



UNIVERSIDAD POLITÉCNICA DE SINALOA
PROGRAMA ACADÉMICO DE
INGENIERÍA EN INFORMÁTICA

Tesina

“Optimización en el desarrollo de aplicaciones web o de escritorio mediante el uso de Frameworks”

Para obtener la acreditación de las estadías profesionales y contar con los créditos para el grado de Ingeniero en Informática.

Autor: Rendón Rendón Pedro Antonio

Asesor: Luis Javier Mena Camare

Asesor OR: Omar Vidaña Peraza

Mazatlán, Sinaloa 13 de Diciembre del 2019

Carta de aceptación



Protegemos tu capital.
Fortalecemos tu empresa.

Mazatlán, Sinaloa, 02 de Septiembre de 2019.

LIC. NUBIA VANESSA FELIX VALENZUELA
Jefa del Departamento de Vinculación
UNIVERSIDAD POLITECNICA DE SINALOA
Presente.

La presente es con el fin de saludarlo y al mismo tiempo hacer de su conocimiento que el **C. PEDRO ANTONIO RENDÓN RENDÓN** de la Carrera de **INGENIERÍA EN INFORMÁTICA**, con número de matrícula **2016030039** fue aceptado para realizar sus estancias académicas profesionales durante el periodo del **02 de Septiembre de 2019 al 06 de Diciembre del 2019**, acumulando un total de **600 hrs.**, en horario de Lunes a Viernes **9:00 a.m. a 5:00 p.m.** y Sábados de **8:00 a.m. a 1:00 p.m.**, integrándose en nuestra Firma **UPC**, en el departamento de **SISTEMAS**.

Por lo anterior quedamos agradecidos con ustedes por proporcionarnos este tipo de apoyo, quedando a sus respetables órdenes.

ATENTAMENTE



LIC. PERLA MARIXA ZATARAIN BARRIOS
ENCARGADO RECURSOS HUMANOS

Sierra Rumorosa No. 321 Fracc. Lomas de Mazatlán
CP 82110 Mazatlán, Sinaloa, México. Tel. (669) 112 0520
www.upc.tax

Carta de aceptación de tema de tesina



UNIVERSIDAD POLITÉCNICA DE SINALOA



C. RENDON RENDON PEDRO ANTONIO
Presente.-

Folio 2016030039-2019-46

Por medio de la presente me permito comunicarle que **es de aceptarse el tema de tesina**, el cuál se ha solicitado bajo el título:

“Optimización en el desarrollo de aplicaciones web o de escritorio mediante el uso de Frameworks”

mismo que usted desarrollará con objeto de dar lugar a los tramites conducentes para la acreditación de la asignatura de Estadías Profesionales de la Unidad Académica de:

Ingeniería en Informática

Así mismo. Le comunico que para el desarrollo de la citada tesina le ha sido asignado como director de la misma a: **Luis Javier Mena Camare**, y como asesores a Ramon Patricio Velazquez Cuadras, **Patricia Guadalupe Torrero Martinez**, y **Alejandro Perez Pasten Borja**.

Sin otro particular por el momento, aprovecho la ocasión para enviarle un cordial saludo.

Atentamente

Dr. Rodolfo Ostos Robles

Director del Programa Académico de Ingeniería en Informática

Universidad Politécnica de Sinaloa
“2019, Año del Caudillo del Sur, Emiliano Zapata”.



Carretera Municipal Libre Mazatlán Higuera Km. 3, Col. Genaro Estrada. C.P. 82199, Mazatlán, Sin. Tel (669) 1800695 y 96

www.upsin.edu.mx

Carta de aceptación de revisión de tesina



UNIVERSIDAD POLITÉCNICA DE SINALOA



C. RENDON RENDON PEDRO ANTONIO
Presente.-

Folio 2016030039-2019-46

Por este conducto le envío un cordial saludo y aprovecho la ocasión para notificarle que el jurado que le fue asignado para evaluar la tesina desarrollada en las estadias profesionales denominada "Optimización en el desarrollo de aplicaciones web o de escritorio mediante el uso de Frameworks" y que después de ser revisada en reunión de sinodales, ante la Dirección de la Unidad Académica de Ingeniería en Informática, integrada por:

PRESIDENTE DEL JURADO: Luis Javier Mena Camare

SINODAL: Ramon Patricio Velazquez Cuadras

SINODAL: Patricia Guadalupe Torrero Martínez

SINODAL: Alejandro Perez Pasten Borja

Ha decidido autorizar y aceptar la digitalización de la misma por el participante, conforme a la normatividad vigente y cumpliendo con los requisitos para tal caso.

Agradeciendo la atención a la presente, le reitero a Usted mi atenta consideración y respeto.



Dr. Rodolfo Ostos Robles

Director del Programa Académico de Ingeniería en Informática
Universidad Politécnica de Sinaloa



"2019, Año del Caudillo del Sur, Emiliano Zapata".



Carretera Municipal Libre Mazatlán Higuera Km. 3, Col. Genaro Estrada. C.P. 82199, Mazatlán, Sin. Tel (669) 1800695 y 96
www.upsin.edu.mx

Carta de terminación



Protegemos tu capital,
Fortalecemos tu empresa.

Mazatlán, Sinaloa, 06 de Diciembre de 2019.

LIC. NUBIA VANESSA FELIX VALENZUELA
Jefa del Departamento de Vinculación
UNIVERSIDAD POLITECNICA DE SINALOA
Presente.

La presente es con el fin de saludarla y al mismo tiempo hacer de su conocimiento que el **C. PEDRO ANTONIO RENDÓN RENDÓN** quien cursa el décimo semestre de **INGENIERIA EN INFORMÁTICA**, ha culminado satisfactoriamente sus **Prácticas Profesionales**, acumulando un total de **600 hrs.** de trabajo, en el Departamento de **SISTEMAS**.

Por lo anterior quedamos agradecidos con ustedes por proporcionarnos este tipo de apoyo, quedando a sus respetables órdenes.

ATENTAMENTE



CONSULTORIA PROFESIONAL

LIC. PERLA MARIKA ZATARAIN BARRIOS
ENCARGADO RECURSOS HUMANOS

Sierra Rumorosa No. 321 Fracc. Lomas de Mazatlán
CP 82110 Mazatlán, Sinaloa, México. Tel. (669) 112 0520
www.upc.tax

Agradecimientos

Agradezco profundamente a mi familia la cual me brindó todo su apoyo durante mis estudios, por ser esa parte fundamental para mí que me daba ánimos de seguir a delante y nunca rendirme ante los retos que enfrentaba, de igual manera agradezco todos sus esfuerzos para que yo siguiera y cumpliera mis sueños.

ÍNDICE TEMÁTICO

Índice de figuras.....	9
Resumen.....	10
Abstract.....	10
Introducción.....	10
Capítulo 1	11
1. 1 Antecedentes	12
1.1.1 Localización.....	12
1.1.2 Objetivos de la institución.....	13
1.1.3 Misión.....	13
1.1.4 Visión	13
1.2 Planteamiento del problema.....	13
1.2.1 Propuesta de la investigación.....	14
1.2.2 Objetivos de la investigación.....	15
1.2.2.1 Objetivo general.....	15
1.2.2.2 Objetivos particulares.....	15
1.2.3 Preguntas de investigación.....	15
1.2.4 Hipótesis.....	15
1.2.5 Limitaciones y supuestos.....	16
1.2.6 Relevancia.....	16
Capítulo 2	17
2.1 Marco conceptual.....	18
2.1.1 Conceptos importantes.....	18
2.1.2 Optimización.....	18
2.1.3 Optimización de aplicaciones.....	19

2.1.4 Optimizar.....	19
2.1.5 Aplicación web.....	20
2.1.6 Aplicación de escritorio.....	20
2.1.7 Aplicación.....	22
2.1.8 Framework.....	23
2.1.9 Librería.....	25
2.1.10 Librerías caseras.....	27
2.1.11 Librerías externas.....	27
2.1.12 JQuery.....	28
2.1.13 Mootools.....	28
Capítulo 3.....	29
3.1 Diseño.....	30
3.2 Desarrollo.....	31
3.2.1 Conociendo CodeIgniter.....	31
3.2.2 Maquetación e implementación del módulo login.....	32
3.2.3 Implementación de Ion_AUTH.....	33
3.2.4 Implementación de librería JWT.....	34
3.2.5 Utilización de svelte js.....	35
Capítulo 4.....	37
4.1 Resultados	38
4.2 Discusión.....	38
Capítulo 5.....	39
5.1 Conclusiones.....	40
5.2 Recomendaciones.....	40

5.3 Bibliografía.....	41
5.4 Glosario.....	42

ÍNDICE DE FIGURAS

1.1 Ubicación de la empresa.....	13
2.1 Base de datos de ejemplo.....	31
2.2 Instalación del framework.....	32
2.3 Eliminación de archivos.....	32
2.4 Librerías usadas en el proyecto.....	33
2.5 Funciones de la librería JWT.....	34
2.6 Reglas para una vista responsiva.....	35
2.7 Vista final del Login.....	36

Resumen

El presente documento ha sido realizado con la finalidad de acreditar la carrera de ingeniería en informática en la Universidad Politécnica de Sinaloa. El siguiente trabajo de investigación se centra en el tema "Optimización en el desarrollo de aplicaciones web o de escritorio mediante el uso de Frameworks" enfocándose en el uso del framework como principal herramienta para desarrollar aplicaciones con el objetivo de reducir el tiempo de desarrollo para aprovechar al máximo los recursos, de igual manera tener una mejor estructura organizacional en el proyecto a desarrollar.

Abstract

This document has been written with the purpose of accrediting the computer science career at Universidad Politécnica de Sinaloa. This research work is focused on "Optimization in the development of web or desktop applications with the use of Frameworks" focusing in the use of framework like main tool to develop applications, with the objective of decreasing the development time to maximize resources, in the same way to have a better organizational structure in the project.

Introducción

Actualmente las nuevas aplicaciones ocupan un desarrollo mas practico y efectivo, por el motivo de ser mas complejas cada vez más, los nuevos cambios que sufren las herramientas de desarrollo hacen que estemos en constante movimiento entorno a estas de tal modo necesitan estar actualizándose para brindar un mejor desempeño. El uso de este tipo de herramientas marca la diferencia entre entregar un producto y entregar una experiencia única al cliente.

En años recientes han salido a la luz nuevas herramientas de desarrollo que benefician tanto al cliente como al desarrollador enfocadas en ofrecer un entorno mas productivo, agilizando optimizando y perfeccionando las actividades relacionadas con el desarrollo de aplicaciones.

CAPÍTULO I

ANTECEDENTES Y PLANTEAMIENTO DEL PROBLEMA.

1.1 Antecedentes

UPC es una firma de servicios profesionales, nos distinguimos por otorgar soluciones en las áreas de defensa Fiscal, Planeación Fiscal y Litigiosa, Tecnologías de Información, Desarrollo e implementación de Procesos Administrativos y Financieros. Nos concentramos en las soluciones que aporten ahorro y beneficio a las empresas, lo que nos permite dar excelentes resultados. Preocupados siempre por la capacitación de todos los que en ella colaboramos, con el fin de superar las expectativas de nuestros clientes. Estamos en constante crecimiento, siempre a la vanguardia de los cambios globales, generando más de 230 fuentes de empleo en 6 ciudades de la República Mexicana y 1 en el extranjero que trabajan para que usted sólo se preocupe por hacer negocios, buenos negocios.

UPC Technologies nace con el fin de diversificar los servicios de la firma UPC y así ofrecer a sus clientes servicios adicionales, enfocados a soluciones en tecnologías de información. La finalidad de este concepto, es que el cliente encuentre la integración de las soluciones basadas en Tecnología, a la medida de su empresa en un solo lugar. Actualmente contamos con la capacidad de brindar estos servicios a sus clientes con experiencia, conocimiento y tecnología, de tal forma que sea más consistente y predecible, garantizando una óptima relación costo/beneficio.

1.1.1 Localización

UPC Technologies se encuentra ubicada por la calle Sierra Rumorosa No. 321, Lomas de Mazatlán, Mazatlán, Sinaloa, México.

Ubicación de la empresa

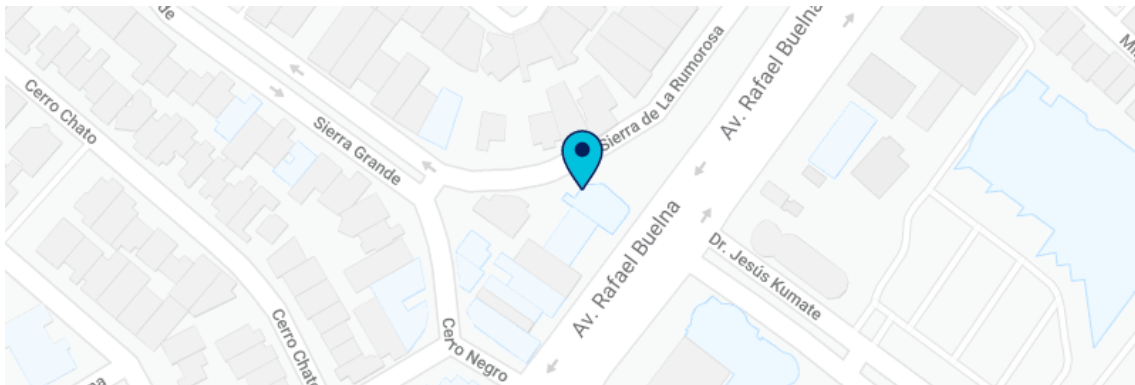


Figura 1.1

Fuente: Google Maps

1.1.2 Objetivos de la institución

El Software para la facturación electrónica es un desarrollo de UPC Technologies con el objetivo de satisfacer las necesidades de las empresas, negocios y/o personas físicas que deseen implementar un control administrativo y financiero, además de poder emitir comprobantes digitales, por medios propios o por PAC.

1.1.3 Misión

Contribuir con nuestro conocimiento y capital humano para proteger, brindar confianza y ofrecer seguridad jurídica en el fortalecimiento del patrimonio de nuestros clientes y sus plataformas de negocios.

1.1.4 Visión

Ser una firma líder, reconocida nacional e internacionalmente, por la diversidad y alta calidad de servicios y desempeño exitoso, asegurando la protección y el crecimiento del capital de nuestros clientes.

1.2 Planteamiento del problema

El desarrollo de aplicaciones web y de escritorio ha avanzado con el paso del tiempo, aunado a esto los grandes proyectos requieren una mayor organización y un mejor control en su estructura de desarrollo, esto debido a que

se tiene que realizar un análisis sobre dónde situar los diferentes archivos de la aplicación (recursos, controladores, vistas, modelos, etc.).

Desarrollar una aplicación web o de escritorio consume mucho tiempo en desarrollar nuevas funciones para cada módulo de la aplicación y de igual manera en este tipo de aplicaciones existe la duplicidad de código, dicho de otra forma, existen ciertos módulos que se repiten constantemente, por ejemplo, la conexión con la base de datos, validación de formularios, páginas de estilos, etc.

El costo y el tiempo en el desarrollo de una aplicación son dos de los factores más importantes tanto para el cliente como para los desarrolladores, terminar antes un proyecto implica que el tiempo sea menor y por lo tanto el coste de la aplicación también disminuye sin dejar de lado la calidad. Otro de los mayores problemas es la combinación de las diferentes capas con las que cuenta una ampliación, estas capas son: la capa de datos, la capa de la lógica del negocio y la capa de la interfaz con el usuario. De igual manera existe otro problema el cual surge cuando se trabaja en equipo ya que resulta complejo y un poco difícil leer el código desarrollado por otra persona, esto debido a que no se tiene la misma forma de trabajar.

Por último, al no contar con una estructura sólida en el desarrollo de la aplicación, los diferentes desarrolladores tendrán problemas en el momento que haya que actualizar la aplicación o realizar algún cambio lo cual conlleva a un mayor costo tanto de tiempo como de dinero en el mantenimiento de la aplicación.

1.2.1 Propuesta de investigación

Para poder resolver la problemática anteriormente expuesta, se han creado un conjunto de archivos y pautas estandarizadas que definen una estructura y metodología acerca de cómo hacer el desarrollo de aplicaciones, esto sirve como una guía que facilita la forma de programar llamada framework.

Su objetivo es el desarrollo ágil de aplicaciones mediante la implementación de librerías, también nos brinda módulos o artefactos que nos dan una rápida solución para el desarrollo, ahorrando tiempo, manteniendo organizado el código y permitiendo una fácil actualización.

Existe un framework llamado CodeIgniter el cual es muy fácil de utilizar y puede resolver la mayoría de los problemas que se presentan en la problemática. Este framework implementa un patrón de arquitectura de software basado en MVC, que separa la lógica de control, la interfaz del usuario y los datos del sistema.

1.2.2 Objetivos de la investigación

A partir del planteamiento del problema y de la propuesta de investigación, se han definido los siguientes objetivos de la investigación.

1.2.2.1 Objetivo general

El objetivo general es implementar el uso del framework CodeIgniter para desarrollar aplicaciones tanto web como de escritorio identificando de esta manera que es la mejor opción para brindar los mayores beneficios.

1.2.2.2 Objetivos particulares

- I. Implementar frameworks de calidad en el desarrollo de aplicaciones web y de escritorio.
- II. Dar a conocer cuáles son los beneficios de utilizar un framework.
- III. Demostrar las ventajas que tiene CodeIgniter sobre otros frameworks.
- IV. Mejorar las expectativas de los desarrolladores al momento de programar en un proyecto mediano, gastando la menor cantidad de recursos.

1.2.3 Preguntas de investigación

¿Cuáles son los beneficios que usar CodeIgniter?

¿Por qué es importante emplear un framework en el desarrollo de aplicaciones?

¿Realmente es necesaria la implementación de un framework?

¿Qué importancia tiene al momento elegir un framework?

1.2.4 Hipótesis

Se reducirán considerablemente los tiempos de desarrollo de aplicaciones web y de escritorio con la implementación del framework lo cual significa un aspecto importante dentro de la empresa.

1.2.5 Limitaciones y supuestos

Para las limitaciones y supuestos de esta investigación se tomaron en cuenta varios factores referentes a las nuevas tecnologías que avanzan mas día con día.

Con estos factores en cuenta en el supuesto de que el framework CodeIgniter se siga actualizando, contemplando las nuevas tecnologías de desarrollo y que la comunidad siga contribuyendo, se cree que tenga muchos años más de vida útil dentro del mercado.

1.2.6 Relevancia

La relevancia de esta investigación se centra en los beneficios y la optimización que se obtendrá atreves de la implementación de los frameworks, también se hará más fácil identificar cuáles son los factores más importantes al momento de desarrollar una aplicación web o de escritorio y cuál sería la manera más eficiente y eficaz para poder desarrollarla.

CAPÍTULO 2

Marco conceptual

2.1 Marco conceptual

2.1.1 Conceptos importantes

2.1.2 Optimización

Optimización es la acción y efecto de optimizar. Este verbo hace referencia a buscar la mejor manera de realizar una actividad. El término se utiliza mucho en el ámbito de la informática. La optimización de software busca adaptar los programas informáticos para que realicen sus tareas de la forma más eficiente posible. Virtualmente, existen infinitas maneras de desarrollar una misma aplicación, y uno de los factores más influyentes a la hora de crear el diseño es la arquitectura de hardware con la cual se desea trabajar. En pocas palabras, conseguir el mejor rendimiento en una plataforma enfocada en el tipo y la cantidad memoria es muy diferente a hacerlo en una cuyo fuerte es la velocidad de los procesadores. A nivel general, la optimización puede realizarse en diversos ámbitos, pero siempre con el mismo objetivo: mejorar el funcionamiento de algo o el desarrollo de un proyecto a través de una gestión perfeccionada de los recursos. La optimización puede realizarse en distintos niveles, aunque lo recomendable es concretarla hacia el final de un proceso.

Optimización es un concepto que casi todas las personas aprendemos de manera natural desde la infancia, aunque no conozcamos el término hasta alcanzada una cierta edad. Los colegios primarios suelen incluir en sus programas la realización de trabajos prácticos en grupo, y es a través de esta actividad en particular que tenemos uno de nuestros primeros acercamientos a la búsqueda de la mejor organización posible: intentamos dividir las tareas, de manera tal que cada integrante se haga cargo de aquello que mejor sabe hacer [1].

En general, la optimización es empleada para que una tarea se realice más rápidamente. Pero este no siempre es el motivo; por ejemplo, en determinados casos lo más importante es que se consuma menos memoria, por lo tanto, se deben crear programas más lentos, pero que estén optimizados con respecto al consumo de la memoria.

La optimización se hace siempre con respecto a uno o más recursos como ser: tiempo de ejecución, uso de memoria, espacio en disco, ancho de banda,

consumo de energía, etc. Muchas veces la optimización de un recurso se hace a expensas de otros recursos [2].

2.1.3 Optimización de aplicaciones

Optimizar una aplicación significa hacer los cambios pertinentes para que esta se ejecute y funcione más rápidamente, o para que ocupe menos memoria, o para que gaste menos batería (útil en computadoras portátiles). Por lo general la optimización de un programa se hace a través de otros programas, una mejor configuración, etc., pero siempre a nivel software.

De todas maneras, la mejor optimización que puede hacerse a una aplicación es a nivel código fuente, es decir, cambios en las sentencias de programación por parte de su desarrollador. Mejorar los algoritmos resulta en una mejora del rendimiento general de la aplicación. También puede emplearse un compilador optimizador que ayude a crear una aplicación más optimizada.

Muchas veces, la optimización de los algoritmos resulta en códigos menos legibles para los programadores (que, al fin y al cabo, son humanos) y más difíciles de mantener y expandir. Pero mejoran notablemente la velocidad y memoria cuando la computadora los ejecuta [2].

2.1.4 Optimizar

Optimizar es un verbo que se puede entender como la acción para hacer algo grande, es la misma que mejorar o mejorar. Quien busca optimizar, busca transformar para mejor, busca algo ideal, busca extraer el máximo.

Así, optimizar es el acto de causar mejoras, elevando las circunstancias para que cierta actividad se realice más rápida y eficientemente; optimizar es hacer que el rendimiento sea más alto. Por ejemplo: “Desmontaré la máquina, limpiaré y probaré todas las piezas. Las piezas problemáticas se sustituirán por otras nuevas. Así que estoy seguro de que voy a optimizar mucho su rendimiento.

Encontraremos el término optimizar en muchas áreas siempre como la idea de mejorar, ir más allá cualitativamente, especialmente en el área de negocios y

administración de empresas, como en la optimización de costos, optimizar tiempos, optimizar productividad, optimizar gastos, optimizar procesos, optimizar distribución y así sucesivamente.

En el campo de la estadística, el término optimización aparece como la acción de buscar el valor más alto de una magnitud dada. En matemáticas, podemos entender cómo optimizar la búsqueda de una función para maximizar o minimizar su valor [3].

2.1.5 Aplicación web

Aplicación web. En la Ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un Servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación (Software) que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. Las aplicaciones web son populares debido a lo práctico del navegador web como Cliente ligero, a la independencia del Sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales.

Aunque existen muchas variaciones posibles, una aplicación web está normalmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web ofrece la primera capa y un motor capaz de usar alguna tecnología web, constituye la capa de en medio. Por último, una base de datos constituye la tercera y última capa. El navegador web manda peticiones a la capa de enmedio que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario [4].

Las aplicaciones Web son aquellas en que los usuarios acceden a ellas en un servidor Web a través de Internet o de una intranet. Las aplicaciones Web se han popularizado tanto que muy pronto las aplicaciones cliente-servidor serán historia. La popularidad de las Aplicaciones Web se debe a que basta con tener solamente una PC, un navegador Web y acceso a la Internet. La facilidad de uso, la habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software es otra razón de su enorme popularidad.

Las aplicaciones Web, por ejemplo, un programa de contabilidad, generan en forma dinámica páginas en formato HTML o XHTML. Generalmente cada página Web individual es vista por el cliente como un formulario o documento estático, pero la secuencia de páginas provee de una experiencia interactiva [5].

Será un servidor el encargado de realizar la funcionalidad del sistema que hemos implementado a través de un programa que manejará el usuario con el navegador web (Internet Explorer, Firefox, Chrome, etc.) de su ordenador.

La principal ventaja será la disponibilidad de la aplicación a través de dispositivos que tengan un navegador web: ordenadores, teléfonos móviles, tablets, etc. De esta forma un escenario posible podría ser un comercial de una empresa que cierra un pedido en el domicilio de su cliente y a través de una tablet deja realizado el mismo y confirmado con el cliente un plazo de entrega. En ese caso el equipo que tramite los pedidos ubicado en la empresa tendrá constancia del pedido en el momento y podrá tramitarlo rápidamente [6].

2.1.6 Aplicación de escritorio

Será un programa el encargado de realizar la funcionalidad del software implementado que instalaremos en cada puesto de trabajo y se conectará a través de Internet con la base de datos. La principal ventaja de este sistema será la rapidez de uso ya que podremos incorporar todos los controles de escritorio y todos los eventos asociados a ellos. Como principal desventaja tendremos la gestión de actualizaciones que nos obligará a actualizar todos los programas instalados en cada puesto de la empresa cuando implementemos evoluciones o corrijamos fallos. Esto nos obligará a diseñar un sistema automático de gestión de actualizaciones ya que un usuario con un software obsoleto puede dañar la base de datos. Otra desventaja importante es la escasa portabilidad ya que, si lo implementamos para un entorno Windows, solo en equipos de ese tipo funcionará y no podremos usarla en una Tablet o un teléfono [6].

Una aplicación Desktop (también llamada de Escritorio) es aquella que está instalada en el ordenador del Usuario, que es ejecutada directamente por el sistema operativo, ya sea Microsoft Windows, Mac OS X, Linux o Solaris, y cuyo rendimiento depende de diversas configuraciones de hardware como memoria RAM, disco duro, memoria de video, etc, algunos ejemplos son: Programas de

Microsoft Office, Cliente FTP, iTunes, Adobe Photoshop. Habitualmente su ejecución no requiere comunicación con el exterior, sino que se realiza de forma local. Esto repercute en mayor velocidad de procesamiento, y por tanto en mayores capacidades a la hora de programar herramientas más complicadas o funcionales [7].

2.1.7 Aplicación

Aplicación informática. Es un programa informático hecho para permitir a un usuario realizar uno o varios tipos de trabajo. Suele resultar una solución informática para la automatización de ciertas tareas complicadas como pueden ser la contabilidad, la redacción de documentos, o la gestión de un almacén. Algunos ejemplos de programas de aplicación son los procesadores de textos, hojas de cálculo, y base de datos. Tales aplicaciones desarrolladas «a medida» suelen ofrecer una gran potencia ya que están exclusivamente diseñadas para resolver un problema específico. Otros, llamados paquetes integrados de software, ofrecen menos potencia, pero a cambio incluyen varias aplicaciones, como un programa procesador de textos, de hoja de cálculo y de base de datos [8].

Las aplicaciones informáticas, en informática es un tipo de proyecto informático creado como herramienta para proporcionar a un usuario realizar uno o varios tipos de trabajos. Esto lo distingue primordialmente de otros tipos de programas, los cuales los sistemas operativos (que hacen trabajar la PC), los utilitarios (que ejecutan las tareas de mantenimientos o del uso frecuente), y los lenguajes de programación (para inventar programas informáticos). Esta suele derivar una solución informática para la automatización de tareas específicas, pero algo dificultosas, las cuales pueden ser la ejecución y redacción de documentos, la contabilidad en cierta parte o la tramitación de un almacén. Unos ejemplos de programas de aplicación son las bases de datos, los procesadores de textos y algunas hojas de cálculo.

Algunas aplicaciones perfeccionadas a providencia suelen prometer una potencia superior ya que están exclusivamente diseñadas para solucionar un inconveniente específico. Otros, denominados paquetes integrados de software, brindan menos potencia, pero a cambio contienen diversas aplicaciones, como

un programa procesador de textos, de hoja de cálculo y base de datos. En la actualidad, con el automatismo de dispositivos móviles se ha desarrollado el término app, aplicación informática para dispositivos móviles o tabletas con aglomeración de funcionalidades. Desde juegos hasta aplicaciones para realizar tareas habituales. Es un aventador gigante que pueden hacer mucho más divertido e interactivo a los dispositivos móviles [9].

Programa informático que permite a un usuario utilizar una computadora con un fin específico. Las aplicaciones son parte del software de una computadora, y suelen ejecutarse sobre el sistema operativo. Una aplicación de software suele tener un único objetivo: navegar en la web, revisar correo, explorar el disco duro, editar textos, jugar (un juego es un tipo de aplicación), etc. Una aplicación que posee múltiples programas se considera un paquete. Son ejemplos de aplicaciones Internet Explorer, Outlook, Word, Excel, WinAmp, etc.

Características de las aplicaciones

En general, una aplicación es un programa compilado (aunque a veces interpretado), escrito en cualquier lenguaje de programación. Las aplicaciones pueden tener distintas licencias de distribución como ser freeware, shareware, trialware, etc. Para más información ver: Licencias de software. Las aplicaciones tienen algún tipo de interfaz, que puede ser una interfaz de texto o una interfaz gráfica (o ambas). También hay que destacar que la distinción entre aplicaciones y sistemas operativos muchas veces no es clara. De hecho, en algunos sistemas integrados no existe una clara distinción para el usuario entre el sistema y sus aplicaciones [10].

2.1.8 Framework

En el desarrollo de Software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, librerías y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades

del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

Dentro de este aspecto, podemos basarnos en el modelo MVC (Controlador => Modelo => Vista) ya que debemos fragmentar nuestra programación. Tenemos que contemplar estos aspectos básicos en cuanto a la implementación de nuestro sistema:

Controlador: Con este apartado podemos controlar el acceso (incluso todo) a nuestra aplicación, esto pueden ser: archivos, scripts o programas; cualquier tipo de información que permita la interfaz. Así, podremos diversificar nuestro contenido de forma dinámica, y estática (a la vez); pues, sólo debemos controlar ciertos aspectos (como se ha mencionado antes).

Modelo: Este miembro del controlador maneja las operaciones lógicas, y de manejo de información (previamente enviada por su ancestro) para resultar de una forma explicable, y sin titubeos. Cada miembro debe ser meticulosamente llamado, en su correcto nombre y en principio, con su verdadera naturaleza: el manejo de información, su complementación directa.

Vista: Al final, a este miembro de la familia le corresponde dibujar, o expresar la última forma de los datos: la interfaz gráfica que interactúa con el usuario final del programa (GUI). Después de todo, a este miembro le toca evidenciar la información obtenida hasta hacerla llegar con el controlador. Solo (e inicialmente), nos espera demostrar la información [11].

Un Framework, que se podría traducir aproximadamente como marco de trabajo, es el esquema o estructura que se establece y que se aprovecha para desarrollar y organizar un software determinado. Esta definición, algo compleja, podría resumirse como el entorno pensado para hacer más sencilla la programación de cualquier aplicación o herramienta actual.

Este sistema plantea varias ventajas para los programadores, ya que automatiza muchos procesos y además facilita el conjunto de la programación. Es útil, por ejemplo, para evitar el tener que repetir código para realizar funciones habituales en un rango de herramientas, como puede ser el acceder a bases de datos o

realizar llamadas a Internet. Todas estas tareas son las que se realizan de forma mucho más fácil cuando se trabaja dentro de un framework. Plantea muchas ventajas y, además, es capaz de hacer que se realicen incluso labores mucho más complejas que, por otros medios, serían imposibles de plantear siquiera a la hora de programar algo. No obstante, su utilidad es algo que depende del tipo de programa y de contexto en el que vaya a emplearse.

Un Framework sirve para poder escribir código o desarrollar una aplicación de manera más sencilla. Es algo que permite una mejor organización y control de todo el código elaborado, así como una posible reutilización en el futuro. Debido a esto, garantiza una mayor productividad que los métodos más convencionales y una minimización del coste al agilizar las horas de trabajo volcadas en el desarrollo [12].

2.1.9 Librería

Las librerías son un grupo de archivos que tienen una funcionalidad pre-construida por terceros, y que puede ser usadas por cualquier ejecutable. Las librerías contienen en su interior variables y funciones, se conoce como librerías (o bibliotecas) a cierto tipo de archivos que podemos importar o incluir en nuestro programa. Estos archivos contienen las especificaciones de diferentes funcionalidades ya construidas y utilizables, como por ejemplo leer del teclado o mostrar algo por pantalla entre muchas otras más. al poder incluir estas librerías con definiciones de diferentes funcionalidades podremos ahorrarnos gran cantidad de cosas [13].

Una librería no es más que una serie de código de programación, un conjunto de código que alguien ha hecho para que tú puedes reutilizar dentro de tus proyectos. El objetivo es sencillo, hacer más fácil y rápido el desarrollo de ciertas funciones dentro de tu app. Normalmente las librerías están enfocadas a solucionar problemas concretos, es decir no te van a hacer un proyecto entero, pero sí que te van a ayudar con ciertas partes de ese proyecto; vamos a ver un ejemplo que siempre queda más claro y se ve de una forma más nítida. Imaginemos que tenemos una aplicación de una tienda online y lo que queremos

hacer es mostrar todos los productos, la información, imágenes que hay en la web pero en nuestra aplicación móvil.

El primer beneficio que obtenemos es agilizar el desarrollo, es decir, si tenemos que hacer algo, si tenemos que llevar a cabo una tarea y alguien ya la ha desarrollado por nosotros, y podemos utilizar ese código que ya está escrito, vamos a cortar mucho el tiempo de desarrollo. No sólo de tiempo, sino también de coste, en vez de hacer muchas cosas completamente desde cero lo que hacemos es no reinventar la “rueda” que se llama en programación y utilizar la librería que ya lo hace por nosotros.

El segundo beneficio es hacer cosas que serían imposibles o muy difíciles de hacer, si las tuviéramos que hacer desde cero, lo que otorga de cierta viabilidad a algunos proyectos que quizás sin estas librerías serían inviables. El tercer beneficio cuando utilizamos librerías, cuando utilizamos API's es que el código que obtenemos es un código probado, testado y mucho más viable que si lo desarrollamos nosotros. Una librería normalmente está desarrollada por mucha gente y lo que es más importante, una librería la utiliza mucha gente.

Lo anterior quiere decir que si hay un error en ese código, al ser utilizada por mucha gente, los errores se van a descubrir antes y por tanto también la corrección de estos errores, va a ser más rápida, más eficaz, con lo cual por norma general y si elegimos bien una librería, el código que vamos obtener es mucho más fiable, que si los desarrolláramos nosotros mismos [14].

En programación, una librería es un archivo o conjunto de archivos que se utilizan para facilitar la programación. Las librerías, también llamadas "frameworks", consisten en archivos de código a los que llamamos al principio de la página, por ejemplo, una librería JavaScript será un archivo en JavaScript que insertamos al principio de la página. Podemos insertar cualquier archivo al principio de la página, sin embargo, no todos los archivos pueden considerarse librerías. Para que un archivo sea una librería, este puede tener que usarse en distintas páginas, no sólo del mismo sitio, sino de varios sitios. Además, debe modificar el código de tal manera que nos facilite la inserción de nuevo código en la página.

Un ejemplo de librería sería la inclusión de un archivo `reset.css` que nos resetea la página para que ésta se vea igual en todos los navegadores. El archivo vale para páginas de distintos sitios, y nos asegura que el estilo de la página se verá igual en todos los navegadores.

Librerías caseras y librerías externas.

2.1.10 Librerías caseras

Si entendemos el concepto de librería, veremos que no es difícil, sabiendo un poco de programación, crear nosotros mismos nuestras pequeñas librerías, que nos sirven para diferentes tareas concretas. Por ejemplo, si has seguido el manual de Ajax en la sección "Nuevas Tendencias" verás que hemos creado un "Objeto Ajax" que nos facilita el trabajo con Ajax. Este objeto lo hemos puesto dentro de un archivo para poder utilizarlo en distintas páginas que enlacen con este archivo. Esto puede considerarse una pequeña librería casera o propia. Las librerías caseras son, por lo tanto, archivos que fabrica el propio programador para facilitar su propio trabajo, y que puede incluir en cualquier página cuando lo necesite.

2.1.11 Librerías externas

El concepto de librería externa es el mismo pero ampliado. Una librería externa es un archivo (o a veces varios) que está fabricado por expertos programadores, y que no sólo vale para un caso concreto, sino que podemos utilizar de forma general. La utilización de nuevos objetos, métodos, y propiedades puede variar la forma de programar el resto del código, ya que éstos afectan a la mayoría de las acciones que pueden realizarse en la programación.

Debemos, por tanto, aprender a programar utilizando estas librerías. En algunas de ellas es casi como aprender otro código de programación. Sin embargo, una vez aprendido vemos cómo el trabajo se hace mucho más fácil, y además podemos realizar acciones que de otro modo nos resultarían muy complicadas utilizando sólo el código de programación. Veremos en esta sección las librerías más utilizadas con javascript. Estas son jQuery y MooTools. Cada una de ellas tiene sus características y su manera de trabajar en la página.

2.1.12 JQuery

Esta es una librería JavaScript que pretende facilitar al máximo el trabajo con JavaScript. Para ello se basa en el objeto jQuery que accede de forma rápida y simple a los elementos de la página y le aplica una serie de métodos y propiedades que simplifican el trabajo con JavaScript.

2.1.13 Mootools

Esta es una librería JavaScript, que, aunque es un poco más complicada de utilizar que JQuery, es bastante más completa. Permite realizar acciones que muchas veces o nos sería muy complicado de hacer o no sabríamos como hacerlas. Y todo ello utilizando objetos y métodos ya definidos. Mootools es particularmente útil por la cantidad de efectos que podemos hacer en la página de una manera bastante sencilla. El manual consta de tres partes. En la primer aprendemos a utilizar mootools para hacer las mismas tareas que con JavaScript, pero de una manera más simple [15].

CAPÍTULO III

DISEÑO Y DESARROLLO

3.1 Diseño

Para poder desarrollar y demostrar que los objetivos planteados previamente se pudieron resolver, primero se tuvieron que investigar los diferentes tipos de frameworks que estaban a nuestro alcance, para poder seleccionar un framework de acuerdo a las necesidades y los recursos con los cuales se contaban en el momento de la planeación del proyecto que se tomó como ejemplo para la comparación.

Una vez teniendo la idea general acerca del proyecto, se repartieron los recursos disponibles, se delimitaron los tiempos de desarrollo en base a proyectos similares que fueron desarrollados sin el uso de frameworks para poder tomar un punto de referencia y sacar a flote las ventajas que conlleva el uso de los frameworks.

Las principales características que se tomaron en cuenta para seleccionar los frameworks a comparar fueron compatibilidad, la cantidad de librerías con las que cuenta el framework, fácil enrutamiento, flexibilidad, facilidad de instalación, soporte para MVC, documentación legible entre otras.

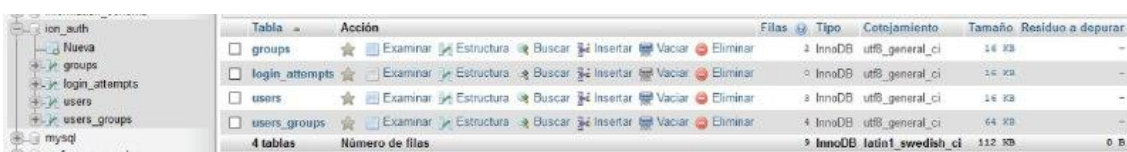
En esta ocasión se fueron descartando frameworks que no cumplían con los requerimientos mínimos de nuestro proyecto base quedando el framework CodeIgniter como una de las opciones más viables para el desarrollo de nuestro proyecto, otras de las opciones que se contempló fue Laravel debido a que algunos expertos consideran que debido a las nuevas versiones de este framework ya no se considera uno como tal, puesto que se requieren muchas más capas para poder organizar todos los aspectos característicos del proyecto o aplicación.

El desarrollo de este proyecto constó de 3 partes fundamentales, la primera fue la investigación y documentación acerca del framework y de las demás

herramientas necesarias. La segunda parte fue el desarrollo del back end dentro con el uso del framework CodeIgniter, el cual fue de gran ayuda debido a su amplia gama de librerías y soporte por parte de otros usuarios. Y la tercer parte fue el desarrollo del front-end con ayuda de otro framework llamado SvelteJS, el cual se utilizó únicamente para realizar el diseño de dicho modulo.

Además, unas de las librerías que se usaron, traía consigo una pequeña base de datos para pruebas (ver figura 2.1), la cual se tomó de ejemplo y de base para probar el correcto funcionamiento del proyecto.

Base de datos de ejemplo



The image shows a screenshot of a database management interface. On the left, there is a tree view showing a database named 'ion_auth' with several tables: 'groups', 'login_attempts', 'users', and 'users_groups'. The main area displays the structure of the 'groups' table. The table has 2 columns: 'id' and 'name'. The 'id' column is of type 'InnoDB' with a 'utf8_general_ci' collation and a size of 16 KB. The 'name' column is also of type 'InnoDB' with a 'utf8_general_ci' collation and a size of 16 KB. The table is currently empty, as indicated by the 'Número de filas' (Number of rows) being 0.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
groups	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	16 KB	-
login_attempts	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	16 KB	-
users	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	16 KB	-
users_groups	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	64 KB	-
4 tablas	Número de filas	0	InnoDB	latin1_swedish_ci	112 KB	0 B

Figura 2.1

Fuente: Propia del Autor

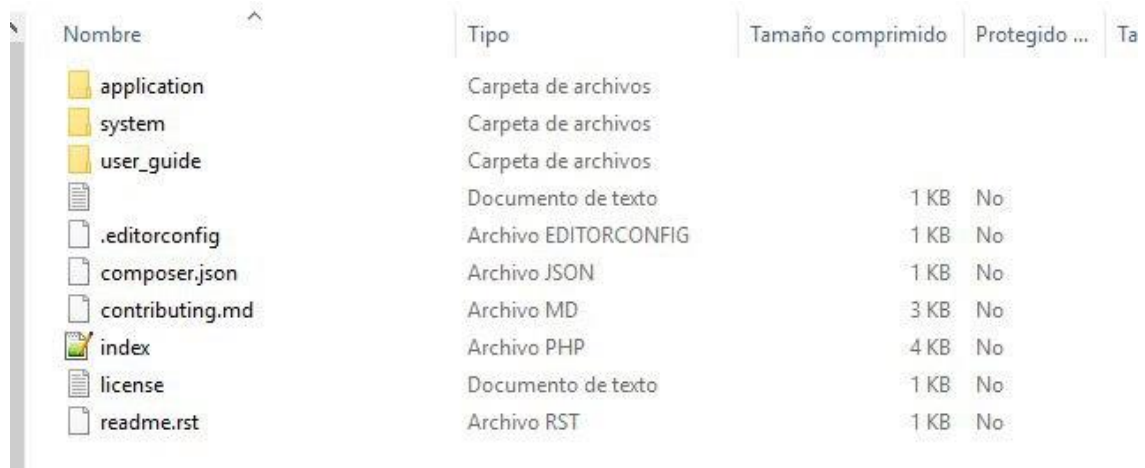
3.2 Desarrollo

Dentro del área de desarrollo en el departamento de technologies nos dimos la tarea de planificar un sistema de notificación el cual por los tiempos de limitados y los recursos disponibles solo se desarrolló el módulo de login. Pero con este módulo fue más que suficiente para comprobar los puntos anteriormente planteados obteniendo mejores resultados a los esperados.

3.2.1 Conociendo CodeIgniter

Después de haber leído el manual de usuario del framework y de haber entendido la sintaxis que este framework maneja con el propósito de aprovechar al máximo todas las herramientas y funcionalidades disponibles se empezó a desarrollar la estructura básica del módulo de login. Una vez hecho esto se procedió a descargar e instalar el framework (ver imagen 2.2).

Instalación del framework.



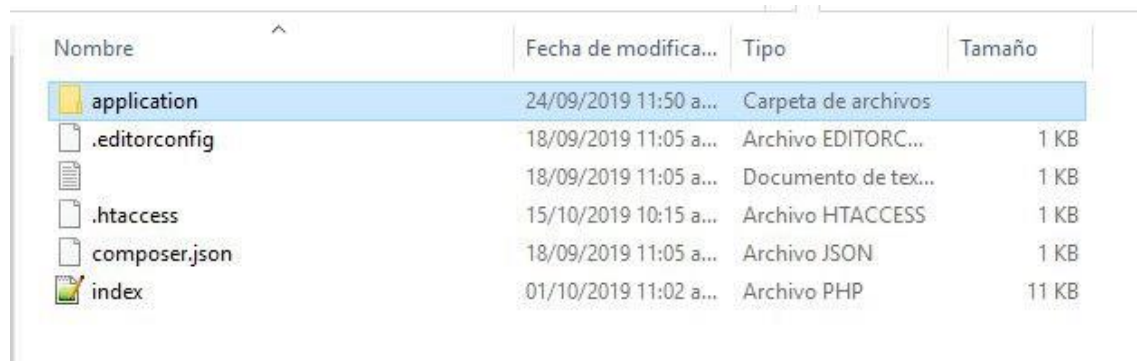
Nombre	Tipo	Tamaño comprimido	Protegido ...	Ta
application	Carpeta de archivos			
system	Carpeta de archivos			
user_guide	Carpeta de archivos			
	Documento de texto	1 KB	No	
.editorconfig	Archivo EDITORCONFIG	1 KB	No	
composer.json	Archivo JSON	1 KB	No	
contributing.md	Archivo MD	3 KB	No	
index	Archivo PHP	4 KB	No	
license	Documento de texto	1 KB	No	
readme.rst	Archivo RST	1 KB	No	

Figura 2.2

Fuente: Propia del Autor

Una vez se descargó e instaló el framework, se eliminaron los archivos que no se necesitaban como por ejemplo la guía de usuario y los archivos de prueba, dejando únicamente los archivos esenciales para el desarrollo de nuestro proyecto (ver figura 2.3).

Eliminación de archivos



Nombre	Fecha de modifica...	Tipo	Tamaño
application	24/09/2019 11:50 a...	Carpeta de archivos	
.editorconfig	18/09/2019 11:05 a...	Archivo EDITORC...	1 KB
	18/09/2019 11:05 a...	Documento de tex...	1 KB
.htaccess	15/10/2019 10:15 a...	Archivo HTACCESS	1 KB
composer.json	18/09/2019 11:05 a...	Archivo JSON	1 KB
index	01/10/2019 11:02 a...	Archivo PHP	11 KB

Figura 2.3

Fuente: Propia del Autor

3.2.2 Maquetación e implementación del módulo de login

El formulario consta únicamente de dos campos para la obtención de datos y un botón el cual se encargará de enviar la información correspondiente, de igual manera se integraron dos opciones más al final, una de las cuales guardar las credenciales del usuario para acelerar el proceso de iniciar sesión, el otro campo

es la opción de recuperar la contraseña en caso de que el usuario la haya olvidado.

3.2.3 Implementación de Ion_Auth

La investigación acerca de algunas librerías ayudó a agilizar el proceso de iniciar sesión y de la misma forma ayudaron en el proceso de autenticación, para esto se usó una librería llamada Ion_Auth la cual es ideal para este tipo de módulos, esta librería cuenta con su propio método de encriptar contraseñas la cual aumenta la seguridad.

Por otro lado, esta librería es muy versátil de modo que se puede adaptar a casi cualquier pequeño y mediano proyecto también su manejo e implementación es muy amigable con los usuarios.

3.2.4 Implementación de librería JWT

Para aumentar la seguridad en nuestro proyecto se optó por usar una librería la cual nos ayudó a generar tokens basados en un estándar json los cuales permiten la propagación de identidad. El cliente entonces utilizó el token para probar que estaba actuando como un administrador dentro del sistema. El token está firmado por la clave del servidor, así que el cliente y el servidor fueron capaces de verificar que el token es legítimo.

Librerías usadas en el proyecto

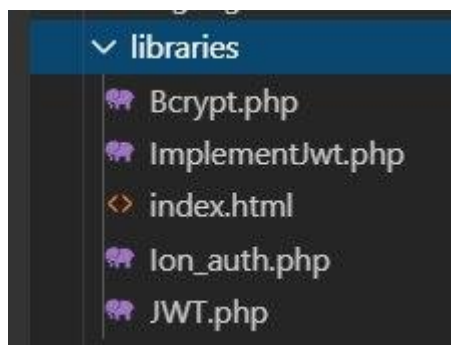
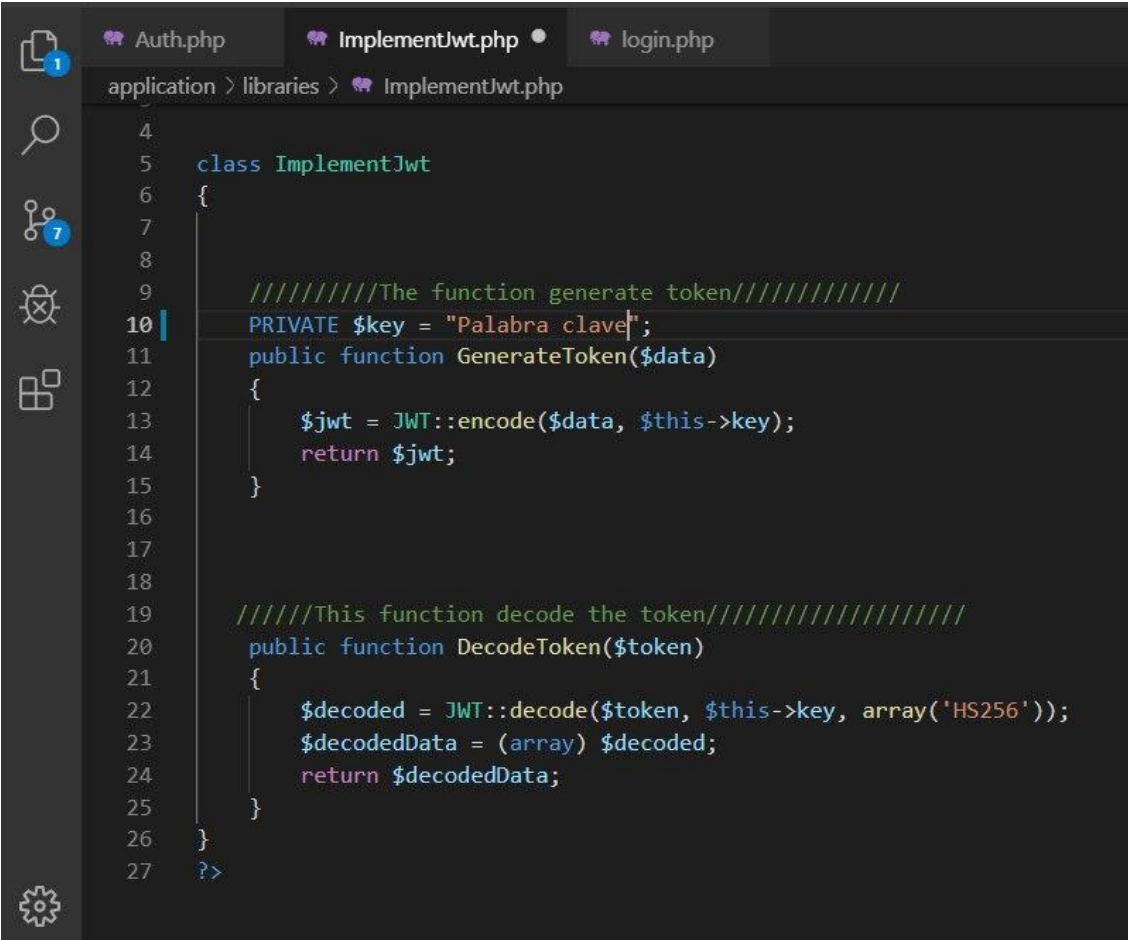


Figura 2.4

Fuente: Propia del Autor

Esta librería cuenta con dos funciones (ver figura 2.5) la primera consiste en codificar los datos con una palabra clave la cual nosotros le asignamos de igual manera diferentes tipos de codificación en base 64. La segunda función ayuda a decodificar los datos utilizando la misma palabra clave en caso de que no sea la misma palabra clave se toma como inválido el token, y se deniega el acceso a la persona. Esta librería nos ayudó mucho en el desarrollo porque es fácil de usar, pero a la vez es difícil de quebrantar esta medida de seguridad por parte de usuarios ajenos.

Funciones de la librería JWT



```
4
5 class ImplementJwt
6 {
7
8
9     //////////////The function generate token////////////////////
10    PRIVATE $key = "Palabra clave";
11    public function GenerateToken($data)
12    {
13        $jwt = JWT::encode($data, $this->key);
14        return $jwt;
15    }
16
17
18
19    ///////This function decode the token////////////////////////////////////
20    public function DecodeToken($token)
21    {
22        $decoded = JWT::decode($token, $this->key, array('HS256'));
23        $decodedData = (array) $decoded;
24        return $decodedData;
25    }
26 }
27 ?>
```

Figura 2.5

Fuente: Propia del Autor

3.2.5 Utilización de svelte js

Para complementar este proyecto en la parte del front end se optó por usar svelte js, debido a que es muy fácil de usar y la forma de programar es muy organizada y limpia estructurando todo en tres bloques, el primer bloque se colocaron todos los elementos de la vista, en el segundo bloque se colocaron las funciones que utilizaron en la vista, y en el último bloque se pusieron los estilos que se utilizaron, de igual manera en este último bloque se colocaron diferentes reglas de acuerdo de la resolución de la pantalla para que el estilo sea responsivo y dinámico (ver figura 2.6), y a la vez sea una interfaz agradable para el usuario dando como resultado un producto de calidad (ver figura 2.7).

Reglas para una vista responsiva

```
> @media (min-width:1200px){ ...  
  }  
> @media (min-width:992px) and (max-width:1199px){ ...  
  }  
> @media (min-width:605px) and (max-width:991px){ ...  
  }  
> @media (max-width:604px){ ...  
  }
```

Figura 2.6

Fuente: Propia del Autor

Vista final del Login

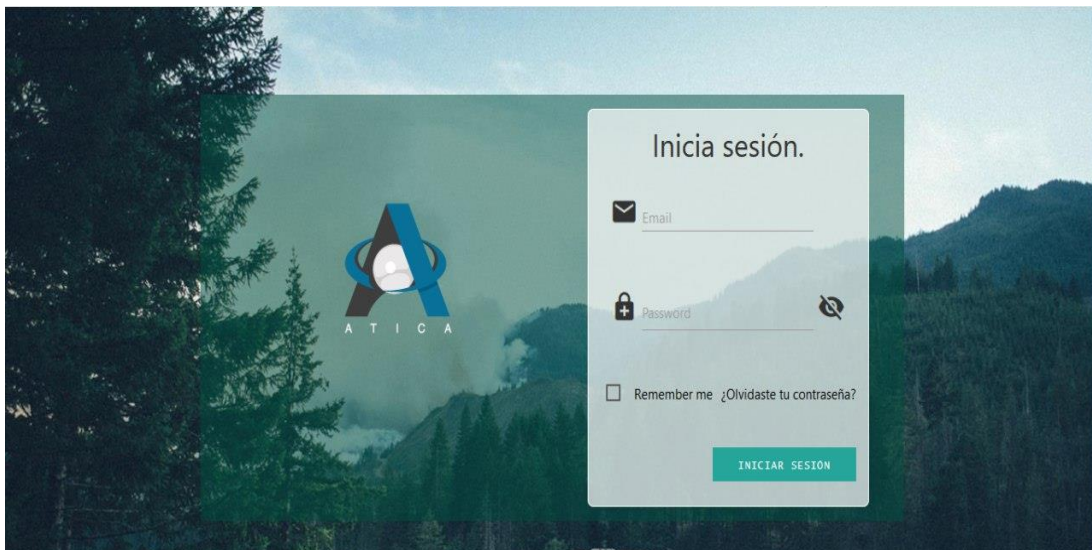


Figura 2.7

Fuente: Propia del Autor

CAPÍTULO 4

RESULTADOS Y DISCUSIONES

4.1 Resultados

Después de mes y medio desarrollando el módulo de login con ayuda del framework los resultados que se obtuvieron fueron satisfactorios, haciéndose notar una enorme diferencia entre el desarrollo de módulos similares de una aplicación web sin ayuda de ningún framework y el desarrollo de este módulo usando un framework.

Con la finalización de este módulo se pretende dar paso a la realización de más módulos implementando la ayuda de frameworks para tener una aplicación web estable, confiable y de calidad que satisfaga todas las necesidades del cliente o la empresa en cuestión.

El tiempo que sobro al finalizar el desarrollo de este módulo se utilizó para corregir errores o hacer pequeñas correcciones que afectaban la estética o el funcionamiento del mismo, si este módulo se hubiera desarrollado sin ayuda de algún framework no se hubieran podido hacer este tipo correcciones debido a la falta de tiempo, o de lo contrario la finalización de este módulo costaría más tiempo de lo esperado.

4.2 Discusión

Otro punto a tener en cuenta es la organización de los componentes y demás archivos que integran este tipo de desarrollo, al no tener una estructura sólida es más difícil establecer un control sobre lo que estamos haciendo. Esto se ve reflejado en el rendimiento y eficiencia de nuestra aplicación web.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

A la hora de empezar con el desarrollo de un proyecto siempre es buena idea buscar las herramientas que podamos usar para ahorrar la mayor cantidad de recursos durante el desarrollo de nuestro proyecto. Cada día este tipo de herramientas van creciendo y evolucionando para adaptarse lo mejor posible a nuestras necesidades y al mismo tiempo a las necesidades del cliente. Cada una de estas herramientas está pensada para que pueda ser implementada con otras que complementen el desarrollo de nuestro proyecto, lamentablemente no todas son compatibles entre sí, pero aun así existe una amplia gama de opciones y combinaciones que brindan un excelente resultado.

Cada framework ofrece diferentes métodos, pero al final todas te ayudaran a realizar tu proyecto de una manera rápida, eficaz y eficiente. No es necesario tener grandes conocimientos en programación para poder utilizar un framework, solo basta con tener nociones básicas programación para poder entender la sintaxis y el suficiente tiempo para leer la documentación necesaria para sacar el máximo potencial y aprovechar al 100% este tipo de herramientas.

5.2 Recomendaciones

Rediseñar y optimizar un proyecto mediano ya existente en base al uso de frameworks, para corroborar que realmente son una excelente herramienta mucho más practica y eficiente, mostrando diferencias significativas que apoyen el objetivo del trabajo.

Se recomienda tener más factores en cuenta al momento de elegir el o los frameworks que se utilizaran con el propósito de aprovechar al máximo los componentes que se pueden crear.

Organizar los tiempos y utilizar una mejor metodología que permita realizar una planeación en base a las metas u objetivos pronosticados a corto y largo plazo, de tal manera que podamos enfocarnos en mejorar algunos aspectos.

Considerar una mejor estructura en la base de datos, que sea más practica y eficaz, se recomienda usar el servidor recomendado es MySQL el cual cuenta con documentación más detallada y cuya interfaz es amigable con el usuario.

5.3 Bibliografía

- [1] J. Pérez Porto and A. Gardey, "Definición de optimización — Definicion.de", *Definición.de*, 2009. [Online]. Available: <https://definicion.de/optimizacion/>. [Accessed: 09- Oct- 2019].
- [2] "Definición de Optimizar software o hardware (informática)", *Alegsa.com.ar*, 2016. [Online]. Available: <http://www.alegsa.com.ar/Dic/optimizar.php>. [Accessed: 09- Oct- 2019].
- [3] "Definición de Optimizar [Qué es, Significado y Concepto] ", *Concepto y Definición*, 2019. [Online]. Available: <https://conceptodefinicion.net/definicion-de-optimizar/>. [Accessed: 09- Oct- 2019].
- [4] "Aplicación web - EcuRed", *Ecured.cu*, 2019. [Online]. Available: https://www.ecured.cu/Aplicaci%C3%B3n_web. [Accessed: 09- Oct- 2019].
- [5] J. Pereira, "Aplicaciones Web y ASPs | MERCADEO.COM", *Mercadeo.com*, 2010. [Online]. Available: <http://www.mercadeo.com/blog/2010/01/aplicaciones-web-y-asps/>. [Accessed: 09- Oct- 2019].
- [6] D. Velázquez Álvarez, "Aplicaciones web Vs Aplicaciones de escritorio | Blog de informática y tecnología | Consultoría Informática", *Webprogramacion.com*, 2012. [Online]. Available: <https://webprogramacion.com/356/blog-informatica-tecnologia/aplicaciones-web-vs-aplicaciones-de-escritorio.aspx>. [Accessed: 11- Oct- 2019].
- [7] "Buyto - ¿Web o Desktop? Principales diferencias entre aplicaciones web y desktop - Ventajas y desventajas de aplicaciones web y desktop | Alcobendas, Madrid, España", *Buyto.es*, 2019. [Online]. Available: <http://www.buyto.es/general-diseno-web/diferencias-entre-aplicaciones-web-y-aplicaciones-desktop.html>. [Accessed: 11- Oct- 2019].
- [8] "Aplicación informática - EcuRed", *Ecured.cu*, 2019. [Online]. Available: https://www.ecured.cu/Aplicaci%C3%B3n_inform%C3%A1tica. [Accessed: 11- Oct- 2019].
- [9] "Aplicaciones informáticas - Tecnomundo", *Tecnomundo*, 2016. [Online]. Available: <http://tecnomundo.es/aplicaciones-informaticas/>. [Accessed: 11- Oct- 2019].
- [10] D. tecnología, "Definición de Aplicación (informática)", *Alegsa.com.ar*, 2010. [Online]. Available: <http://www.alegsa.com.ar/Dic/aplicacion.php>. [Accessed: 11- Oct- 2019].
- [11] "Framework - EcuRed", *Ecured.cu*, 2019. [Online]. Available: <https://www.ecured.cu/Framework>. [Accessed: 11- Oct- 2019].
- [12] "¿Qué es un Framework y para que sirve? - Neo Wiki | NeoAttack", *Neoattack*, 2019. [Online]. Available: <https://neoattack.com/neowiki/framework/>. [Accessed: 11- Oct- 2019].
- [13] F. Chaluisa, *Programacion-en-c.wixsite.com*, 2017. [Online]. Available: <https://programacion-en-c.wixsite.com/programacion-en-c/librerias-y-tipo-de-datos>. [Accessed: 12- Oct- 2019].

[14]"49. ¿Qué son las librerías y las APIs? - Diego Laballós", *Diego Laballós*, 2018. [Online]. Available: <https://diegolaballos.com/podcast/49-que-son-las-librerias-y-las-apis/>. [Accessed: 12- Oct- 2019].

[15]"Librerías para la programación web", *Aprende-web.net*, 2018. [Online]. Available: <https://aprende-web.net/librerias/>. [Accessed: 12- Oct- 2019].

5.4 Glosario

Optimizar

Conseguir que algo llegue a la situación óptima (obtener los mejores resultados utilizando la menor cantidad de recursos y manteniendo la calidad).

Modelo

El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.

Vista

La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.

Controlador

El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

Componente

Un componente es una unidad de composición de aplicaciones software, que posee un conjunto de interfaces y un conjunto de requisitos, y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes de forma independiente, en tiempo y espacio.

Arquitectura

Es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación.

Framework

Un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado

Modulo

Un módulo es una porción de un programa de ordenador. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente, una de dichas tareas (o varias, en algún caso).

Implementación

Una implementación es la ejecución u/o puesta en marcha de una idea programada, ya sea, de una aplicación informática, un plan, modelo científico, diseño específico, estándar, algoritmo o política.