “nombres”, “apellidos”, “numeroCuenta”, “carrera”, “email”, “contraseña”. El campo de “id” debe ser autoincrementable, ya que este irá cambiando su numeración conforme se agreguen o quiten usuarios. Todos serán tipo “varchar” a excepción del campo “id” que será de tipo “int”



Se verá de la siguiente manera.

Estructura de tabla		Vista de relaciones								
#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción	
<input type="checkbox"/>	1 id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más	
<input type="checkbox"/>	2 nombres	varchar(80)	utf8_unicode_ci		No	Ninguna			Cambiar Eliminar Más	
<input type="checkbox"/>	3 apellidos	varchar(80)	utf8_unicode_ci		No	Ninguna			Cambiar Eliminar Más	
<input type="checkbox"/>	4 numeroCuenta	varchar(9)	utf8_unicode_ci		No	Ninguna			Cambiar Eliminar Más	
<input type="checkbox"/>	5 carrera	varchar(80)	utf8_unicode_ci		No	Ninguna			Cambiar Eliminar Más	
<input type="checkbox"/>	6 email	varchar(80)	utf8_unicode_ci		No	Ninguna			Cambiar Eliminar Más	
<input type="checkbox"/>	7 contraseña	varchar(80)	utf8_unicode_ci		No	Ninguna			Cambiar Eliminar Más	

Con esto finalizamos la creación de la base de datos, con la respectiva tabla que se irá informando a través del código de PHP incrustado en el HTML.

4.8. Implementación del código PHP en los HTML

Una vez creada la base de datos con su respectiva tabla donde serán informados los datos, el siguiente paso es implementar el código PHP en el HTML para su correcto funcionamiento. A continuación, se mostrarán las líneas de código que se emplearon, así como una breve descripción de su función. Cabe mencionar que se hace una continua reutilización de código y este será explicado o mencionado.

Las clases en donde se ha implementado un código PHP fueron (ordenadas alfabéticamente) las siguientes:

“acercaDe.php”, “apoyo.php”, “database.php”, “index.php”, “login.php”, “logout.php” y “signUp.php”.

Código PHP en la clase “*database.php*”.

El código es el siguiente:

```
database.php X
database.php > ...
1  <!--
2  | Datos de la conexión de PHP
3  -->
4
5  <?php
6
7  $server = 'localhost:3306';
8  $username = 'root';
9  $password = '';
10 $database = 'php_registro_usuarios';
11
12 try {
13     $conn = new PDO("mysql:host=$server;dbname=$database;", $username, $password);
14 } catch (PDOException $e) {
15     die('Connection Failed: ' . $e->getMessage());
16 }
17
18 ?>
```

El código comienza desde la línea número 5. En donde se abre la etiqueta que delimitará que se trabaja con un código php, dicha etiqueta cierra en la línea número 18.

La línea 7 contiene la variable *\$server* que hace referencia a la IP o dominio en donde se encuentra la base de datos de manera local.

La línea 8 contiene la variable *\$username* que hace referencia al usuario XAMPP con el que la base de datos fue creada.

La línea 9 contiene la variable *\$password* que hace referencia a la contraseña utilizada para acceder a XAMPP.

La línea 10 contiene la variable `$database` que hace referencia al nombre de la tabla en la base de datos de XAMPP

De la línea 12 a 16 contiene una sentencia condicional *try/catch* en la cual en el *try* se realizará la conexión con la base de datos y la tabla referenciada, en caso de que ocurra un error entrará en el *catch* enviando un *PDOException* matando a través de la sentencia *die* la conexión y enviando un mensaje al usuario de que la conexión ha fallado.

Nota 1: PDO significa PHP Data Objects, traducido como Objeto de Datos de PHP.

Nota 2: Entiéndase como excepción al comportamiento anómalo que se produce durante la ejecución de un programa.

Código PHP en la clase “logout.php”.

El código es el siguiente:

```
logout.php X
logout.php
1  <?php
2
3  session_start();
4
5  session_unset();
6
7  session_destroy();
8
9  header('Location: /paginatesis/login.php');
10
11 ?>
```

El código comienza con en la línea 1 donde abriendo las etiquetas que delimitan el código en PHP, dicha etiqueta cierra en la línea número 11

La línea 3 contiene un `session_start ()` este se encarga de crear una sesión o reanudar la sesión actual, esto mediante una petición GET o POST, o mediante una cookie.

La línea 5 contiene un `session_unset ()` que se encarga de liberar o limpiar el espacio que está consumiendo la variable de sesión.

La línea 7 contiene un `session_destroy ()` este se encarga de destruir la variable de sesión creada.

La línea 9 contiene un `header ()` este se encarga de redireccionar las páginas a una localización establecida antes de que estas carguen. En este caso se redirige a la página de `login.php` para evitar que los usuarios accedan a las distintas páginas de la plataforma sin haber iniciado sesión anteriormente.

Código PHP utilizado en las clases “`acercaDe.php`”, “`apoyo.php`”, “`index.php`” y “`signUp.php`”

El código es el siguiente:

```
1 <?php
2 session_start();
3
4 require 'database.php';
5
6 if (isset($_SESSION['user_id'])) {
7     $records = $conn->prepare('SELECT id, password, nombres, numeroCuenta FROM usuarios WHERE id = :id');
8     $records->bindParam(':id', $_SESSION['user_id']);
9     $records->execute();
10    $results = $records->fetch(PDO::FETCH_ASSOC);
11
12    $user = null;
13
14    if (count($results) > 0) {
15        $user = $results;
16    }
17 }
18
19 ?>
```

El código comienza con en la línea 1 donde abriendo las etiquetas que delimitan el código en php, dicha etiqueta cierra en la línea número 19

La línea 2 contiene un `session_start ()` encargado de crear una nueva sesión o reanudarla

La línea 4 contiene un *requiere* `'database.php'` esto se encargará de llamar a la clase mencionada.

La línea 6 contiene una condicional *if*, esta condición se cierra en la línea 17. Dicha línea contiene un `isset($_SESSION['user_id'])` cuya funcionalidad es evaluar si una variable está definida, en este caso será la variable `$_SESSION['user_id']`.

La línea 7 realiza una conexión a la tabla *'usuarios'* en donde se encuentran los datos del usuario, realizando una consulta, en donde selecciona el *id, password, nombres, número de cuenta*, de la tabla *usuarios*, donde el id es igual a el valor de la sesión definido en la línea 8

Una vez obtenido el valor de la sesión en la línea 8, la línea 9 hace una ejecución, para posteriormente en la línea 10 se obtienen los resultados y se almacenan en la variable `$results`

La línea 14 contiene una condición *if* comprueba que la variable `$results` es mayor que 0 para verificar que no esté vacío. Si es el caso enviará el valor de `$results` a la variable `$user` definida en la línea 12.

Código PHP en la clase “*acercaDe.php*”, “*apoyo.php*”, “*index.php*”, “*login.php*” y “*signUp.php*”.

El código es el siguiente:

```
<!--VALIDACIONES PARA QUE REDIRECCIONE AL SIGNUP SI LA SESION NO SE ENCUENTRA INICIADA----->

<?php if (!empty($user)) : ?>
<?php else : ?>
|   <meta http-equiv="refresh" content="0;url=login.php">
<?php endif; ?>
```

```
<?php if (!empty($user)) : ?>
|   <meta http-equiv="refresh" content="0;url=index.php">
<?php else : ?>
|   <? die(); ?>
<?php endif; ?>
```

Es una condición en donde se verifica si la variable *\$user* NO está vacía. Si la variable *\$user* está vacía nos redirigirá a la sección de login para iniciar sesión. En caso contrario de que la sesión este iniciada y la variable *\$user* esté informada, nos dirigirá a la sección de *index*. La última línea se encarga de informar que terminó la condición *if/else*

Código PHP en la clase “*signUp.php*”.

El código es el siguiente:

```

signUp.php > ...
1  <?php
2  require 'database.php';
3
4  $message = '';
5
6  if [
7      !empty($_POST['nombres']) &&
8      !empty($_POST['apellidoPaterno']) &&
9      !empty($_POST['apellidoMaterno']) &&
10     !empty($_POST['numeroCuenta']) &&
11     !empty($_POST['carrera']) &&
12     !empty($_POST['email']) &&
13     !empty($_POST['password'])
14 ] {
15     $sql = "INSERT INTO usuarios (nombres, apellidoPaterno,
16     apellidoMaterno, numeroCuenta, carrera, email, password)
17     VALUES (:nombres, :apellidoPaterno, :apellidoMaterno,
18     :numeroCuenta, :carrera, :email, :password)";
19
20     $stmt = $conn->prepare($sql);
21
22     $stmt->bindParam(':nombres', $_POST['nombres']);
23     $stmt->bindParam(':apellidoPaterno', $_POST['apellidoPaterno']);
24     $stmt->bindParam(':apellidoMaterno', $_POST['apellidoMaterno']);
25     $stmt->bindParam(':numeroCuenta', $_POST['numeroCuenta']);
26     $stmt->bindParam(':carrera', $_POST['carrera']);
27     $stmt->bindParam(':email', $_POST['email']);
28
29     $password = password_hash($_POST['password'], PASSWORD_BCRYPT);
30
31     $stmt->bindParam(':password', $password);
32
33     if ($stmt->execute()) {
34         $message = 'El usuario se ha creado exitosamente';
35     } else {
36         $message = 'Lo sentimos, hubo un error al crear su usuario. Intente mas tarde';
37     }
38 }
39 }
40
41 >>

```

En esta sección de *signUp* (*inscribirse/registrarse*), funciona en conjunto con el formulario HTML (en el mismo documento) para crear un usuario.

El código comienza con en la línea 1 donde abriendo las etiquetas que delimitan el código en php, dicha etiqueta cierra en la línea número 41

La línea 2 contiene un *requiere* `'database.php'` esto se encargará de llamar a la clase mencionada.

La línea 4 crea la variable `$message`, esta variable aun no muestra ningún texto, ya que este será agregado más adelante.

De la línea 6 a la 14 contiene una condicional *if* en donde verifica que los campos: `'nombres'`, `'apellidoPaterno'`, `'apellidoMaterno'`, `'numeroCuenta'`, `'carrera'`, `'email'`, `'password'` estén informados. Si los campos vienen informados se hará un *INSERT* a la tabla usuarios.

De la línea 15 a la 18 se crea la variable `$sql`, esta variable contiene el *INSERT* que se hará a la tabla usuarios.

La línea 20 prepara la conexión a la base de datos para hacer el *INSERT* a la tabla de usuarios y ser reutilizada en las líneas siguientes.

De la línea 22 a la 27 inserta los valores recibidos en el formulario a la tabla de usuarios.

La línea 29 realiza un encriptamiento al campo `password`, para que al ser escrita en la base de datos esta no muestre la contraseña del usuario.

La línea 31 inserta el valor de la contraseña recibida (ya encriptada) a la tabla de usuarios.

De la línea 33 a la 38 es una condicional *if/else*, en donde si se ejecutó correctamente la escritura de datos en la tabla, enviará la variable `message` con el texto: *“El usuario se ha creado exitosamente”* en caso de que la escritura de datos haya sido insatisfactoria se enviará la variable `message` con el texto: *“Lo sentimos, hubo un error al crear su usuario. Intente más tarde”*

Código PHP en la clase “login.php” y “signUp.php”

```
<?php if (!empty($message)) : ?>
    <p> <?= $message ?></p>
<?php endif; ?>
```

Este código PHP permite a las páginas de *login.php* y *signUp.php* mostrar un mensaje ya sea de error o de éxito en la parte superior de la página.

Código PHP en la clase “login.php”.

El código es el siguiente:

```
login.php > ...
1  <?php
2
3  session_start();
4
5  if (isset($_SESSION['user_id'])) {
6      header('Location: /paginatesis/index.php');
7  }
8  require 'database.php';
9
10 if (
11     !empty($_POST['numeroCuenta']) &&
12     !empty($_POST['password'])
13 ) {
14
15     $records = $conn->prepare('SELECT id, numeroCuenta, password FROM usuarios WHERE numeroCuenta = :numeroCuenta');
16     $records->bindParam(':numeroCuenta', $_POST['numeroCuenta']);
17     $records->execute();
18     $results = $records->fetch(PDO::FETCH_ASSOC);
19
20     $message = '';
21
22     if ($results && count($results) > 0 && password_verify($_POST["password"], $results["password"])) {
23         // if (count($results) > 0 && password_verify($_POST["password"], $results["password"])) {
24         $_SESSION['user_id'] = $results['id'];
25         header("Location: /paginatesis/index.php");
26     } else {
27         $message = 'Error: El usuario no existe, intente nuevamente';
28     }
29 }
30
31 ?>
```

En esta sección de *login* (*iniciar sesión/ingresar*), funciona en conjunto con el formulario HTML (en el mismo documento) para ingresar.

El código comienza con en la línea 1 donde abriendo las etiquetas que delimitan el código en php, dicha etiqueta cierra en la línea número 31

La línea 3 contiene un *session_start ()* encargado de crear una sesión o reanudar la sesión actual.

De la línea 5 a la 7 hay una condicional if, en la cual si el usuario está registrado y con su sesión iniciada automáticamente lo reubicará al índice de la página.

La línea 8 contiene un *requiere 'database.php'* esto se encargará de llamar a la clase mencionada.

De la línea 10 a la 13 se inicia una condición if, la cual verifica si el número de cuenta y password pedidos en el formulario están informados.

La línea 15 realiza una consulta a la tabla usuarios, en donde se obtendrá el número de cuenta para así en la línea 16 vincularlo al número de cuenta obtenido en el formulario HTML.

La línea 17 ejecutará la consulta y el valor obtenido se guardará en la variable *\$results*.

La línea 20 crea la variable *\$message*, esta variable aun no muestra ningún texto, ya que este será agregado más adelante.

De la línea 22 a la 28 se inicia una condición if, la cual compara las contraseñas, la contraseña de la base de datos, con la contraseña que el usuario ingresa en el formulario. Si las contraseñas coinciden se crea una variable de sesión y el usuario accede satisfactoriamente al índice. Si las contraseñas no coinciden manda un mensaje de error.

4.9. Actualización de la página (HTML, CSS y JS)

Se realizó una pequeña actualización a la página en HTML, en la cual hubo arreglos a la parte estética. No afectando el funcionamiento del código PHP explicado en capítulos anteriores, ni a la tabla “*usuarios*” en la base de datos.

Clase *login.php* (Inicio de sesión)

Se incluyen las fuentes que pertenecen a la página, los iconos dentro de las etiquetas link, al igual que la hoja de estilos css. En la cabecera del sitio se ha incluido el correo principal de la plataforma. Mientras tanto el cuerpo del documento agregamos una sección que contiene el formulario con el nombre de la plataforma, al igual que inputs donde el usuario coloca su número de cuenta y contraseña, en caso de no tener una cuenta, podrá dirigirse a crear una dentro de las etiquetas `<a>Regístrate aquí `

Fragmento del código:

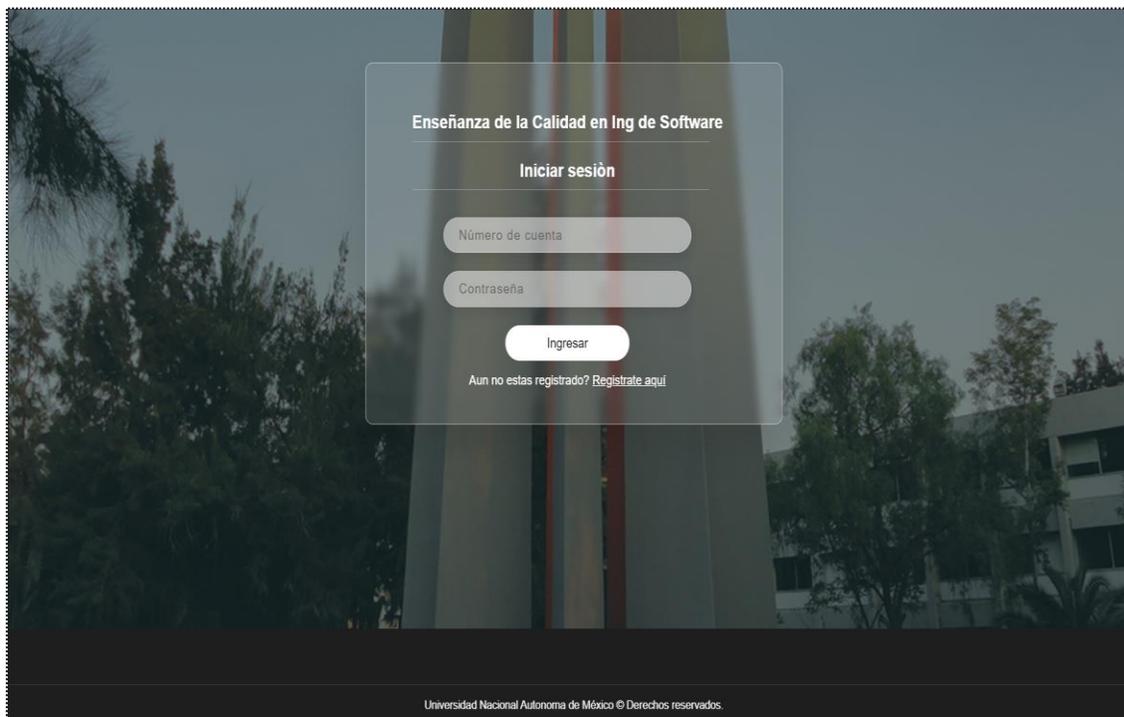
```
login.php x
paginaTesis > login.php > html
86 <!--Cuerpo del documento-->
87 <main>
88 <article id="home">
89
90 <!-- Sección Uno ---->
91 <section>
92 <div id="seccionUno" class="container">
93 <div id="formulario">
94 <form class="form" action="login.php" method="POST">
95
96 <div class="tituloForm">
97 <h2>Enseñanza de la Calidad en Ing de Software</h2>
98 <h2>Iniciar sesión</h2>
99 </div>
100
101
102 <!-- <Label for="suNumeroCuenta">Número de cuenta: </Label> -->
103 <div class="inputBox">
104 <input type="text" placeholder="Número de cuenta" name="numeroCuenta" id="suNumeroCuenta" required>
105 </div>
106
107 <!-- <Label for="suPassword">Contraseña: </Label> -->
108 <div class="inputBox">
109 <input type="password" placeholder="Contraseña" name="password" id="suNumeroCuenta" required>
110 </div>
111
112 <div class="inputButtonEnviar">
113 <input type="submit" value="Ingresar">
114 </div>
115
116 <small class="olvidado">Aun no estas registrado? <a href="signUp.php" target="#">Regístrate aquí</a>
117 </small>
118 </form>
119 </div>
120 </div>
121 </section>
122 </article>
123 </main>
124
125 <!-- Pie de pagina---->
126 <footer>
127 <div id="copyright">
128 <small> Universidad Nacional Autonoma de México &copy; Derechos reservados. </small>
129 </div>
130 </footer>
131 </body>
132
133 </html>
```

Se ha incluido un footer en todas las clases, este footer contiene el pie de página del sitio donde se ha incluido el nombre de la universidad junto a un copyright.

Fragmento del código:

```
<!-- Pie de pagina---->
<footer>
  <div id="copyright">
    <small> Universidad Nacional Autonoma de México &copy; Derechos reservados. </small>
  </div>
</footer>
</body>
</html>
```

En el navegador se visualiza de la siguiente manera:



Clase `signUp.php` (Página de registro)

Se han actualizado las fuentes de la página, los iconos utilizados, al igual que la hoja de estilos CSS. En la cabecera del sitio se ha incluido el correo principal de la plataforma, y este se activará en pantallas de dispositivos portátiles.

Fragmento del código:

```

signUp.php X
paginaTesis > signUp.php > ...
71 <head>
72 <meta charset="UTF-8">
73 <meta name="viewport" content="width=device-width, initial-scale=1.0">
74 <meta http-equiv="X-UA-Compatible" content="ie=edge">
75 <title>Calidad Ing Software</title>
76
77 <!--Estilos CSS-->
78 <link rel="stylesheet" href="css/estilos.css">
79
80 <!--Estilo de Fuente-->
81 <link href="https://fonts.googleapis.com/css?family=Ibarra+Real+Nova&display=swap" rel="stylesheet">
82
83 <!-- IconFile --->
84 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@5.9.0/css/all.css">
85
86 <!--Favicon-->
87 <link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
88 <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
89 <link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
90 <link rel="manifest" href="/site.webmanifest">
91 <meta name="msapplication-TileColor" content="#da532c">
92 <meta name="theme-color" content="#ffffff">
93 <!-- Fin de FavIcon-->
94 </head>
95
96 <body>
97
98 <!-- Cabecera-->
99 <div class="top">
100 <ul class="container">
101 <li><i class="far fa-envelope"></i></i> plataformaweb.fesaragon@gmail.com</li>
102 </ul>
103 </div>
104
105 <!-- PHP --->
106 <?php if (!empty($message)) : ?>
107 <p> <?= $message ?></p>
108 <?php endif; ?>
109

```

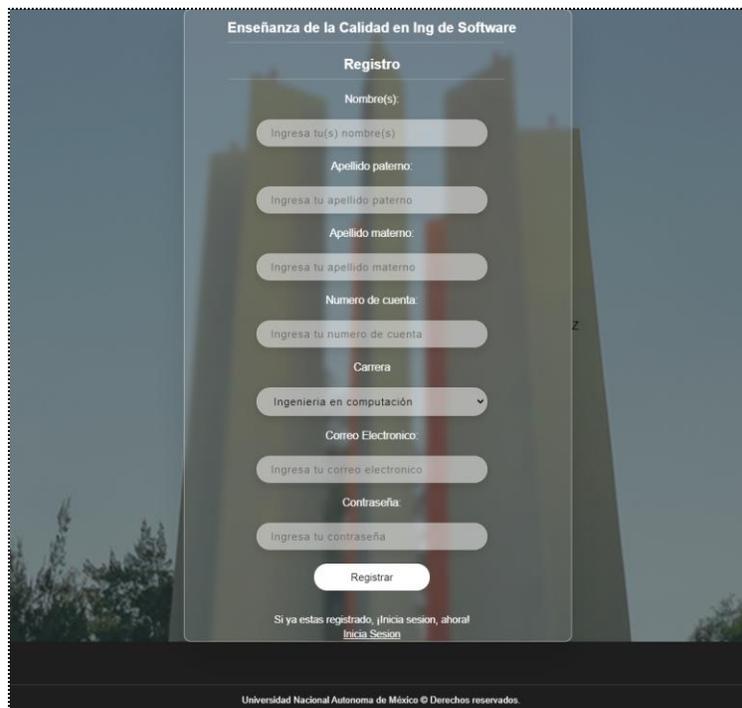

Clase `formulario.js`

Esta clase verifica que el nombre y el apellido no estén vacíos o incluyan caracteres.

Fragmento del código:

```
formulario.js X
paginaTesis > formulario.js > ...
1  const nombre = document.querySelector("#nombre");
2  const pattern = new RegExp("^[A-Z]+$", "i");
3
4  //Leemos el input
5  nombre.addEventListener("input", (e) => {
6    if (e.target.value === "" || !pattern.test(e.target.value)) {
7      console.log("No puedes registrar");
8    }
9  });
10
```

En el navegador se visualiza de la siguiente manera:



The screenshot shows a registration form with the following fields and elements:

- Registro** (Title)
- Nombre(s):** Input field with placeholder "Ingresar tu(s) nombre(s)".
- Apellido paterno:** Input field with placeholder "Ingresar tu apellido paterno".
- Apellido materno:** Input field with placeholder "Ingresar tu apellido materno".
- Numero de cuenta:** Input field with placeholder "Ingresar tu numero de cuenta".
- Carrera:** Dropdown menu with "Ingeniería en computación" selected.
- Correo Electronico:** Input field with placeholder "Ingresar tu correo electronico".
- Contraseña:** Input field with placeholder "Ingresar tu contraseña".
- Registrar** (Submit button)
- Si ya estas registrado, ¡inicia sesion, ahora!** (Link: [Inicia Sesion](#))
- Universidad Nacional Autonoma de México © Derechos reservados.** (Footer)

Clase `index.php` (Página principal)

Se incluyen las fuentes que pertenecen a la página, los iconos dentro de las etiquetas `link`, al igual que la hoja de estilos CSS.

En la cabecera del sitio incluimos el correo principal de la plataforma, la cual se muestra cuando la página disminuye su resolución para dispositivos móviles.

Fragmento del código:

```

index.php X
paginaTesis > index.php > ...
28
29 <!DOCTYPE html>
30 <html Lang="es">
31
32 <head>
33 <meta charset="UTF-8">
34 <meta name="viewport" content="width=device-width, initial-scale=1.0">
35 <meta http-equiv="X-UA-Compatible" content="ie=edge">
36 <title>Enseñanza de la materia de calidad</title>
37
38 <!--Estilos CSS-->
39 <link rel="stylesheet" href="css/estilos.css">
40
41 <!--Estilo de Fuente-->
42 <link href="https://fonts.googleapis.com/css?family=Ibarra+Real+Nova&display=swap" rel="stylesheet">
43
44 <!-- IconFile --->
45 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.9.0/css/all.css">
46
47 <!--Favicon-->
48 <link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
49 <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
50 <link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
51 <link rel="manifest" href="/site.webmanifest">
52 <meta name="msapplication-TileColor" content="#da532d">
53 <meta name="theme-color" content="#ffffff">
54 <!-- Fin de Favicon-->
55 </head>
56
57 <body>
58

```

En la etiqueta `header`, se ha agregado una barra de navegación, esta contiene el logotipo de la facultad, iconos del `link` de navegación como `home`, `material de apoyo` y `acerca de`, al igual del usuario que ingresó sesión.

Se ha añadido una nueva sección que incluye el objetivo de la plataforma junto con un link para realizar una pre evaluación dentro de las etiquetas `<a>Realizar pre-evaluación`, además de una imagen referente a esta evaluación para lograr la meta.

Fragmento del código:

```
index.php X
paginaTesis > index.php > html > body
102
103 <!-- Sección Dos ---->
104 <section>
105     <div id="seccionDos" class="container">
106         <div>
107             <h2>Nuestro objetivo:</h2>
108             <p> Es brindar a los estudiantes de ingeniería una guía
109                 práctica y confiable para la materia de "Calidad"
110                 impartida en la FES Aragon, así como hacer de este
111                 aprendizaje algo más entretenido.</p>
112
113             <a class="btn" href="apoyo.php"> Realizar pre-evaluación</a>
114         </div>
115         
116     </div>
117 </section>
118
```

Se ha adicionado una nueva sección que contiene un párrafo de introducción a la siguiente sección.

Fragmento del código:

```
index.php X
paginaTesis > index.php > html > body
120 <!-- Sección Tres---->
121 <section>
122
123     <div id="seccionTres" class="container">
124         <p>APRENDE SOBRE QUE ES LA CALIDAD <br>
125             Y COMO SE LLEVA A CABO EN LA ING DE SOFTWARE
126         </p>
127     </div>
128 </section>
```

Se ha incluido una nueva sección que contiene los capítulos de forma ordenada para descargar en PDF, que abarcaran información para realizar un examen dentro de la plataforma

Fragmento del código:

```

index.php X
paginaTesis > index.php > html > body
130 <!-- Seccion Cuatro-->
131 <section>
132 <h2>Capitulos</h2>
133 <p>Puedes leer o descargar los capitulos de esta tesis para leerlos de forma individual segun sea tu interes</p>
134
135 <div id="seccionCuatro" class="container">
136 <!--Iconos de FotnAwesome-->
137
138 <div class="col">
139
140 <a href="download/CalidadSoftware.pdf" download="CalidadSoftware.pdf"><i class="fas fa-file-pdf"></i>
141 <h3>Control de calidad en la Ingenieria de Software</h3>
142 </a>
143 </div>
144
145 <div class="col">
146
147 <!-- CAMBIAR ARCHIVOS -->
148 <a href="download/NormasIsoCalidad.pdf" download="NormasIsoCalidad.pdf"><i class="fas fa-file-pdf"></i>
149 <h3>Normas ISO en la calidad de un Software</h3>
150 </a>
151 </div>
152
153 <div class="col">
154 <a href="download/Medición de la calidad de software.pdf" download="Medición de la calidad de software.pdf"><i class="fas fa-file-pdf"></i>
155 <h3>Medición de la calidad de software: Requerimiento de tiempo</h3>
156 </a>
157 </div>
158
159 <div class="col">
160 <a href="download/CalidadSoftware.pdf" download="CalidadSoftware.pdf"><i class="fas fa-file-pdf"></i>
161 <h3>Desarrollo de la página web</h3>
162 </a>
163 </div>
164 </div>
165 </section>
166

```

Se ha agregado la sección final que contiene el material didáctico, esta es un resumen de los PDF que funciona para complementar y repasar con la información anexada anteriormente mencionada.

Fragmento del código:

```

index.php X
paginaTesis > index.php > html > body
167     <!-- Seccion Cinco---->
168
169     <section>
170         <div id="seccionCinco" class="container">
171             <div>
172                 <h2>Material didactico</h2>
173                 <p>Revisa el material didactico para despues presentar una pequeña prueba que evalúe tus conocimientos</p>
174             </div>
175             <a class="btn" href="apoyo.php">Acceder</a>
176         </div>
177     </section>
178 </article>
179 </main>
180
181

```

Se ha incluido un footer que indica el pie de página del sitio que incluye logos para la pestaña principal, de apoyo y acerca de, además del nombre de la universidad.

Fragmento del código:

```

index.php X
paginaTesis > index.php > html > body
181
182 <!-- Pie de pagina---->
183 <footer>
184
185     <div id="pieDePagina" class="container">
186         <a class="tooltip" href="index.php" > <i class="fas fa-home"></i> <span class="tooltip-box">Ir a inicio</span></a>
187         <a class="tooltip" href="apoyo.php" > <i class="fas fa-book"></i> <span class="tooltip-box">Apoyo</span> </a>
188         <a class="tooltip" href="acercaDe.php" > <i class="fas fa-question-circle"> </i> <span class="tooltip-box">Acerca de</span> </a>
189         <a class="tooltip" href="https://forms.gle/jhXUZ4BZsjZopi7F6" > <i class="fas fa-brain"> </i> <span class="tooltip-box">Realizar Test</span> </a>
190
191     </div>
192     <div id="copyright">
193         <small>Universidad Nacional Autonoma de México &copy; Derechos reservados. </small>
194     </div>
195 </footer>
196

```

La siguiente sentencia de JavaScript actúa sobre el icono navbar-toggle, haciéndolo funcionar como un botón que despliega un menú. Este botón se mostrará en pantallas de dispositivos móviles.

Fragmento del código:

```
index.php X
paginaTesis > index.php > html > body
196
197 <script>
198 // Comportamientos para el menú de navegación.
199 let mainNav = document.getElementById('main-nav');
200 let navbarToggle = document.getElementById('navbar-toggle');
201 navbarToggle.addEventListener('click', function() {
202
203     if (this.classList.contains('active')) {
204         mainNav.style.display = "none";
205         this.classList.remove('active');
206     } else {
207         mainNav.style.display = "flex";
208         this.classList.add('active');
209     }
210 });
```

Se ha agregado una función que recorre las letras mostradas al inicio de la plataforma para dar una animación como si estas se estuvieran escribiendo. Al igual que un último efecto que al dar scroll la imagen de algunas secciones dieran un efecto como si se moviera con la página, definido por un tiempo y velocidad

Fragmento del código:

```
212 //Animacion de texto del area 1
213
214 function typeEffect(element, speed) {
215     var text = element.innerHTML;
216     element.innerHTML = "";
217
218     var i = 0;
219     var timer = setInterval(function() {
220         if (i < text.length) {
221             element.append(text.charAt(i));
222             i++;
223         } else {
224             clearInterval(timer);
225         }
226     }, speed);
227 }
228
229 // application
230 var speed = 75;
231 var h1 = document.querySelector('h1');
232 var p = document.querySelector('p');
233 var delay = h1.innerHTML.length * speed + speed;
234
235 // type affect to header
236 typeEffect(h1, speed);
237
238
239 // type affect to body
240 setTimeout(function() {
241     p.style.display = "inline-block";
242     typeEffect(p, speed);
243 }, delay);
244
245 //Fin de la animacion de texto del area 1
246 </script>
247 </body>
248
249 </html>
```

En el navegador la página "index.php" se visualiza de la siguiente manera:



APRENDE SOBRE QUE ES LA CALIDAD Y COMO SE LLEVA A CABO EN LA ING DE SOFTWARE

Capitulos

Puedes leer o descargar los capitulos de esta tesis para leerlos de forma individual segun sea tu interes



[Control de calidad en la Ingeniería de Software](#)



[Normas ISO en la calidad de un Software](#)



[Medición de la calidad de software: Requerimiento de tiempo](#)



[Desarrollo de la página web](#)

Material didactico

Revisa el material didactico para despues presentar una pequeña prueba que evalúe tus conocimientos

[Acceder](#)



Clase apoyo.php (Página secundaria)

Se ha encapsulado en una *div* con la clase *container* el contenido del *header* que conlleva el título de la página, posteriormente se ha agregado dentro de un artículo una sección que contiene toda la información resumida acerca de las normas en la calidad del software, su medición

Fragmento del código:

```

apoyo.php X
paginaTesis > apoyo.php > ...
89
90 <!--Cuerpo del documento-->
91 <main>
92 <header>
93 <div class="container">
94 <h1>Material didactico para apoyo a la calidad</h1>
95 </div>
96 </header>
97 <article id="pages" class="container">
98 <section>
99 <h2>Control de calidad en la ingeniería de software</h2>
100 <p>Quizá para muchos estudiantes el aspecto de la calidad en el desarrollo de software es algo a lo que a veces no se le da
101
102 </p> </br> </br>Mientras que Pressman menciona que:</br></br>
103
104 «El software de computadora es el producto que construyen los programadores profesionales y al que después le dan mante
105
106 « El software es la interfaz y la lógica abstracta que vincula al ser humano con una tecnología tangible y/o otro softw
107 </br>
108 «En computación, el software en sentido estricto es todo programa o aplicación programada para realizar tareas especifi
109 desempeñar tareas inteligentes, conduciendo a los componentes físicos o hardware con instrucciones y datos a través de
110 que las computadoras utilizan para manipular datos. » Así mismo se puede decir que es todo lo no físico o intangible de
111 <p></p>
112
113 <!-- NORMAS EN LA CALIDAD DEL SOFTWARE -->
114
115 <h2>Normas ISO en la calidad de un Software</h2>
116 <p>
117 <b>Familias de las normas ISO</b><br>
118
119 Se encarga de los estándares de calidad en los productos o servicios de las empresas, independientemente de su tamaño.
120
121 <br> <br> <b>ISO 27001 : 2013</b><br><br>
122
123
124 « La responsabilidad social está definida bajo esta norma como el compromiso de una organización ante los impactos que
125 el medio ambiente mediante un comportamiento ético y transparente que contribuya al desarrollo sostenible incluyendo su
126
127 <br><br> ISO 9004-2 : 2008</b><br><br>
128
129 Sistema de calidad. Control del producto no aceptado por las directrices de calidad. Tratamiento, almacenamiento, packe
130
131 Registros de calidad. Su finalidad es proveer las especificaciones a la implementación y el mantenimiento del desarroll
132
133 Sus beneficios son los siguientes
134
135 Aumenta la productividad y la eficiencia del software.<br>
136
137 <br><br> Gandarillas Plantea lo siguiente acerca de la norma ISO 12207</b><br><br>
138
139 «Es el estándar para los procesos de ciclo de vida del software de la organización ISO. Los procesos de soporte y de or
140
141 Procesos principales
142
143 Procesos de soporte. En definitiva, encontramos que en la actualidad los softwares son uno de los mayores ingresos econ
144
145 El estándar está dividido en cuatro partes las cuales dirigen, realidad, métricas externas, métricas internas y calidad
146 </p>
147

```

Al final de la información el usuario podrá realizar el test para verificar lo aprendido.

Fragmento del código:

```
<!-- MEDICION DE LA CALIDAD SOFTWARE -->
<h2>Medición de la calidad de software</h2>
<p>
  Siempre habrá un elemento cualitativo en la creación del software para computadoras. " Las métricas de producto por las que
  pruebas pueden realizarse de manera más objetiva y valorarse de modo más cuantitativo".

  <br> Para realizar esto Pressman dice lo siguiente: <br><br>

  ""El primer paso en el proceso de medición es derivar las mediciones y métricas del software que sean adecuadas para la
  análisis se interpretan para obtener comprensión acerca de la calidad del software, y los resultados de la interpretaci
  En algunas instancias, también puede conducir a modificación del proceso de software en sí".

  <br> Para saber si el resultado que realmente se quería obtener Pressman afirma lo siguiente: <br>

  <br> "Defina sólo algunas métricas y luego úselas para obtener comprensión acerca de la calidad de un producto de traba

  <br> Problemas al estimar </b><br>

  <br> Pertenecen a las características físicas del software que se desarrolla, involucra a los agentes envueltos en su a
  deban conocer las características del software. Esto hace que las personas que no tengan conocimientos técnicos acerca
  por lo tanto no nos dejan dar información al cliente final de la aplicación que desconoce del tema y con esto el usuari
  que tiene la aplicación para comparar las estimaciones ya hechas.

  <br> Al momento de realizar mediciones de los aspectos del software de desarrollo, se puede llegar a depender de forma
  y las herramientas que se utilicen en el proceso de codificación y diseño del proyecto.
</p>

<h2>Me siento listo <span style='font-size:20px;'>#128526</span></h2>
<a href="https://forms.gle/jhXUZ48ZsZopi7F6" class="btn">Realizar test</a>
```

Se ha agregado una etiqueta aside que contiene la red social de la facultad.

Fragmento del código:

```
</section>
<aside>
  <h3>Sigue a la Facultad De Estudios Superiores Aragon en Facebook</h3>
  <div id="socialMedia">
    <a href="https://www.facebook.com/FESAragonUNAM" target="blank"> <i class="fab fa-facebook"></i></a>
  </div>
</aside>
</article>
</main>
```

Clase acercaDe.php (Página secundaria)

Se han incluido las fuentes que pertenecen a la página, los iconos dentro de las etiquetas link, al igual que la hoja de estilos CSS.

Fragmento del código:

```
<!DOCTYPE html>
<html Lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Acerca de...</title>

  <!--Estilos CSS-->
  <link rel="stylesheet" href="css/estilos.css">

  <!--Estilo de Fuente-->
  <link href="https://fonts.googleapis.com/css?family=Pathway+Gothic+One&display=swap" rel="stylesheet">

  <!-- IconFile --->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.9.0/css/all.css">

  <!--Favicon-->
  <link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
  <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
  <link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
  <link rel="manifest" href="/site.webmanifest">
  <meta name="msapplication-TileColor" content="#da532d">
  <meta name="theme-color" content="#ffffff">
  <!-- Fin de FavIcon-->
</head>
```

En la etiqueta header, se agrega la barra de navegación, esta contiene el logotipo de la facultad, iconos del link de navegación como home, material de apoyo y acerca de, al igual del usuario que ingresó sesión.

En el navegador se visualiza de la siguiente manera:

The screenshot shows a web browser interface. At the top left is the UNAM logo. The top right navigation bar contains icons for home, a list, a question mark, and a user profile labeled 'Alexis'. Below this is a yellow banner with the text 'ACERCA DE...'. The main content area is divided into two columns. The left column has a section titled 'Sobre este trabajo' followed by text describing the thesis: 'Esta pagina web es parte de la tesis titulada: PÁGINA WEB DIDÁCTICA APLICADA A LA ENSEÑANZA DE LA CALIDAD EN LA INGENIERÍA DE SOFTWARE'. It lists technologies used: HTML5, CSS, JAVASCRIPT, PHP, and MYSQL, and credits Miguel Alexis Rodriguez Peña and Brandon Eliu Cervantes Juarez. The right column features a social media link: 'Sigue a la Facultad De Estudios Superiores Aragon en Facebook' with a Facebook 'f' icon. The bottom of the page contains several lines of placeholder text (Lorem ipsum).

Conclusión

Esta plataforma podrá ser un apoyo en la enseñanza de la asignatura de calidad. Un sistema de calidad es un producto que es ser tomado con importancia en cualquier sector, ya que el software puede ser vulnerable por lo que requiere de mucha planeación para desarrollarlo igual que al proceso, cuidar hasta el más mínimo detalle para que este sea funcional y simple que cualquier usuario pueda usar.

Hay cientos de estándares o normas como lo son las normas ISO que tienen esta finalidad para que un producto o servicio sea desarrollado correctamente, y cumpla con las necesidades que un usuario pueda llegar a tener.

Es importante en el desarrollo de software considerar a la Calidad no solo como un factor necesario sino indispensable para el desarrollo de software. La mayoría de veces para los desarrolladores o estudiantes que se están sumergiendo en este mundo pasa como algo desapercibido, algo que no es necesario implementar o algo que simplemente puede ser dejado al último.

El hacer un software de calidad, o simplemente considerar y seguir las diferentes normas, metodologías o buenas prácticas que se utilizan en el momento de desarrollar software con calidad hará la diferencia entre un buen y un mal trabajo.

Hay gran variedad de metodologías, estándares y normas como lo son las Normas ISO que tienen como finalidad hacer que un producto sea desarrollado correctamente desde su inicio para que este cumpla con las necesidades que el usuario/cliente haya requerido. Ahorrando todos los problemas que puede ocasionar el lanzar un software defectuoso.

La información que contiene esta tesis, es una pequeña guía inicial para aquel que desee saber un poco más sobre la calidad en la Ingeniería de Software para que pueda serle útil, ya sea en un ambiente académico o de aprendizaje autodidacta.

Es muy amplia toda la información para poder incluirla en este trabajo, ya que, por nuestra experiencia propia, al iniciar en el mundo laboral se deben adaptar y adoptar metodologías, estándares, y reglas para desarrollar software. Así como también se empiezan a conocer formas de trabajo distintas a las que uno como buen profesionalista debe adaptarse.

Así mismo, consideramos que esta plataforma puede ayudar a la enseñanza con respecto a la Calidad en la Ingeniería de Software, ya que contiene toda la información de esta tesis y con esto facilitar su lectura.

Anexo: Código completo del proyecto.

El proyecto consiste en 8 clases, así como una clase de estilos en CSS que funciona para darle estilo a la página. Dando un total de 9 clases.

1. acercaDe.php
2. apoyo.php
3. database.php
4. formulario.js
5. index.php
6. login.php
7. logout.php
8. signUp.php
9. estilos.css

Nota: Se acomodaron algunas líneas de código para poder estar en este documento.

1. Clase “*acercaDe.php*”

```
<?php
session_start();

require 'database.php';

if (isset($_SESSION['user_id'])) {
    $records = $conn->prepare('SELECT id, password, nombres, numeroCuenta FROM usuarios WHERE id =
:id');
    $records->bindParam(':id', $_SESSION['user_id']);
    $records->execute();
```

```

$results = $records->fetch(PDO::FETCH_ASSOC);

$user = null;

if (count($results) > 0) {
    $user = $results;
}
}
?>

<!--VALIDACIONES PARA QUE REDIRECCIONE AL SIGNUP SI LA SESION NO SE ENCUENTRA INICIADA----->

<?php if (!empty($user)) : ?>
<?php else : ?>
    <meta http-equiv="refresh" content="0;url=login.php">
<?php endif; ?>

<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Acerca de...</title>

    <!--Estilos CSS-->
    <link rel="stylesheet" href="css/estilos.css">

    <!--Estilo de Fuente-->
    <link href="https://fonts.googleapis.com/css?family=Pathway+Gothic+One&display=swap"
rel="stylesheet">

    <!-- IconFile --->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.9.0/css/all.css">

    <!--Favicon-->
    <link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
    <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
    <link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
    <link rel="manifest" href="/site.webmanifest">
    <meta name="msapplication-TileColor" content="#da532c">
    <meta name="theme-color" content="#ffffff">
    <!-- Fin de FavIcon--->
</head>

<body>
    <!-- Cabecera--->
    <div class="top">
        <ul class="container">
            <li><i class="far fa-envelope"></i></i> plataformaweb.fesaragon@gmail.com</li>
        </ul>
    </div>

    <header>
        <nav id="navbar">
            <div id="navegacion" class="container">
                <a href="index.php"></a>

                <span id="navbar-toggle">
                    <i class="fas fa-bars"></i>
                </span>
                <ul id="main-nav">
                    <li>
                        <a class="tooltip" href="index.php">
                            <i class="fas fa-home"></i> <span class="tooltip-box">Ir a inicio</span></a></li>
                    <li>
                        <a class="tooltip" href="apoyo.php">
                            <i class="fas fa-book"></i> <span class="tooltip-box">Apoyo</span> </a></li>
                </ul>
            </div>
        </nav>
    </header>

```

```

<li>
  <a class="tooltip" href="acercaDe.php">
    <i class="fas fa-question-circle"> </i>
    <span class="tooltip-box">Acerca de</span> </a> </li>
<li>
  <a <i class="fas fa-user-tie">&nbsp;&nbsp;&nbsp;</i><?= $user['nombres']; ?></a>
  <ul>
    <li> <a href="logout.php"><i class="fas fa-power-off"></i>&nbsp;&nbsp;&nbsp;Cerrar
sesión</a></li>
  </ul>
</ul>
</div>
</nav>
</header>

<!--Cuerpo del documento-->
<main>
  <header>
    <div class="container">
      <h1>Acerca de...</h1>
    </div>
  </header>
  <article id="pages" class="container">
    <section>
      <h2>Sobre este trabajo</h2>
      <p>Esta pagina web es parte de la tesis titulada: <br>
PÁGINA WEB DIDÁCTICA APLICADA A LA ENSEÑANZA DE LA CALIDAD EN LA INGENIERÍA DE
SOFTWARE <br>

Desarrollada con: <br>

HTML5 <br>

CSS <br>

JAVASCRIPT <br>

PHP <br>

MYSQL <br>

Elaborada por: Miguel Alexis Rodriguez Peña y Brandon Eliu Cervantes Juarez</p>

<p>Lorem ipsum dolor sit, amet consectetur adipisicing elit.
Eaque, quam incidunt voluptatem
obcaecati animi ad. Eius, itaque laboriosam
quo saepe, error, magni temporibus asperiores
consectetur molestias veritatis incidunt ipsum hic!
Lorem ipsum dolor sit, amet consectetur adipisicing
elit. Eaque, quam incidunt voluptatem
obcaecati animi ad. Eius, itaque
laboriosam quo saepe, error, magni temporibus asperiores
consectetur molestias veritatis incidunt ipsum hic!
Lorem ipsum dolor sit, amet consectetur adipisicing
elit. Eaque, quam incidunt voluptatem
obcaecati animi ad. Eius, itaque laboriosam quo saepe,
error, magni temporibus asperiores
consectetur molestias veritatis incidunt
ipsum hic! Lorem ipsum dolor sit, amet consectetur
adipisicing elit. Eaque, quam incidunt voluptatem
obcaecati animi ad. Eius, itaque laboriosam quo saepe,
error, magni temporibus asperiores
consectetur molestias veritatis incidunt ipsum hic!</p>

<h2>Apoyo</h2>
<p>Lorem ipsum dolor sit, amet consectetur adipisicing elit.
Molestiae itaque accusamus hic vero.
Velit labore quas harum nisi sequi consectetur provident

```

```

perferendis illum consequatur sit.
Nostrum non ex commodi eaque?Lorem ipsum dolor sit, amet
consectetur adipisicing elit.
Molestiae itaque accusamus hic vero.Velit labore quas
harum nisi sequi consectetur provident
perferendis illum consequatur sit. Nostrum non ex
commodi eaque?
Lorem ipsum dolor sit, amet consectetur adipisicing elit.
Molestiae itaque accusamus hic vero.
Velit labore quas harum nisi sequi consectetur provident
perferendis illum consequatur sit.
Nostrum non ex commodi eaque?
Lorem ipsum dolor sit, amet consectetur adipisicing elit.
Molestiae itaque accusamus hic vero.
Velit labore quas harum nisi sequi consectetur provident
perferendis illum consequatur sit.
Nostrum non ex commodi eaque?

Lorem ipsum dolor sit, amet consectetur adipisicing elit.
Molestiae itaque accusamus hic vero.
Velit labore quas harum nisi sequi consectetur provident
perferendis illum consequatur sit.
Nostrum non ex commodi eaque?Lorem ipsum dolor sit, amet
consectetur adipisicing elit.
Molestiae itaque accusamus hic vero.Velit labore quas harum
nisi sequi consectetur provident
perferendis illum consequatur sit. Nostrum non ex commodi
eaque?
Lorem ipsum dolor sit, amet consectetur adipisicing elit.
Molestiae itaque accusamus hic vero.
Velit labore quas harum nisi sequi consectetur provident
perferendis illum consequatur sit.
Nostrum non ex commodi eaque?
Lorem ipsum dolor sit, amet consectetur adipisicing elit.
Molestiae itaque accusamus hic vero.
Velit labore quas harum nisi sequi consectetur provident
perferendis illum consequatur sit.
Nostrum non ex commodi eaque?
Nostrum non ex commodi eaque?

```

```

</p>

</section>

<aside>
  <h3>Sigue a la Facultad De Estudios Superiores Aragon en Facebook</h3>
  <div id="socialMedia">
    <a href="https://www.facebook.com/FESAragonUNAM" target="blank">
      <i class="fab fa-facebook"></i> </a>
    </div>
</aside>
</article>
</main>

<!-- Pie de pagina---->
<footer>
  <div id="pieDePagina" class="container">
    <a class="tooltip" href="index.php">
      <i class="fas fa-home"></i> <span class="tooltip-box">Ir a inicio</span></a>
    <a class="tooltip" href="apoyo.php">
      <i class="fas fa-book"></i> <span class="tooltip-box">Apoyo</span> </a>
    <a class="tooltip" href="acercaDe.php">
      <i class="fas fa-question-circle"> </i> <span class="tooltip-box">Acerca de</span> </a>
    </div>

    <div id="copyright">
      <small>Universidad Nacional Autonoma de México &copy; Derechos reservados. </small>
    </div>
</footer>

<script>
  // Comportamientos para el menú de navegación.

```

```

let mainNav = document.getElementById('main-nav');
let navbarToggle = document.getElementById('navbar-toggle');
navbarToggle.addEventListener('click', function() {

    if (this.classList.contains('active')) {
        mainNav.style.display = "none";
        this.classList.remove('active');
    } else {
        mainNav.style.display = "flex";
        this.classList.add('active');
    }
});
</script>

</body>

</html>

```

2. Clase “apoyo.php”

```

<?php
session_start();

require 'database.php';

if (isset($_SESSION['user_id'])) {
    $records = $conn->prepare('SELECT id, password, nombres, numeroCuenta FROM usuarios WHERE id =
:id');
    $records->bindParam(':id', $_SESSION['user_id']);
    $records->execute();
    $results = $records->fetch(PDO::FETCH_ASSOC);

    $user = null;

    if (count($results) > 0) {
        $user = $results;
    }
}

?>

<!--VALIDACIONES PARA QUE REDIRECCIONE AL SIGNUP SI LA SESION NO SE ENCUENTRA INICIADA----->

<?php if (!empty($user)) : ?>
<?php else : ?>
    <meta http-equiv="refresh" content="0;url=login.php">
<?php endif; ?>

<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Material didactico para apoyo a la calidad</title>

    <!--Estilos CSS-->
    <link rel="stylesheet" href="css/estilos.css">

    <!--Estilo de Fuente-->
    <link href="https://fonts.googleapis.com/css?family=Pathway+Gothic+One&display=swap"
rel="stylesheet">

    <!-- IconFile --->

```

```

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.9.0/css/all.css">

<!--Favicon-->
<link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
<link rel="manifest" href="/site.webmanifest">
<meta name="msapplication-TileColor" content="#da532c">
<meta name="theme-color" content="#ffffff">
<!-- Fin de Favicon-->
</head>

<body>
<!-- Cabecera-->
<div class="top">
  <ul class="container">
    <li><i class="far fa-envelope"></i></i> plataformaweb.fesaragon@gmail.com</li>
  </ul>
</div>

<header>
  <nav id="navbar">
    <div id="navegacion" class="container">
      <a href="index.php"></a>

      <span id="navbar-toggle">
        <i class="fas fa-bars"></i>
      </span>
      <ul id="main-nav">
        <li><a class="tooltip" href="index.php"> <i class="fas fa-home"></i> <span
class="tooltip-box">Ir a inicio</span></a></li>
        <li><a class="tooltip" href="apoyo.php"> <i class="fas fa-book"></i> <span
class="tooltip-box">Apoyo</span> </a></li>
        <li><a class="tooltip" href="acercaDe.php"> <i class="fas fa-question-circle"> </i>
<span class="tooltip-box">Acerca de</span> </a> </li>
        <li><a class="tooltip" href="https://forms.gle/jhXUZ4BZsjZopi7F6"> <i class="fas
fa-brain"> </i> <span class="tooltip-box">Realizar Test</span> </a> </li>

        <li>
          <a <i class="fas fa-user-tie">&nbsp;&nbsp;&nbsp;</i><? = $user['nombres']; ?></a>
          <ul>
            <li> <a href="logout.php"><i class="fas fa-power-off"></i>&nbsp;&nbsp;&nbsp;Cerrar
sesión</a></li>
          </ul>
        </li>
      </ul>
    </div>
  </nav>
</header>

<!--Cuerpo del documento-->
<main>
  <header>
    <div class="container">
      <h1>Material didactico para apoyo a la calidad</h1>
    </div>
  </header>
  <article id="pages" class="container">
    <section>
      <h2>Control de calidad en la ingeniería de software</h2>
      <p>Quizá para muchos estudiantes el aspecto de la calidad en el desarrollo de software
es algo a lo que a veces no se le da mucha importancia, pasa desapercibido, o por malas prácticas no es
tomado en cuenta. Y esto no es un problema aislado, actualmente la mayoría de software está
compenetrado en los bienes y servicios de la sociedad, pueden ser áreas tales como las de la salud, las
telecomunicaciones, la seguridad nacional, la economía y muchas más. Resultaría realmente desastroso
que fallara por malas prácticas, es por eso que existen normas y prácticas que se deben seguir, para
que un software sea confiable, eficaz y eficiente. Ingeniería.

      </p> </p>Mientras que Pressman menciona que:</p></p>

```

«El software de computadora es el producto que construyen los programadores profesionales y al que después le dan mantenimiento durante un largo tiempo.»

« El software es la interfaz y la lógica abstracta que vincula al ser humano con una tecnología tangible y/o otro software, permitiendo una interacción y retroalimentación mutuas.»

«En computación, el software en sentido estricto es todo programa o aplicación programada para realizar tareas específicas. Es el soporte lógico e inmaterial que permite que la computadora pueda

desempeñar tareas inteligentes, conduciendo a los componentes físicos o hardware con instrucciones y datos a través de diferentes tipos de programas. El software es el conjunto de instrucciones

que las computadoras utilizan para manipular datos. » Así mismo se puede decir que es todo lo no físico o intangible dentro de un sistema de cómputo.

<p></p>

<!-- NORMAS EN LA CALIDAD DEL SOFTWARE -->

<h2>Normas ISO en la calidad de un Software</h2>

<p>

Familias de las normas ISO

Se encarga de los estándares de calidad en los productos o servicios de las empresas, independientemente de su tamaño.

 ISO 27001 : 2013

« La responsabilidad social está definida bajo esta norma como el compromiso de una organización ante los impactos que sus decisiones y actividades ocasionan en la sociedad y

el medio ambiente mediante un comportamiento ético y transparente que contribuya al desarrollo sostenible incluyendo salud y bienestar de la sociedad.

 ISO 9004-2 : 2008

Sistema de calidad. Control del producto no aceptado por las directrices de calidad. Tratamiento, almacenamiento, packaging y entrega del producto. Producto que se proporciona al comprador.

Registros de calidad. Su finalidad es proveer las especificaciones a la implementación y el mantenimiento del desarrollo del software.

Sus beneficios son los siguientes

Aumenta la productividad y la eficiencia del software.

 Gandarillas Plantea lo siguiente acerca de la norma ISO 12207

«Es el estándar para los procesos de ciclo de vida del software de la organización ISO. Los procesos de soporte y de organización deben existir independientemente de la organización y del proyecto ejecutado.

Procesos principales

Procesos de soporte. En definitiva, encontramos que en la actualidad los softwares son uno de los mayores ingresos económicos alrededor del mundo, ya que se ofrece una gran oportunidad y diversas fuentes de negocio para países que están en desarrollo, por lo que el software se debe estar actualizando constantemente. Es por ello que fue creada la Norma ISO 12207, esta norma es la encargada de definir el ciclo de vida que tendrá el software, desde los procesos, las actividades que realizará y las configuraciones que tendrá. « Es un estándar internacional para la evaluación de la calidad del software. »

El estándar está dividido en cuatro partes las cuales dirigen, realidad, métricas externas, métricas internas y calidad en las métricas de uso y expendido.

</p>

<!-- MEDICION DE LA CALIDAD SOFTWARE -->

<h2>Medición de la calidad de software</h2>

<p>

Siempre habrá un elemento cualitativo en la creación del software para computadoras. " Las métricas de producto proporcionan una base desde donde el análisis, el diseño, la codificación y las

pruebas pueden realizarse de manera más objetiva y valorarse de modo más cuantitativo".

 Para realizar esto Pressman dice lo siguiente:

"El primer paso en el proceso de medición es derivar las mediciones y métricas del software que sean adecuadas para la presentación del software que se está construyendo. Los resultados del

análisis se interpretan para obtener comprensión acerca de la calidad del software, y los resultados de la interpretación conducen a modificación de requerimientos y modelos de diseño, código fuente o casos de prueba. "

En algunas instancias, también puede conducir a modificación del proceso de software en sí".

 Para saber si el resultado que realmente se quería obtener Pressman afirma lo siguiente:

 "Defina sólo algunas métricas y luego úselas para obtener comprensión acerca de la calidad de un producto de trabajo de ingeniería del software".

 Problemas al estimar

 Pertenecen a las características físicas del software que se desarrolla, involucra a los agentes envueltos en su análisis y que deban revisar los valores, deban conocer las características del software. Esto hace que las personas que no tengan conocimientos técnicos acerca del desarrollo de software, no puedan entender correctamente los valores obtenidos y

por lo tanto no nos dejan dar información al cliente final de la aplicación que desconoce del tema y con esto el usuario no nos aporta sus conocimientos del entorno y su experiencia de los procesos de negocio

que tiene la aplicación para comparar las estimaciones ya hechas.

 Al momento de realizar mediciones de los aspectos del software de desarrollo, se puede llegar a depender de forma extrema a los lenguajes de programación y las herramientas que se utilicen en el proceso de codificación y diseño del proyecto.

</p>

<h2>Me siento listo #128526;</h2>
Realizar test

</section>

<aside>

<h3>Sigue a la Facultad De Estudios Superiores Aragon en Facebook</h3>

<div id="socialMedia">

 <i class="fab fa-facebook"></i>

</div>

</aside>

</article>

</main>

<!-- Pie de pagina---->

<footer>

<div id="pieDePagina" class="container">

 <i class="fas fa-home"></i> Ir a inicio

 <i class="fas fa-book"></i> Apoyo

 <i class="fas fa-question-circle"> </i> Acerca de

</div>

<div id="copyright">

<small>Universidad Nacional Autonoma de México © Derechos reservados. </small>

</div>

</footer>

<script>

```

// Comportamientos para el menú de navegación.
let mainNav = document.getElementById('main-nav');
let navbarToggle = document.getElementById('navbar-toggle');
navbarToggle.addEventListener('click', function() {

    if (this.classList.contains('active')) {
        mainNav.style.display = "none";
        this.classList.remove('active');
    } else {
        mainNav.style.display = "flex";
        this.classList.add('active');
    }
});
</script>
</body>
</html>

```

3. Clase “database.php”

```

<!--
  Datos de la conexión de PHP
-->
<?php

$server = 'localhost:3306';           // Ip o dominio donde se encuentra la base de datos
$username = 'root';                  // Usuario de XAMPP
$password = '';                      // Contraseña de XAMPP
$dbname = 'php_registro_usuarios';   // Nombre de la base de datos de MySQL

try {
    $conn = new PDO("mysql:host=$server;dbname=$dbname;", $username, $password);
} catch (PDOException $e) {
    die('Connection Failed: ' . $e->getMessage());
}

?>

```

4. Clase “formulario.js”

```

const nombre = document.querySelector("#nombre");
const pattern = new RegExp("[A-Z]+$", "i");

//Leemos el input
nombre.addEventListener("input", (e) => {
    if (e.target.value === "" || !pattern.test(e.target.value)) {
        console.log("No puedes");
    }
});

```

5. Clase “index.php”

```

<?php
session_start();

```

```

require 'database.php';

if (isset($_SESSION['user_id'])) {
    $records = $conn->prepare('SELECT id, password, nombres, numeroCuenta FROM usuarios WHERE id =
:id');
    $records->bindParam(':id', $_SESSION['user_id']);
    $records->execute();
    $results = $records->fetch(PDO::FETCH_ASSOC);

    $user = null;

    if (count($results) > 0) {
        $user = $results;
    }
}

?>

<!--VALIDACIONES PARA QUE REDIRECCIONE AL SIGNUP SI LA SESION NO SE ENCUENTRA INICIADA----->
<?php if (!empty($user)) : ?>
<?php else : ?>
    <meta http-equiv="refresh" content="0;url=login.php">
    <? die(); ?>
<?php endif; ?>

<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Enseñanza de la materia de calidad</title>

    <!--Estilos CSS-->
    <link rel="stylesheet" href="css/estilos.css">

    <!--Estilo de Fuente-->
    <link href="https://fonts.googleapis.com/css?family=Ibarra+Real+Nova&display=swap"
rel="stylesheet">

    <!-- IconFile --->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.9.0/css/all.css">

    <!--Favicon-->
    <link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
    <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
    <link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
    <link rel="manifest" href="/site.webmanifest">
    <meta name="msapplication-TileColor" content="#da532c">
    <meta name="theme-color" content="#ffffff">
    <!-- Fin de FavIcon-->
</head>

<body>

    <!-- Cabecera--->
    <div class="top">
        <ul class="container">
            <li><i class="far fa-envelope"></i></i> plataformaweb.fesaragon@gmail.com</li>
        </ul>
    </div>

    <header>
        <nav id="navbar">
            <div id="navegacion" class="container">
                <a href="index.php"></a>

                <span id="navbar-toggle">

```

```

        <i class="fas fa-bars"></i>
    </span>
    <ul id="main-nav">
        <li>
            <a class="tooltip" href="index.php">
                <i class="fas fa-home"></i>
                <span class="tooltip-box">Ir a inicio</span></a>
            </li>

            <li><a class="tooltip" href="apoyo.php">
                <i class="fas fa-book"></i>
                <span class="tooltip-box">Material de Apoyo</span> </a></li>

            <li><a class="tooltip" href="acercaDe.php">
                <i class="fas fa-question-circle"> </i>
                <span class="tooltip-box">Acerca de</span> </a> </li>

            <li>
                <a <i class="fas fa-user-tie">&nbsp;&nbsp;&nbsp;</i><?=$user['nombres']; ?></a>
                <ul>
                    <li> <a href="logout.php"><i class="fas fa-power-off"></i>&nbsp;&nbsp;&nbsp;</a></li>
                </ul>
            </li>
        </ul>
    </div>
</nav>
</header>

<!--Cuerpo del documento-->
<main>
    <article id="home">

        <!-- Seccion Uno ---->
        <section>
            <div id="seccionUno" class="container">
                <div id="centaralign">
                    <h1> Enseñanza de la calidad en ingenieria de software </h1>
                </div>
            </div>
        </section>

        <!-- Seccion Dos ---->
        <section>
            <div id="seccionDos" class="container">
                <div>
                    <h2>Nuestro objetivo:</h2>
                    <p> Es brindar a los estudiantes de ingenieria una guia practica y confiable para la materia de "Calidad" impartida en la FES Aragon, asi como hacer de este aprendizaje algo mas entretenido.</p>

                    <a class="btn" href="apoyo.php"> Realizar pre-evaluacion</a>
                </div>
                
            </div>
        </section>

        <!-- Seccion Tres---->
        <section>

            <div id="seccionTres" class="container">
                <p>APRENDE SOBRE QUE ES LA CALIDAD <br>
                Y COMO SE LLEVA A CABO EN LA ING DE SOFTWARE
            </p>
            </div>
        </section>

        <!-- Seccion Cuatro---->
        <section>

```

```

<h2>Capitulos</h2>
<p>Puedes leer o descargar los capitulos de esta tesis para leerlos de forma individual
segun sea tu interes</p>

<div id="seccionCuatro" class="container">
  <!--Iconos de FotnAwesome-->

  <div class="col">

    <a href="download/CalidadSoftware.pdf" download="CalidadSoftware.pdf">
      <i class="fas fa-file-pdf"></i>
      <h3>Control de calidad en la Ingeniería de Software</h3>
    </a>
  </div>

  <div class="col">

    <!-- CAMBIAR ARCHIVOS -->
    <a href="download/NormasIsoCalidad.pdf" download="NormasIsoCalidad.pdf">
      <i class="fas fa-file-pdf"></i>
      <h3>Normas ISO en la calidad de un Software</h3>
    </a>
  </div>

  <div class="col">
    <a href="download/Medición de la calidad de software.pdf" download="Medición de
la calidad de software.pdf">
      <i class="fas fa-file-pdf"></i>
      <h3>Medición de la calidad de software: Requerimiento de tiempo</h3>
    </a>
  </div>

  <div class="col">
    <a href="download/CalidadSoftware.pdf" download="CalidadSoftware.pdf">
      <i class="fas fa-file-pdf"></i>
      <h3>Desarrollo de la página web</h3>
    </a>
  </div>
</div>
</section>

<!-- Seccion Cinco---->

<section>
  <div id="seccionCinco" class="container">
    <div>
      <h2>Material didactico</h2>
      <p>Revisa el material didactico para despues presentar
una pequeña prueba que evalúe tus conocimientos</p>
    </div>
    <a class="btn" href="apoyo.php">Acceder</a>

  </div>
</section>
</article>
</main>

<!-- Pie de pagina---->
<footer>

  <div id="pieDePagina" class="container">
    <a class="tooltip" href="index.php">
      <i class="fas fa-home"></i>
      <span class="tooltip-box">Ir a inicio</span></a>

    <a class="tooltip" href="apoyo.php">
      <i class="fas fa-book"></i>
      <span class="tooltip-box">Apoyo</span> </a>

    <a class="tooltip" href="acercaDe.php">
      <i class="fas fa-question-circle"> </i>
      <span class="tooltip-box">Acerca de</span> </a>
  </div>

```

```

<a class="tooltip" href="https://forms.gle/jhXUZ4BZsjZopi7F6">
<i class="fas fa-brain"> </i>
<span class="tooltip-box">Realizar Test</span> </a>

</div>
<div id="copyright">
  <small>Universidad Nacional Autonoma de México &copy; Derechos reservados. </small>
</div>
</footer>

<script>
  // Comportamientos para el menú de navegación.
  let mainNav = document.getElementById('main-nav');
  let navbarToggle = document.getElementById('navbar-toggle');
  navbarToggle.addEventListener('click', function() {

    if (this.classList.contains('active')) {
      mainNav.style.display = "none";
      this.classList.remove('active');
    } else {
      mainNav.style.display = "flex";
      this.classList.add('active');
    }
  });

  //Animacion de texto del area 1

  function typeEffect(element, speed) {
    var text = element.innerHTML;
    element.innerHTML = "";

    var i = 0;
    var timer = setInterval(function() {
      if (i < text.length) {
        element.append(text.charAt(i));
        i++;
      } else {
        clearInterval(timer);
      }
    }, speed);
  }

  // application
  var speed = 75;
  var h1 = document.querySelector('h1');
  var p = document.querySelector('p');
  var delay = h1.innerHTML.length * speed + speed;

  // type affect to header
  typeEffect(h1, speed);

  // type affect to body
  setTimeout(function() {
    p.style.display = "inline-block";
    typeEffect(p, speed);
  }, delay);

  //Fin de la animacion de texto del area 1
</script>
</body>
</html>

```

6. Clase “login.php”

```

<?php

session_start();

if (isset($_SESSION['user_id'])) {
    header('Location: /paginatesis/index.php');
}
require 'database.php';

if (
    !empty($_POST['numeroCuenta']) &&
    !empty($_POST['password'])
) {

    $records = $conn->prepare('SELECT id, numeroCuenta, password FROM usuarios WHERE numeroCuenta =
:numeroCuenta');
    $records->bindParam(':numeroCuenta', $_POST['numeroCuenta']);
    $records->execute();
    $results = $records->fetch(PDO::FETCH_ASSOC);

    $message = '';

    if ($results && count($results) > 0 && password_verify($_POST["password"], $results["password"])) {
        // if (count($results) > 0 && password_verify($_POST['password'], $results['password'])) {
        $_SESSION['user_id'] = $results['id'];
        header("Location: /paginatesis/index.php");
    } else {
        $message = 'Error: El usuario no existe, intente nuevamente';
    }
}

?>

<!--VALIDACIONES PARA QUE REDIRECCIONE AL INDEX SI LA SESION YA SE ENCUENTRA INICIADA----->

<?php if (!empty($user)) : ?>
<meta http-equiv="refresh" content="0;url=index.php">
<?php else : ?>
<? die(); ?>
<?php endif; ?>

<!DOCTYPE html>
<html lang="es">

<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Calidad Ing Software</title>

<!--Estilo de Fuente-->
<link href="https://fonts.googleapis.com/css?family=Ibarra+Real+Nova&display=swap"
rel="stylesheet">

<!-- IconFile --->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.9.0/css/all.css">

<!--Favicon-->
<link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">

```

```

<link rel="manifest" href="/site.webmanifest">
<meta name="msapplication-TileColor" content="#da532c">
<meta name="theme-color" content="#ffffff">
<!-- Fin de FavIcon-->
<!--Estilos CSS-->
<link rel="stylesheet" href="css/estilos.css">
</head>

<body>

<!-- Cabecera-->
<div class="top">
  <ul class="container">
    <li><i class="far fa-envelope"></i></i> plataformaweb.fesaragon@gmail.com</li>
  </ul>
</div>

<!-- PHP --->

<?php if (!empty($message)) : ?>
  <p> <?= $message ?></p>
<?php endif; ?>

<!--Cuerpo del documento-->
<main>
  <article id="home">

    <!-- Seccion Uno ---->
    <section>
      <div id="seccionUno" class="container">
        <div id="formulario">
          <form class="form" action="login.php" method="POST">

            <div class="tituloForm">
              <h2>Enseñanza de la Calidad en Ing de Software</h2>
              <h2>Iniciar sesión</h2>
            </div>

            <!-- <label for="suNumeroCuenta">Número de cuenta: </label> -->
            <div class="inputBox">
              <input type="text" placeholder="Número de cuenta"
                name="numeroCuenta" id="suNumeroCuenta" required>
            </div>

            <!-- <label for="suPassword">Contraseña: </label> -->
            <div class="inputBox">
              <input type="password" placeholder="Contraseña"
                name="password" id="suNumeroCuenta" required>
            </div>

            <div class="inputButtonEnviar">
              <input type="submit" value="Ingresar">
            </div>

            <small class="olvidado">Aun no estas registrado?
            <a href="signUp.php" target="#">Regístrate aquí</a>
            </small>

          </form>
        </div>
      </div>
    </section>
  </article>
</main>

<!-- Pie de pagina---->
<footer>
  <div id="copyright">
    <small> Universidad Nacional Autonoma de México &copy; Derechos reservados. </small>
  </div>
</footer>
</body>

```

```
</html>
```

7. Clase “logout.php”

```
<?php
session_start();
session_unset();
session_destroy();
header('Location: /paginatesis/login.php');
?>
```

8. Clase “signUp.php”

```
<?php
require 'database.php';

$message = '';

if (
    !empty($_POST['nombres']) &&
    !empty($_POST['apellidoPaterno']) &&
    !empty($_POST['apellidoMaterno']) &&
    !empty($_POST['numeroCuenta']) &&
    !empty($_POST['carrera']) &&
    !empty($_POST['email']) &&
    !empty($_POST['password'])
) {
    $sql = "INSERT INTO usuarios (nombres, apellidoPaterno,
    apellidoMaterno, numeroCuenta, carrera, email, password)
    VALUES (:nombres, :apellidoPaterno, :apellidoMaterno,
    :numeroCuenta, :carrera, :email, :password)";

    $stmt = $conn->prepare($sql);

    $stmt->bindParam(':nombres', $_POST['nombres']);
    $stmt->bindParam(':apellidoPaterno', $_POST['apellidoPaterno']);
    $stmt->bindParam(':apellidoMaterno', $_POST['apellidoMaterno']);
    $stmt->bindParam(':numeroCuenta', $_POST['numeroCuenta']);
    $stmt->bindParam(':carrera', $_POST['carrera']);
    $stmt->bindParam(':email', $_POST['email']);

    $password = password_hash($_POST['password'], PASSWORD_BCRYPT);

    $stmt->bindParam(':password', $password);

    if ($stmt->execute()) {
        $message = 'El usuario se ha creado exitosamente';
    } else {
        $message = 'Lo sentimos, hubo un error al crear su usuario. Intente mas tarde';
    }
}
?>
```

```

<!--VALIDACIONES PARA QUE REDIRECCIONE AL INDEX SI LA SESION YA SE ENCUENTRA INICIADA----->

<?php
session_start();

require 'database.php';

if (isset($_SESSION['user_id'])) {
    $records = $conn->prepare('SELECT id, password, nombres, numeroCuenta FROM usuarios WHERE id =
:id');
    $records->bindParam(':id', $_SESSION['user_id']);
    $records->execute();
    $results = $records->fetch(PDO::FETCH_ASSOC);

    $user = null;

    if (count($results) > 0) {
        $user = $results;
    }
}

?>

<?php if (!empty($user)) : ?>
<meta http-equiv="refresh" content="0;url=index.php">
<?php else : ?>
<? die(); ?>
<?php endif; ?>

<!DOCTYPE html>
<html lang="es">

<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Calidad Ing Software</title>

<!--Estilos CSS-->
<link rel="stylesheet" href="css/estilos.css">

<!--Estilo de Fuente-->
<link href="https://fonts.googleapis.com/css?family=Ibarra+Real+Nova&display=swap"
rel="stylesheet">

<!-- IconFile --->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.9.0/css/all.css">

<!--Favicon-->
<link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
<link rel="manifest" href="/site.webmanifest">
<meta name="msapplication-TileColor" content="#da532c">
<meta name="theme-color" content="#ffffff">
<!-- Fin de FavIcon-->
</head>

<body>

<!-- Cabecera--->
<div class="top">
    <ul class="container">
        <li><i class="far fa-envelope"></i></li> plataformaweb.fesaragon@gmail.com</li>
    </ul>
</div>

```

```

<!-- PHP --->
<?php if (!empty($message)) : ?>
    <p> <?= $message ?></p>
<?php endif; ?>

<!--Cuerpo del documento-->
<main>
    <article id="home">

        <!-- Seccion Uno ---->

        <section>
            <div id="seccionUno" class="container">

                <div id="formulario">

                    <form action="signUp.php" method="POST">
                        <div class="tituloForm">
                            <h2>Enseñanza de la Calidad en Ing de Software</h2>
                            <h2>Registro</h2>
                        </div>

                        <label for="nombres">Nombre (s): </label>

                        <div class="inputBox">
                            <input name="nombres" maxlength="40" id="nombre"
                                type="text" placeholder="Ingresa tu(s) nombre(s)" required>
                        </div>

                        <label for="apellidoPaterno">Apellido paterno: </label>

                        <div class="inputBox">
                            <input name="apellidoPaterno" maxlength="15"
                                type="text" placeholder="Ingresa tu apellido paterno" required>
                        </div>

                        <label for="apellidoMaterno">Apellido materno: </label>
                        <div class="inputBox">
                            <input name="apellidoMaterno" type="text"
                                maxlength="15" placeholder="Ingresa tu apellido materno" required>
                        </div>

                        <label for="numeroCuenta">Numero de cuenta: </label>
                        <div class="inputBox">
                            <input name="numeroCuenta" id="numCuenta"
                                type="number" placeholder="Ingresa tu numero de cuenta" required>
                        </div>

                        <label for="carrera">Carrera </label>
                        <select name="carrera" required>
                            <option disabled="">Selecciona una carrera</option>
                            <option value='Ingenieria en computación'> Ingenieria en computación
                            <option value='Ingenieria electrica electronica'>Ingenieria electrica
                                electronica
                            <option value='Ingenieria industrial'> Ingenieria industrial
                        </select>

                        <label for="email">Correo Electronico: </label>
                        <div class="inputBox">
                            <input name="email" type="email"
                                placeholder="Ingresa tu correo electronico" required>
                        </div>

                        <label for="password">Contraseña: </label>
                        <div class="inputBox">
                            <input name="password" type="password"
                                placeholder="Ingresa tu contraseña" required>

```

```

        </div>
        <div class="inputButtonEnviar">
            <input type="submit" value="Registrar">
        </div>

        <small class="olvidado"><br>Si ya estas registrado, ¡Inicia sesion,
ahora!<br>
            <a href="login.php" target="#">Inicia Sesion</a></small>
        </form>
    </div>
</div>

</section>
</article>
</main>

<!-- Fin de la seccion ---->

<!-- Pie de pagina---->
<footer>
    <div id="copyright">
        <small>Universidad Nacional Autonoma de México &copy; Derechos reservados.</small>
    </div>

</footer>
<script src="formulario.js"></script>
</body>

</html>

```

9. Clase “estilos.css”

```

/*Estilos generales*/

html, body {
    height: 100%;
}

/*Modificar el modelo de cajas de CSS*/

html {
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
}

*, ::before, *::after {
    -webkit-box-sizing: inherit;
    box-sizing: inherit
}

/* Resetear margenes y padding*/

* {
    margin: 0;
    padding: 0;
}

body {
    font-family: 'arial', sans-serif;
    font-size: 1.09rem;
}

.container {
    width: 100%;
}

```

```

    max-width: 1366px;
    margin: 0 auto;
}

/*Estilo general en los textos*/

h2 {
    font-size: 1.45rem;
    margin-bottom: 20px;
}

p {
    line-height: 1.8;
    padding-right: 1em;
    margin-bottom: 25px;
}

/*Estilos de los botones*/

.btn {
    display: block;
    background-color: #eaa522;
    width: 250px;
    padding: 12px 1em;
    text-align: center;
    text-decoration: none;
    color: #fff;
    border: 1px solid #eaa522;
    font-weight: bold;
}

.btn:hover {
    transition: all 0.8s;
    background-color: transparent;
    color: #eaa522;
    border-color: #eaa522;
}

/*Estilos de los conetenidos*/

/*Barra de navegacion*/

#navbar {
    background-color: #003560;
}

#navegacion {
    display: -webkit-box;
    display: -ms-flexbox;
    display: flex;
    -webkit-box-pack: justify;
    -ms-flex-pack: justify;
    justify-content: space-between;
}

#main-nav {
    display: -webkit-box;
    display: -ms-flexbox;
    display: flex;
    list-style: none;
    font-family: 'lato', sans-serif;
}

/* Modificar iconos de la barra de tareas */

#main-nav li a {
    color: #fff;
    text-decoration: none;
    padding: 50px 1.9rem;
    display: inline-block;
}

```

```

#navbar-toggle {
  color: #fff;
  font-size: 1.5rem;
  position: absolute;
  top: 62px;
  right: 35px;
  display: none;
}

.top {
  display: none;
}

nav li ul {
  list-style: none;
  position: absolute;
  display: none;
}

nav li:hover ul {
  background-color: #003560;
  display: block;
}

/*Tooltip de los iconos de la barra de nav*/

.tooltip {
  font-weight: normal;
  color: rgb(25, 27, 27);
  text-decoration: none;
  position: relative;
}

input:invalid {
  box-shadow: 0 0 5px 1px red;
}

input:focus:invalid {
  box-shadow: none;
}

.tooltip:hover {
  text-decoration: underline;
}

.tooltip:hover .tooltip-box {
  display: inline-block;
}

.tooltip-box {
  display: none;
  position: absolute;
  background: rgb(9, 163, 163);
  line-height: 15px;
  z-index: 500;
  text-align: center;
  color: #fff;
  font-size: 10px;
  padding: 5px 20px;
  border-radius: 5px;
  left: 0;
  right: 0;
  bottom: -5px;
}

.tooltip-box::after {
  content: "";
  display: block;
}

/* Correo electronico*/

```

```

#main-nav-tel {
  color: #fff;
  list-style: none;
  padding: 1em;
  text-decoration: none;
  font-weight: bold;
  display: grid;
  font-weight: bold;
  font-family: 'lato', sans-serif;
}

#main-nav li a:hover {
  background-color: #005090;
  transition: all 0.8s;
}

#navbar-toggle {
  color: #fff;
  font-size: 1.5rem;
  position: absolute;
  top: 62px;
  right: 35px;
  display: none;
}

.top {
  display: none;
}

/*PIE DE PAGINA*/

footer {
  background-color: #1e1e1e;
  padding: 50px 0 0;
}

#pieDePagina {
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  -webkit-box-pack: center;
  -ms-flex-pack: center;
  justify-content: center;
}

#pieDePagina a i {
  color: #fff;
  font-size: 2rem;
  margin-left: 12px;
  padding: 15px;
}

#copyright {
  text-align: center;
  color: #fff;
  font-size: 1rem;
  padding: 12px 0;
  border-top: 1px solid #423d3d;
  margin: 12px 0 0;
}

/*Estilos de la pagina de inicio de sesion*/

#formulario {
  position: relative;
  width: 560px;
  min-height: 400px;
  background: rgba(255, 255, 255, 0.1);
  border-radius: 10px;
  display: flex;
  justify-content: center;
  align-items: center;
}

```

```

    backdrop-filter: blur(5px);
    box-shadow: 0 25px 45px rgba(0, 0, 0, 0.1);
    border: 1px solid rgba(255, 255, 255, 0.5);
}

#formulario .tituloForm H2 {
    position: relative;
    color: #fff;
    font-size: 20px;
    margin-bottom: 30px;
    margin-top: 15px;
}

#formulario h2::before {
    content: "";
    position: absolute;
    left: 0;
    bottom: -10px;
    width: 399px;
    height: 1px;
    background: rgb(190, 190, 190, 0.5);
}

#formulario .inputBox input {
    width: 80%;
    background: rgba(255, 255, 255, 0.5);
    border: none;
    outline: none;
    margin: 10px 0;
    padding: 10px 20px;
    border-radius: 20px;
    border: 1px solid rgba(255, 255, 255, 0.2);
    color: rgb(0, 0, 0);
    font-size: 15px;
    letter-spacing: 1px;
    box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
}

#formulario .inputBox input::placeholder {
    color: rgba(83, 83, 83, 0.8);
}

/* Boton de ingresar */

#formulario .inputButtonEnviar input[type="submit"] {
    width: 40%;
    background: rgba(255, 255, 255, 0.5);
    border: none;
    outline: none;
    margin: 10px 0;
    padding: 10px 10px;
    border-radius: 20px;
    border: 1px solid rgba(255, 255, 255, 0.2);
    color: rgb(0, 0, 0);
    font-size: 15px;
    background: #fff;
}

#formulario .inputButtonEnviar input[type="submit"]:hover {
    background: #000;
    color: #fff;
    transition: 0.3s;
}

.olvidado {
    color: rgb(255, 255, 255);
}

.olvidado a {
    color: #ffffff;
}

```

```

.olvidado a:hover {
  color: #c7c7c7;
  transition: 500ms;
}

/* Formulario del login */

#formulario label {
  color: #fff;
}

#formulario select {
  width: 80%;
  background: rgba(255, 255, 255, 0.5);
  border: none;
  outline: none;
  margin: 10px 0;
  padding: 10px 20px;
  border-radius: 20px;
  border: 1px solid rgba(255, 255, 255, 0.2);
  color: rgb(0, 0, 0);
  font-size: 15px;
  letter-spacing: 1px;
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
}

#formulario select {
  width: 80%;
  background: rgba(255, 255, 255, 0.5);
  border: none;
  outline: none;
  margin: 10px 0;
  padding: 10px 20px;
  border-radius: 20px;
  border: 1px solid rgba(255, 255, 255, 0.2);
  color: rgb(0, 0, 0);
  font-size: 15px;
  letter-spacing: 1px;
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
}

/*Estilos de la pagina de inicio*/

/*Seccion 1*/

article#home>section:nth-child(1) {
  background-image: -webkit-gradient(linear, left top, left bottom, from(rgba(26, 47, 50, 0.6)),
to(rgba(26, 47, 50, 0.9))), url(..images/portadaFES.jpg);
  background-image: -o-linear-gradient(rgba(26, 47, 50, 0.6), rgba(26, 47, 50, 0.6)),
url(..images/portadaFES.jpg);
  background-image: linear-gradient(rgba(26, 47, 50, 0.6), rgba(26, 47, 50, 0.6)),
url(..images/portadaFES.jpg);
  background-repeat: no-repeat;
  background-position: center center;
  background-size: cover;
  background-attachment: fixed;
}

#seccionUno {
  background-position: center center;
  height: 100%;
  min-height: 850px;
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  -webkit-box-pack: center;
  -ms-flex-pack: center;
  justify-content: center;
  -webkit-box-align: center;
  -ms-flex-align: center;
  align-items: center;
}

```

```

}
#seccionUno h1 {
  text-transform: uppercase;
  color: #fff;
  font-size: 3.5rem;
  text-transform: uppercase;
  font-weight: 700;
  font-family: arial, verdana, sans-serif;
}

/*Animacion de texto principal*/

#centeralign {
  display: table-cell;
  vertical-align: middle;
}

/* Fin de la animacion del primer texto */

#seccionUno p {
  color: #fff;
  font-size: 1.8rem;
  padding: 15px;
  margin-bottom: 0;
  display: none;
}

#seccionUno div {
  text-align: center;
}

/*Seccion Dos*/

#seccionDos {
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  padding: 30px;
}

#seccionDos div {
  padding: 0 0.5rem;
  -ms-flex-item-align: center;
  -ms-grid-row-align: center;
  align-self: center;
  -ms-flex-preferred-size: 50%;
  flex-basis: 50%;
}

#seccionDos img {
  -ms-flex-preferred-size: 50%;
  flex-basis: 50%;
  max-width: 50%;
  -ms-flex-item-align: start;
  align-self: flex-start;
}

/*Seccion 3*/

article>section:nth-child(3) {
  background-image: linear-gradient(rgba(26, 47, 50, 0.5), rgba(26, 47, 50, 0.5)),
url(../images/codigo.jpg);
  background-repeat: no-repeat;
  background-size: cover;
  background-position: center center;
  background-attachment: fixed;
}

```

```

#seccionTres {
  display: flex;
  justify-content: center;
  text-align: center;
  font-size: 2rem;
  height: 360px;
  align-items: center;
  color: #fff;
}

#seccionTres p {
  font-weight: bold;
}

/*Seccion 4*/

article>section:nth-child(4) {
  text-align: center;
  padding: 80px 0 12px;
}

#seccionCuatro {
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  margin: 80px auto;
}

.col {
  margin: 0 0.9em;
}

.col .fas {
  color: #003560;
  font-size: 8rem;
  margin-bottom: 25px;
}

/*Efectos de la tarjeta de la pag de inicio*/

.col:hover {
  transition: all 0.8s;
  filter: opacity(.7);
  -webkit-transform: scale(1.1);
  transform: scale(1.1);
}

.col h3 {
  margin-bottom: 12px;
  color: #3c3c3c;
  font-size: 1.3rem;
  font-weight: 100;
}

.col p {
  font-size: 0.85rem;
  text-align: justify;
}

/*Seccion 5*/

article>section:nth-child(5) {
  background-color: #f3f1fe;
  padding: 50px 0;
}

#seccionQuinta {
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  -webkit-box-pack: justify;
  -ms-flex-pack: justify;
}

```

```

    justify-content: space-between;
    -webkit-box-align: center;
    -ms-flex-align: center;
    align-items: center;
}

/*Estilos de las paginas*/

main header {
    background-color: #eaa522;
    /*Cambiar la barra de color*/
    padding: 15px 0;
    margin-bottom: 40px;
}

main header h1 {
    font-size: 1.9rem;
    color: #fff;
    text-transform: uppercase;
}

#pages {
    display: -webkit-box;
    display: -ms-flexbox;
    display: flex;
    -webkit-box-orient: horizontal;
    -webkit-box-direction: normal;
    -ms-flex-flow: row wrap;
    flex-flow: row wrap;
    overflow: hidden;
    /*margin-bottom: 500px;          CAMBIA LA POSICION DEL FOOTER*/
}

#pages section {
    -ms-flex-preferred-size: 70%;
    flex-basis: 70%;
}

aside {
    -ms-flex-preferred-size: 30%;
    flex-basis: 30%;
}

#pages section {
    -ms-flex-preferred-size: 70%;
    flex-basis: 70%;
    padding-right: 1em;
    text-align: justify;
    margin-bottom: 30px;
}

aside h3 {
    background-color: #f3f1fe;
    padding: 12px 0 12px 0.5em;
    margin-bottom: 20px;
    font-size: 1.20rem;
    font-weight: 100;
}

#socialMedia {
    margin: 0 0 25px 2.3em;
    font-size: 4.5rem;
}

#socialMedia i {
    margin-right: 0.8em;
    color: #3b5998;
}

#socialMedia i:hover {
    transition: all 0.8s;
    color: #2f477a;
}

```

```

    -webkit-transform: scale(1.1);
    transform: scale(1.1);
}

#map iframe {
margin: 0 0 25px 0.5em;
overflow: hidden;
}

.row {
clear: both;
overflow: auto;
margin-bottom: 22px;
}

.row p {
font-size: 1rem;
text-align: justify;
margin-bottom: 20px;
}

/*ICONOS DE LA SECCION DE SERVICIOS*/

.row .fas {
color: #79c1c2;
font-size: 5rem;
/*10 rem para celular*/
margin-bottom: 15px;
}

.row .check-list {
text-decoration: none;
display: inline-block;
font-weight: bold;
}

.row .check-list .fas {
color: #00bb2d;
font-size: 1.5rem;
/*10 rem para celular*/
}

.row .fas:hover {
transition: all 0.8s;
filter: opacity(.7);
-webkit-transform: scale(1.1);
transform: scale(1.1);
}

/*Fin de los iconos de la seccion de servicios*/

form {
margin-right: 1em;
}

.asterisco {
color: #eaa522;
}

label {
display: block;
margin: 10px 0;
font-size: 0.98rem;
}

/* input[type="text"], input[type="email"] {
display: inline-block;
width: 100%;
height: 45px;
margin: 5px 0;
border: 1px solid #ccc;
} */

```

```

textarea#comentario {
  width: 100%;
  border: 1px solid #ccc;
  margin-bottom: 30px;
}

/* input[type="submit"] {
  background-color: #00bb2d;
  display: inline-block;
  padding: 10px 1em;
  width: 100%;
  border: none;
  color: #fff;
  font-size: 1rem;
} */

/*MEDIA QUERYS*/

@media screen and (max-width: 1199) {
  body {
    font-size: 0.95rem;
  }
}

/*Aplicar los siguientes estilos cuando el viewport tenga un tamaño entre 960 y 1199 pixeles*/

@media screen and (min-width: 960px) and (max-width: 1199px) {
  #seccionUno h1 {
    font-size: 2.5rem;
  }
  #seccionUno p {
    font-size: 1.25rem;
  }
  h2 {
    font-size: 1.3rem;
  }
  #seccionTres img {
    -ms-flex-preferred-size: 45%;
    flex-basis: 45%;
    max-width: 45%;
  }
  #seccionTres div {
    -ms-flex-preferred-size: 55%;
    flex-basis: 55%;
    padding-left: 2em;
    padding-right: 2em;
  }
  .col p {
    font-size: 0.85rem;
    text-align: center;
  }
  #pages {
    margin-bottom: 12px;
  }
  aside ul li a {
    font-size: 0.95rem;
  }
  .row {
    margin: 0 1em 25px 0;
  }
  .row img {
    float: none;
    width: 100%;
  }
  .row i {
    margin-bottom: 20px;
    margin-right: 1em;
  }
  #pages.row:not(:last-child) {
    border-bottom: 1px solid #ccc;
  }
}

```

```

.transparent {
    margin-bottom: 30px;
}
}

/*Aplicar los siguientes estilos cuando el viewport tenga un tamaño entre menor de 960 pixeles*/

@media screen and (max-width: 959px) {
    #navigation {
        display: none;
        -webkit-box-orient: vertical;
        -webkit-box-direction: normal;
        -ms-flex-direction: column;
        flex-direction: column;
    }
    #main-nav {
        display: none;
        -webkit-box-orient: vertical;
        -webkit-box-direction: normal;
        -ms-flex-direction: column;
        flex-direction: column;
    }
    #navbar-toggle {
        display: block;
    }
    #main-nav li a {
        display: block;
        padding: 20px 1em;
    }
    #main-nav li:not(:last-child) {
        border-bottom: 1px solid #353535;
    }
    /* Mostrar menu desplegable*/
    .tooltip {
        font-weight: initial;
        color: #fff;
    }
    .tooltip:hover .tooltip-box {
        display: contents;
    }
    .tooltip-box {
        display: contents;
    }
    /*No mostrar ambos telefonos*/
    #main-nav-tel {
        display: none;
    }
    .top {
        background-color: #2F2935;
        color: #aeaeae;
        padding: 10px;
        font-size: 1rem;
        display: block;
    }
    .top ul {
        display: -webkit-box;
        display: -ms-flexbox;
        display: flex;
        -webkit-box-pack: center;
        -ms-flex-pack: center;
        justify-content: center;
        list-style: none;
    }
    /* Estilos de la pagina de inicio */
    #seccionUno h1 {
        font-size: 2.5rem;
    }
    #seccionUno p {
        font-size: 1.25rem;
    }
    #seccionDos, #seccionTres, #seccionCuatro, #seccionCinco {
        -webkit-box-orient: vertical;

```

```

        -webkit-box-direction: normal;
        -ms-flex-direction: column;
        flex-direction: column;
    }
    #seccionDos div, #seccionTres div {
        padding: 0;
        -ms-flex-item-align: normal;
        -ms-grid-row-align: normal;
        align-self: normal;
        -ms-flex-preferred-size: 100%;
        flex-basis: 100%;
        text-align: center;
    }
    .btn, .transparent {
        width: 100%;
    }
    .transparent {
        margin-bottom: 20px;
    }
    #seccionDos img, #seccionTres img {
        display: none;
    }
    #seccionTres.container {
        width: 90%;
        padding: 20px;
    }
    article>section:nth-child(4) {
        padding: 35px 0 12px 0;
    }
    #seccionCuatro {
        margin: 35px auto;
    }
    .col {
        margin: 20px 0.9em 30px;
        border: 1px solid #eaa522;
        padding: 30px 2em;
    }
    #seccionCinco {
        text-align: center;
    }
    /*Estilos del resto de las paginas*/
    #pages section {
        -ms-flex-preferred-size: 100%;
        flex-basis: 100%;
        padding-right: 0;
    }
    aside {
        -ms-flex-preferred-size: 100%;
        flex-basis: 100%;
    }
    .row img {
        float: none;
        width: 100%;
    }
    #pages .row:not(:last-child) {
        border-bottom: 1px solid #ccc;
    }
    .row i {
        margin-right: 1em;
    }
    #seccionTres p {
        font-size: 1.2rem;
    }
}

/*Media Query para el pie de pagina*/

/*Aplicar cuando el viewport sea igual o menor a 659 px*/
@media screen and (max-width: 659px) {
    #pieDePagina {
        -webkit-box-orient: vertical;
    }
}

```

```

        -webkit-box-direction: normal;
        -ms-flex-direction: initial;
        flex-direction: initial;
        -webkit-box-align: center;
        -ms-flex-align: center;
        align-items: center;
        text-align: center;
    }
    .top ul {
        display: -webkit-box;
        display: -ms-flexbox;
        display: flex;
        -webkit-box-pack: center;
        -ms-flex-pack: center;
        justify-content: center;
        list-style: none;
    }
}

/*Aplicar cuando el viewport sea igual o menor a 407 px*/
@media screen and (max-width: 515px) {
    .top ul {
        display: -webkit-box;
        display: -ms-flexbox;
        display: flex;
        -webkit-box-pack: center;
        -ms-flex-pack: center;
        justify-content: center;
        list-style: none;
    }
    #contactoList {
        font-size: 1rem;
    }
    #formulario {
        width: 400px;
    }
}

/*Aplicar cuando el viewport sea igual o menor a 407 px*/
@media screen and (max-width: 407px) {
    .top {
        font-size: 0.8rem;
    }
    #contactoList {
        font-size: 0.75rem;
    }
}

/*Aplicar cuando el viewport sea igual o menor a 362 px*/
@media screen and (max-width: 370px) {
    .top {
        font-size: 0.70rem;
    }
}

/*KeyFrame de texto principal*/
@keyframes move {
    0% {
        top: 0px;
    }
    20% {
        top: -50px;
    }
    40% {
        top: -100px;
    }
    60% {
        top: -150px;
    }
}

```

```
    }  
    80% {  
        top: -200px;  
    }  
}  
  
@-webkit-keyframes move {  
    0% {  
        top: 0px;  
    }  
    20% {  
        top: -50px;  
    }  
    40% {  
        top: -100px;  
    }  
    60% {  
        top: -150px;  
    }  
    80% {  
        top: -200px;  
    }  
}  
  
@-moz-keyframes move {  
    0% {  
        top: 0px;  
    }  
    20% {  
        top: -50px;  
    }  
    40% {  
        top: -100px;  
    }  
    60% {  
        top: -150px;  
    }  
    80% {  
        top: -200px;  
    }  
}  
  
@-o-keyframes move {  
    0% {  
        top: 0px;  
    }  
    20% {  
        top: -50px;  
    }  
    40% {  
        top: -100px;  
    }  
    60% {  
        top: -150px;  
    }  
    80% {  
        top: -200px;  
    }  
}  
  
@keyframes move {  
    0% {  
        top: 0px;  
    }  
    20% {  
        top: -50px;  
    }  
    40% {  
        top: -100px;  
    }  
    60% {  
        top: -150px;  
    }  
}
```

```
    }  
    80% {  
      top: -200px;  
    }  
  }  
}
```

Referencias.

Referencias de libros electrónicos:

1. Ávila Jiménez, J. L. (2015). El ciclo de vida del desarrollo de aplicaciones. España: Elearning S.L.
2. Fábrega, J. L. (2005). Gerencia de Proyectos de Tecnología de la Información. Caracas: Los Libros de El Nacional.
3. Ishikawa, K. (1997). *Introducción Al control de calidad*. Diaz de Santos.
4. Naur, P. y Randel B. (1985). Software Engineering: Report of a Conference Sponsored by the NATO Science Committee. EE.UU: Nato.
5. Pons, Claudia Desarrollo de software dirigido por modelos: conceptos teóricos y su aplicación práctica / Claudia Pons; Roxana Giandini; Gabriela Pérez. - 1a ed. - La Plata: Universidad Nacional de La Plata, 2010.
6. Portillo, D. D. (2015). Gestión de la calidad de productos editoriales multimedia. ARGN0110. Málaga: IC Editorial.
7. Pressman, R. S. (2010). Ingeniería de software. Connecticut: Mc Graw Hill.
8. Software. (2018). Disquisición. En *Diccionario de la lengua española* (Actualización 2018). Recuperado desde: <https://dle.rae.es/software>
9. Sommerville, I. (2005). *Ingeniería del software* (M. I. A. Galipienso, A. Botia Martínez, F. Mora Lizan, & J. P. Trigueros Jover, Trans.; 7th ed.). Pearson Educación.
10. Valdéz, J. L. (2014). Implementación del modelo integral colaborativo (MDSIC) como fuente de innovación para el desarrollo ágil de software en las empresas de la zona centro - occidente en México. (Tesis de doctorado). Universidad Popular Autónoma del Estado de Puebla, Puebla, México.

Referencias electrónicas:

1. Abdallah Aberouch. (2015). 5 bugs informáticos que marcaron la historia. Recuperado desde: <https://www.bbvaopenmind.com/tecnologia/innovacion/5-bugs-informaticos-que-marcaron-la-historia/>
2. C. (2019). CSS, ¿qué es?. Arumeinformatica. Recuperado desde: <https://www.arumeinformatica.es/blog/css/>
3. CAVSI. (2012). ¿Qué es Software del Sistema? Foro Técnico. Recuperado desde: <https://www.cavsi.com/preguntasrespuestas/que-es-software-del-sistema>
4. CAVSI. (2015). ¿Qué es Software de Aplicación? Definición y Tipos. Foro Técnico. Recuperado desde: <https://www.cavsi.com/preguntasrespuestas/que-es-software-de-aplicacion/>
5. Chain, R. E. S. (2020). *Guía PMBOK: definición, estructura y tips de estudio*. EAE Business School. Recuperado desde: <https://retos-operaciones-logistica.eae.es/que-es-la-guia-pmbok-y-como-influye-en-la-administracion-de-proyectos/>
6. Chuburu, L. (2020). *Qué es JQuery y cómo implementarlo*. Laura Chuburu. Recuperado desde: <https://www.laurachuburu.com.ar/tutoriales/que-es-jquery-y-como-implementarlo.php>
7. Colaboradores de PHP. (2021). ¿Qué es PHP?. PHP.net. Recuperado desde: <https://www.php.net/manual/es/intro-what-is.php>
8. Colaboradores de Wikipedia. (2021). *Desarrollo ágil de software*. Wikipedia, la enciclopedia libre. Recuperado desde: https://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software
9. Colaboradores de Wikipedia. (2021). *Metodología*. Wikipedia, la enciclopedia libre. Recuperado desde: <https://es.wikipedia.org/wiki/Metodolog%C3%ADa>
10. Colaboradores de Wikipedia. (2021). *PHP*. Wikipedia, la enciclopedia libre. Recuperado desde: <https://es.wikipedia.org/wiki/PHP>
11. Colaboradores Heflo. (2017). *Metodología ágil*. HEFLO ES. Recuperado desde: <https://www.heflo.com/es/definiciones/metodologia-agil/>
12. ConceptoDefinicion.de, Redacción. (2019). *Definición de Página Web*. ConceptoDefinición. Recuperado desde: <https://conceptoDefinicion.de/pagina-web/>
13. Cruz J. (S.F.) *Historia de la calidad*. EXCELLENTIA. Recuperado desde: http://docencia.fca.unam.mx/~lvaldes/cal_pdf/cal5.pdf

14. DesarrolloWeb. (2002). CSS. DesarrolloWeb. Recuperado desde: <https://desarrolloweb.com/home/css>
15. Díaz, G. (2016). *¿Qué es el PMBOK?* Creación de Proyectos. Recuperado desde: <https://www.creaciondeproyectos.com/el-pmbok/>
16. EcuRed. (2013). Métricas para la calidad del software. Recuperado desde: https://www.ecured.cu/index.php?title=Metricas_para_la_calidad_del_software&action=history
17. Ernesto, S. R. (Sin fecha). Aseguramiento de la Calidad. Recuperado desde: <https://slideplayer.es/slide/10163092/>
18. ESAN Graduate School of Business. (2018). *La importancia del PMBOK y su influencia en un proyecto*. ESAN. Recuperado desde: <https://www.esan.edu.pe/apuntes-empresariales/2018/10/la-importancia-del-pmbok-y-su-influencia-en-un-proyecto/>
19. Escuela Técnica Superior de Ingeniería de Sistemas Informáticos. (s.f.). Objetivos de Ingeniería del Software. Universidad Politécnica de Madrid. Recuperado desde: <http://www.etsisi.upm.es/estudios/grados/software/objetivos>
20. Finding Passion for Quality. (2015). SQA (Aseguramiento de la calidad del software). Obtenido de SQA (Aseguramiento de la calidad del software): Recuperado desde: <https://findingtc.com/sqa-aseguramiento-de-la-calidad-del-software/>
21. G. (2019). *Ingeniería de software: Qué es, objetivos, características y más*. Carreras Universitarias. Recuperado desde: <https://micarrerauniversitaria.com/c-ingenieria/ingenieria-de-software/>
22. Gandarillas, A. (27 de Julio de 2017). Metodologías para el desarrollo y mantenimiento de software y sistemas de información. Recuperado desde: <https://metodologia.es/iso-12207/>
23. García, O. (2020). *5 grupos de procesos y 9 áreas de conocimiento + 1*. Proyectum. Recuperado desde: <https://www.proyectum.com/sistema/blog/5-grupos-de-procesos-y-9-areas-de-conocimiento-1/>
24. Garzas, J. (2013). Top 8 de errores informáticos más costosos de la historia. Recuperado desde: <https://www.javiergarzas.com/2013/05/top-7-de-errores-informaticos.html>
25. Gascón, O. (2016). *Identificar a los interesados*. TodoPMP. Recuperado desde: <http://todopmp.com/identificar-los-interesados/>
26. Gascón, O. (2017) *Controlar los riesgos*. TodoPMP. Recuperado desde: <http://todopmp.com/controlar-los-riesgos/>

27. Gascón, O. (2017) *Realizar el aseguramiento de la calidad*. TodoPMP. Recuperado desde: <http://todopmp.com/realizar-aseguramiento-la-calidad/>
28. Gascón, O. (2017). *Adquirir al equipo del proyecto*. TodoPMP. Recuperado desde: <http://todopmp.com/adquirir-al-equipo-del-proyecto/>
29. Gascón, O. (2017). *Cerrar el proyecto o fase*. TodoPMP. Recuperado desde: <http://todopmp.com/cerrar-proyecto-fase/>
30. Gascón, O. (2017). *Controlar el cronograma*. TodoPMP. Recuperado desde: <http://todopmp.com/controlar-el-cronograma/>
31. Gascón, O. (2017). *Controlar la calidad*. TodoPMP. Recuperado desde: <http://todopmp.com/controlar-la-calidad/>
32. Gascón, O. (2017). *Controlar la participación de los interesados*. TodoPMP. Recuperado desde: <http://todopmp.com/controlar-la-participacion-los-interesados/>
33. Gascón, O. (2017). *Controlar las adquisiciones*. TodoPMP. Recuperado desde: <http://todopmp.com/controlar-las-adquisiciones/>
34. Gascón, O. (2017). *Controlar las comunicaciones*. TodoPMP. Recuperado desde: <http://todopmp.com/controlar-las-comunicaciones/>
35. Gascón, O. (2017). *Controlar los costos (Valor ganado)*. TodoPMP. Recuperado desde: <http://todopmp.com/controlar-los-costos/>
36. Gascón, O. (2017). *Desarrollar el cronograma*. TodoPMP. Recuperado desde: <http://todopmp.com/desarrollar-el-cronograma/>
37. Gascón, O. (2017). *Dirigir el equipo del proyecto*. TodoPMP. Recuperado desde: <http://todopmp.com/dirigir-el-equipo-del-proyecto/>
38. Gascón, O. (2017). *Efectuar las adquisiciones*. TodoPMP. Recuperado desde: <http://todopmp.com/efectuar-las-adquisiciones/>
39. Gascón, O. (2017). *Gestionar la participación de los interesados*. TodoPMP. Recuperado desde: <http://todopmp.com/gestionar-la-participacion-los-interesados/>
40. Gascón, O. (2017). *Gestionar las comunicaciones*. TodoPMP. Recuperado desde: <http://todopmp.com/gestionar-las-comunicaciones/>
41. Gascón, O. (2017). *Monitorear el trabajo del proyecto y realizar el control integrado de cambios, ¿cuál es la diferencia?* TodoPMP. Recuperado desde: <http://todopmp.com/monitorear-controlar-trabajo-del-proyecto/>
42. Gascón, O. (2017). *Recopilar requisitos*. TodoPMP. Recuperado desde: <http://todopmp.com/recopilar-requisitos/>

43. Gascón, O. (2017). *Secuenciar las actividades*. TodoPMP. Recuperado desde: <http://todopmp.com/secuenciar-las-actividades/>
44. Gascón, O. (2017). *Validar el alcance*. TodoPMP. Recuperado desde: <http://todopmp.com/validar-el-alcance/>
45. Gascón, O. (2018). *Estimar los recursos de las actividades*. TodoPMP. Recuperado desde: <http://todopmp.com/estimar-los-recursos-las-actividades/>
46. Gascón, O. (2020) *Planificar la gestión de las adquisiciones*. TodoPMP. Recuperado desde: <http://todopmp.com/planificar-la-gestion-de-las-adquisiciones/>
47. Gascón, O. (2020). *Crear la EDT/WBS*. TodoPMP. Recuperado desde: <http://todopmp.com/crear-la-edt-wbs/>
48. Gascón, O. (2020). *Definir el alcance*. TodoPMP. Recuperado desde: <http://todopmp.com/definir-el-alcance/>
49. Gascón, O. (2020). *Definir las actividades*. TodoPMP. Recuperado desde: <http://todopmp.com/definir-las-actividades/>
50. Gascón, O. (2020). *Desarrollar el acta de constitución del proyecto*. TodoPMP. Recuperado desde: <http://todopmp.com/desarrollar-acta-constitucion-del-proyecto/>
51. Gascón, O. (2020). *Desarrollar el acta de constitución del proyecto*. TodoPMP. Recuperado desde: <http://todopmp.com/desarrollar-acta-constitucion-del-proyecto/>
52. Gascón, O. (2020). *Desarrollar el plan para la dirección de proyectos*. TodoPMP. Recuperado desde: <http://todopmp.com/desarrollar-plan-la-direccion-proyectos/>
53. Gascón, O. (2020). *Determinar el presupuesto*. TodoPMP. Recuperado desde: <http://todopmp.com/determinar-el-presupuesto/>
54. Gascón, O. (2020). *Dirigir y gestionar el trabajo del proyecto*. TodoPMP. Recuperado desde: <http://todopmp.com/dirigir-y-gestionar-el-trabajo-del-proyecto/>
55. Gascón, O. (2020). *Estimar la duración de las actividades*. TodoPMP. Recuperado desde: <http://todopmp.com/estimar-la-duracion-las-actividades/>
56. Gascón, O. (2020). *Estimar los costos*. TodoPMP. Recuperado desde: <http://todopmp.com/estimar-los-costos/>
57. Gascón, O. (2020). *Gestionar el conocimiento del proyecto*. TodoPMP. Recuperado desde: <https://todopmp.com/gestionar-conocimiento-del-proyecto/>
58. Gascón, O. (2020). *Planificar la gestión de la calidad*. TodoPMP. Recuperado desde: <http://todopmp.com/planificar-la-gestion-la-calidad/>

59. Gascón, O. (2020). *Planificar la gestión de las comunicaciones*. TodoPMP. Recuperado desde: <http://todopmp.com/planificar-la-gestion-las-comunicaciones/>
60. Gascón, O. (2020). *Planificar la gestión de los costos*. TodoPMP. Recuperado desde: <http://todopmp.com/planificar-la-gestion-los-costos/>
61. Gascón, O. (2020). *Planificar la gestión de los interesados*. TodoPMP. Recuperado desde: <http://todopmp.com/planificar-la-gestion-los-interesados/>
62. Gascón, O. (2020). *Planificar la gestión de los recursos*. TodoPMP. Recuperado desde: <http://todopmp.com/planificar-la-gestion-los-recursos-humanos/>
63. Gascón, O. (2020). *Planificar la gestión de los riesgos*. TodoPMP. Recuperado desde: <http://todopmp.com/planificar-la-gestion-los-riesgos/>
64. Gascón, O. (2020). *Planificar la gestión del alcance*. TodoPMP. Recuperado desde: <http://todopmp.com/planificar-la-gestion-del-alcance/>
65. Gascón, O. (2020). *Planificar la gestión del cronograma*. TodoPMP. Recuperado desde: <http://todopmp.com/planificar-la-gestion-del-cronograma/>
66. Gascón, O. (2020). *Planificar la respuesta a los riesgos*. TodoPMP. Recuperado desde: <http://todopmp.com/planificar-la-respuesta-los-riesgos/>
67. Gascón, O. (2017). *Controlar el alcance*. TodoPMP. Recuperado desde: <http://todopmp.com/controlar-el-alcance/>
68. Gbegnedji, G. (2017). 11.2. *Identificar los Riesgos del Proyecto*. Project Management. Gladys Gbegnedji. Recuperado desde: <https://www.gladysgbegnedji.com/identificar-los-riesgos-del-proyecto/>
69. Gbegnedji, G. (2017). 11.3. *Realizar el Análisis Cualitativo de Riesgos*. Project Management. Gladys Gbegnedji. Recuperado desde: <https://www.gladysgbegnedji.com/realizar-el-analisis-cualitativo-de-riesgos/>
70. Gbegnedji, G. (2017). 11.4. *Realizar el Análisis Cuantitativo de Riesgos*. Project Management. Gladys Gbegnedji. Recuperado desde: <https://www.gladysgbegnedji.com/realizar-el-analisis-cuantitativo-de-riesgos/>
71. Gbegnedji, G. (2017). 5.6. *Controlar el Alcance*. Project Management. Gladys Gbegnedji. Recuperado desde: <https://www.gladysgbegnedji.com/controlar-el-alcance/>
72. Global STD Certification. (16 de Junio de 2019). Global STD. Recuperado desde: <https://www.globalstd.com/networks/blog/guia-iso-26000-responsabilidad-social>

73. Gómez, J. (16 de Noviembre de 2012). Laboratorio TI. Recuperado desde: <https://www.laboratorioti.com/2012/11/16/6-desventajas-clave-de-las-metricas-tecnicas-frente-a-las-metricas-funcionales/>
74. González, E. (2009). *Cómo obtener claves o contraseñas de redes wifi (cracking «a por naranjas») ¿Es segura una red inalámbrica?* aprenderaprogramar.com. Recuperado desde: https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193
75. Guía de los fundamentos para la dirección de proyectos. (2019, 26 de octubre). Wikipedia, La enciclopedia libre. Recuperado desde: https://es.wikipedia.org/w/index.php?title=Gu%C3%ADa_de_los_fundamentos_para_la_direcci%C3%B3n_de_proyectos&oldid=120787097.
76. Gutimar, Soluciones Informáticas. (2012). Team Software Process (TSP) y Team Process Software (TSP). Recuperado desde: <https://gutimarsoluciones.wordpress.com/2012/05/09/team-software-process-tsp-y-team-process-software-tsp/>
77. ISOTools Excellence. (15 de Marzo de 2015). ISOTools. Recuperado desde: <https://www.isotools.org/2015/03/19/que-son-las-normas-iso-y-cual-es-su-finalidad/>
78. La oficina de proyectos de la informática. (09 de Abril de 2018). PMOInformatica.com. Recuperado desde: <http://www.pmoinformatica.com/2018/04/metricas-del-software-ventajas.html>
79. Manual de jQuery. (s/f). Desarrolloweb.com. Recuperado el 27 de septiembre de 2021, de <https://desarrolloweb.com/manuales/manual-jquery.html>
80. MDN contributors. (2021). *JavaScript | MDN*. MDN Web Docs. Recuperado desde: <https://developer.mozilla.org/es/docs/Web/JavaScript>
81. MDN. (2020). *HTML: Lenguaje de etiquetas de hipertexto*. Documentación web de MDN. Recuperado desde: <https://developer.mozilla.org/es/docs/Web/HTML>
82. Medina Prieto, R. (24 de Julio de 2017). Modelos y estándares de calidad aplicados al sistema de información. Recuperado desde: <http://unidad4rociomp.blogspot.com/2017/07/46.html>
83. Merma. (2019, 3 de noviembre). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 01:55, noviembre 28, 2019. Recuperado desde: <https://es.wikipedia.org/w/index.php?title=Merma&oldid=121035092>.
84. Muradas, Y. (2018). *Conoce las 3 metodologías ágiles más usadas*. OpenWebinars.net. Recuperado desde: <https://openwebinars.net/blog/conoce-las-3-metodologias-agiles->

[mas-usadas/](#)

85. Nirian, P. (2021). *Control de calidad*. Economipedia. Recuperado desde: <https://economipedia.com/definiciones/control-de-calidad.html>
86. Pérez y Gardey. (2009). *Definición de ingeniería de software*. Definicion.de. Recuperado desde: <https://definicion.de/ingenieria-de-software/>
87. Pérez y Merino. (2009) *Definición de página web*. Definicion.de. Recuperado desde: <https://definicion.de/pagina-web/>
88. Pérez, D. (2018). Cambios en el área Gestión de los Recursos del PMBOK. BPMSAT. Recuperado desde: <https://www.bpmsat.com/cambios-en-el-area-gestion-de-los-recursos-del-pmbok/>
89. Pérez, M. (2021). *Metodología*. Concepto de - Definición de. Recuperado desde: <https://conceptodefinicion.de/metodologia/>
90. Personal Software Process. (2019, 14 de noviembre). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 01:08, noviembre 28, 2019. Recuperado desde: https://es.wikipedia.org/w/index.php?title=Personal_Software_Process&oldid=121303720
91. Piattini, M. (2016). Evolución de la Ingeniería del Software y la formación de profesionales. *Bit & Byte, año 2*. (Número 4). Recuperado desde: http://sedici.unlp.edu.ar/bitstream/handle/10915/57358/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y
92. PMOInformatica. (2018). Gestión de riesgos en el PMBOK 6. PMOinformatica.com Recuperado desde: <http://www.pmoinformatica.com/2018/03/gestion-riesgos-PMBOK-6.html>
93. R., J. (2019). *Software de aplicación: Qué es, ejemplos y características*. 247 Tecno. Recuperado desde: <https://247tecno.com/software-de-aplicacion-ejemplos-caracteristicas/>
94. Raffino, M. (2020) *Software de Aplicación*. Concepto.de. Recuperado desde: <https://concepto.de/software-de-aplicacion/>
95. Raffino, M. (2020). *Página Web*. Concepto.de. Recuperado desde: <https://concepto.de/pagina-web/>
96. Raffino, M. (2021). *Metodología - Concepto, tipos, investigación y método*. Concepto.de. Recuperado desde: <https://concepto.de/metodologia/>

97. Red Hat. (2021). *¿Qué es la metodología ágil?*. redhat.com. Recuperado desde: <https://www.redhat.com/es/devops/what-is-agile-methodology>
98. Red Hat. (2021). *¿Qué es la metodología ágil?*. redhat.com. Recuperado desde: <https://www.redhat.com/es/devops/what-is-agile-methodology>
99. Resultados de búsqueda Fragmento destacado de la Web Normalización y Certificación Electrónica, A.C. (2016). MoProSoft. Recuperado desde: <https://www.nyce.org.mx/respaldo/moprosoft-nyce/>
100. Rodríguez, C. (2015). Cuadro Comparativo de las Normas y Estándares para Proyectos de TI y para Desarrollo de Software. Recuperado desde: <http://blogsistemasdecalidad.blogspot.com/2015/02/normasy-estandares-en-proyectos-de-ti-y.html>
101. Roselló, V. (2021). *Qué son las metodologías ágiles y cuáles son sus ventajas empresariales.* iebsschool.com. Recuperado desde: <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>
102. Russia Today. (2014). 10 errores informáticos que provocaron catástrofes. Recuperado desde: <https://actualidad.rt.com/actualidad/view/138158-catastrofes-programacion-culpa-software-computadora>
103. S. (2019). *Significado de Software.* Significados. Recuperado desde: <https://www.significados.com/software/>
104. Salgado, G. (2019). MoProSoft: Un modelo para mejorar la calidad del software en México. Recuperado desde: <http://conogasi.org/articulos/moprosoft-un-modelo-para-mejorar-la-calidad-del-software-en-mexico/>
105. Sfc. TM, J. P. S. (2016). *Los 49 procesos del PMBOK® 6(sin tener que memorizar).* Administrador de Proyectos. <https://medium.com/administrador-de-proyectos/los-47-procesos-del-pmbok-5-sin-tener-que-memorizar-befdde74024>
106. T. (2018). *Ingeniería de software timeline.* Timetoast. Recuperado desde: <https://www.timetoast.com/timelines/ingenieria-de-software-42b32522-2a88-4abd-8379-7f69ac4c0b29>
107. Team Software Process. (2019, 15 de julio). *Wikipedia, La enciclopedia libre.* Fecha de consulta: 22:02, noviembre 27, 2019. Recuperado desde: https://es.wikipedia.org/w/index.php?title=Team_Software_Process&oldid=117439224.
108. Tecnología, C. L. N. Y. (2019). *¿Qué son las metodologías ágiles?*. LN Creatividad y Tecnología Blog. Recuperado desde: <https://www.luisan.net/blog/transformacion-digital/que-son-las-metodologias-agiles>

109. TecnoMagazine. (2019). *Software de Sistema*. TecnoMagazine. Recuperado desde: <https://tecnomagazine.net/2019/05/17/software-de-sistema/>
110. Torres, I. (2020). *Cómo realizar un Control de Calidad a Tu Producto o Servicio*. IVE Consultores. Recuperado desde: <https://iveconsultores.com/control-de-calidad/>
111. Vargas, A. (10 de Octubre de 2010). Ingeniería del software. Recuperado desde: http://arielvargasu.blogspot.com/2010/10/garantia-de-calidad-de-software-sqa_18.html
112. Wikipedia. (2020). *Hoja de estilos en cascada*. Wikipedia, la enciclopedia libre. Recuperado desde: https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada#Historia
113. Wikipedia. (2020). *JQuery*. Wikipedia, la enciclopedia libre. Recuperado desde: <https://es.wikipedia.org/wiki/JQuery>
114. Wikipedia. (2020). *Página web*. Wikipedia, la enciclopedia libre. Recuperado desde: https://es.wikipedia.org/wiki/P%C3%A1gina_web#cite_note-1