



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

**“DESARROLLO DE UN SISTEMA WEB PARA LA FACTURACIÓN DE LA
LAVADORA Y LUBRICADORA “LOS ÁNGELES” APLICANDO EL
FRAMEWORK LARAVEL”**

TRABAJO DE TITULACIÓN
TIPO: PROYECTO TÉCNICO

Presentado para optar al grado académico de:
INGENIERA EN SISTEMAS INFORMÁTICOS

AUTORA: MARÍA VANESSA YUPANGUI CHULLI
TUTOR: ING. MAYRA ALEJANDRA OÑATE ANDINO

Riobamba-Ecuador
2019

@2019, María Vanessa Yupangui Chulli

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELÉCTRICA
ESCUELA DE INGENIERÍA EN SISTEMAS

El tribunal del trabajo de Titulación certifica que: El proyecto Técnico: “**DESARROLLO DE UN SISTEMA WEB PARA LA FACTURACIÓN DE LA LAVADORA Y LUBRICADORA “LOS ÁNGELES” APLICANDO EL FRAMEWORK LARAVEL**”, de responsabilidad de la señorita María Vanessa Yupangui Chulli, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

FIRMA

FECHA

Ing. Washington Luna

**DECANO DE LA FACULTAD DE
INFORMÁTICA Y
ELECTRÓNICA**

Ing. Patricio Moreno

**DIRECTOR DE LA ESCUELA DE
INGENIERÍA EN SISTEMAS**

Ing. Alejandra Oñate

**DIRECTORA DE TRABAJO DE
TITULACIÓN**

Ing. Raúl Rosero

MIEMBRO DEL TRIBUNAL

Yo, María Vanessa Yupangui Chulli soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo.

María Vanessa Yupangui Chulli

DEDICATORIA

El siguiente trabajo de titulación se lo dedico primeramente a Dios por permitirme culminar con esta etapa profesional, a mis padres por su apoyo y esfuerzo incondicional. Y a esas dos personitas especiales, que me llenan de amor y me impulsan a cumplir mis metas Alejandro y Sebastián.

María

AGRADECIMIENTO

Agradezco a todas aquellas personas que contribuyeron con la culminación de esta etapa de mi vida profesional, a mi familia, amigos, docentes y compañeros. A mis tutores de mi trabajo de titulación, que me guiaron en este proceso, además agradecer a la ESPOCH por su gran dedicación en formar profesionales de éxito.

María

TABLA DE CONTENIDO

INDICE DE TABLAS.....	xiii
INDICE DE FÍGURAS.....	xv
INDICE DE GRÁFICOS.....	xvii
RESUMEN.....	xix
ABSTRACT.....	xx
INTRODUCCIÓN	1

CAPÍTULO I

1. MARCO TEORICO.....	8
1.1. Aplicaciones web	8
<i>1.1.1. Ventajas</i>	<i>8</i>
1.2. Sistemas de facturación	9
<i>1.2.1. Factura</i>	<i>9</i>
<i>1.2.1.1. Características de una factura</i>	<i>9</i>
1.3. Framework de desarrollo	10
<i>1.3.1. ¿Qué es un framework?.....</i>	<i>10</i>
<i>1.3.2. Framework Laravel</i>	<i>10</i>
<i>1.3.2.1. Características</i>	<i>11</i>
<i>1.3.3. Comparativa entre frameworks de PHP</i>	<i>12</i>
<i>1.3.4. Estructura del directorio de carpetas Laravel</i>	<i>13</i>
1.4. Servidor web.....	15
<i>1.4.1. Apache.....</i>	<i>16</i>
<i>1.4.1.1. Características del servidor Apache</i>	<i>17</i>
1.5. Arquitectura de software.....	17
1.6. Comparativa de arquitecturas.....	19

1.7.	Arquitectura MVC	20
1.7.1.	Modelo	20
1.7.2.	Vista	21
1.7.3.	Controlador	22
1.7.4.	Funcionamiento de la arquitectura MVC – Laravel	22
1.7.5.	Ventajas	23
1.8.	Base de datos	24
1.8.1.	MySQL	24
1.8.1.1.	Características:	24
1.9.	Otras herramientas para el desarrollo de la aplicación	25
1.9.1.	Navegadores web	25
1.9.2.	Sublime Text	25
1.9.3.	Lenguaje de programación	26
1.9.3.1.	PHP	26
1.9.3.2.	Características	26
1.9.4.	JavaScript	27
1.9.5.	Ajax	28
1.9.6.	Chart.js	28
1.9.7.	HTML5	29
1.9.8.	CSS3	29
1.9.9.	Bootstrap	30
1.9.9.1.	Ventajas	30
1.10.	Metodología de desarrollo SCRUM	31
1.10.1.	Beneficios	32
1.10.2.	Roles	32
1.10.3.	Artefactos	33
1.10.4.	Reuniones	34
1.11.	Herramienta de gestión de pruebas	35
1.11.1.	TestLink	35

<i>1.11.2.</i>	<i>Unidades básicas utilizadas por TestLink</i>	35
<i>1.11.3.</i>	<i>Estructura de TestLink</i>	36
<i>1.11.4.</i>	<i>Estructura de TestLink con la metodología SCRUM</i>	36
<i>1.11.5.</i>	<i>Beneficios</i>	37
1.12.	ISO/IEC 9126	38
<i>1.12.1.</i>	<i>Características de ISO/IEC 9126</i>	38
<i>1.12.1.1.</i>	<i>Eficiencia</i>	39

CAPÍTULO II

2.	MARCO METODOLÓGICO	41
2.1.	Metodología de la investigación	41
<i>2.1.1.</i>	<i>Tipo, métodos y técnicas de investigación</i>	41
<i>2.1.1.1.</i>	<i>Tipo de investigación</i>	41
<i>2.1.1.2.</i>	<i>Métodos de investigación</i>	41
<i>2.1.1.3.</i>	<i>Técnicas de investigación</i>	42
2.2.	Metodología para el desarrollo de software SCRUM	42
2.3.	Fase de planificación	43
<i>2.3.1.</i>	<i>Reuniones</i>	43
<i>2.3.2.</i>	<i>Análisis de procesos</i>	44
<i>2.3.3.</i>	<i>Especificación de requerimientos</i>	46
<i>2.3.3.1.</i>	<i>Propósito</i>	46
<i>2.3.3.2.</i>	<i>Alcance</i>	46
<i>2.3.3.3.</i>	<i>Restricciones</i>	46
<i>2.3.3.4.</i>	<i>Requerimientos funcionales</i>	47
<i>2.3.3.5.</i>	<i>Requerimientos no funcionales</i>	49
<i>2.3.4.</i>	<i>Planificación</i>	49
<i>2.3.5.</i>	<i>Roles y personas</i>	51
<i>2.3.6.</i>	<i>Tipos y roles de usuarios</i>	51
<i>2.3.7.</i>	<i>Pila de producto</i>	51

2.3.8.	<i>Análisis económico</i>	55
2.4.	Fase de diseño	58
2.4.3.	<i>Diagrama de casos de uso</i>	58
2.4.4.	<i>Diagrama de clases</i>	59
2.4.5.	<i>Diagrama de componentes</i>	61
2.4.6.	<i>Recursos necesarios</i>	61
2.4.6.1.	<i>Recursos hardware</i>	61
2.4.6.2.	<i>Recursos software</i>	62
2.4.7.	<i>Estándar de codificación</i>	63
2.4.8.	<i>Diseño de la interfaz de usuario</i>	63
2.4.8.5.	<i>Página de inicio del sistema web</i>	64
2.4.8.6.	<i>Pantalla de inicio de sesión</i>	64
2.4.8.7.	<i>Página principal administrador</i>	65
2.4.8.8.	<i>Interfaz de formularios de ingreso de datos</i>	66
2.4.8.9.	<i>Interfaz de registro de ventas</i>	66
2.4.8.10.	<i>Vista de la Interfaz listado de compras dispositivo móvil</i>	67
2.4.9.	<i>Diseño de la base de datos</i>	68
2.4.9.5.	<i>Diccionario de datos</i>	70
2.5.	Fase de desarrollo	70
2.5.3.	<i>Sprint backlog</i>	70
2.5.4.	<i>Historias de usuario</i>	87
2.6.	Instalación, configuración y codificación	88
2.7.	Plan de pruebas	95
2.8.	Gestión del proyecto	97
2.8.3.	<i>BurnDown Chart</i>	97
2.9.	Métodos utilizados para evaluar la eficiencia del sistema web	98
2.9.3.	<i>Estudio comparativo sistema manual vs. sistema automatizado</i>	98
2.9.3.5.	<i>Procesos a evaluar la eficiencia</i>	99

CAPÍTULO III

3.	MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS RESULTADOS .	100
3.3.	Diseño del pre experimento	100
3.4.	Desarrollo del pre experimento	101
3.5.	Análisis de prueba T-Student para muestras relacionadas.....	104
3.5.3.	<i>Planteamiento de hipótesis</i>	<i>104</i>
3.5.4.	<i>Nivel de significancia</i>	<i>104</i>
3.5.5.	<i>Estadístico de prueba</i>	<i>104</i>
3.5.6.	<i>Regla de decisión</i>	<i>105</i>
3.5.7.	<i>Toma de decisión</i>	<i>105</i>
3.6.	Análisis de medias	106

CONCLUSIONES.....	107
--------------------------	------------

RECOMENDACIONES.....	108
-----------------------------	------------

BIBLIOGRAFIA

ANEXOS

ÍNDICE DE ABREVIATURAS

ACID	Atomicity, Consistency, Isolation, Durability
AJAX	Asynchronous JavaScript and XML
CLI	Command line interface
CSS	Cascading Style Sheets
GNU	Licencia Pública General
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
MVC	Modelo Vista Controlador
ORM	Object-Relational mapping
PDF	Portable Document Format
PHP	Hypertext Pre-Processo
RAM	Random Access Memory
RUC	Registro Único de Contribuyentes
SQL	Structured Query Language
SRI	Servicio de Rentas Internas
SSL	Secure Sockets Layer
UML	Unified Modeling Language
URL	Uniform Resource Locator

INDICE DE TABLAS

Tabla 1-1: Comparativo de frameworks de PHP	12
Tabla 2-1: Comparativa de arquitecturas de software.....	19
Tabla 3-1: Características de la norma ISO/IEC 9126.....	38
Tabla 1-2: Reuniones en la fase de planificación.....	43
Tabla 2-2: Roles y personas	51
Tabla 3-2: Tipos y roles de usuarios	51
Tabla 4-2: Talla de la camiseta	52
Tabla 5-2: Product Backlog	53
Tabla 6-2: Presupuesto de hardware	56
Tabla 7-2: Presupuesto de software	56
Tabla 8-2: Presupuesto materiales adicionales	57
Tabla 9-2: Presupuesto mano de obra.....	57
Tabla 10-2: Resumen de presupuesto del proyecto.....	58
Tabla 11-2: Recursos hardware.....	62
Tabla 12-2: Recursos software.....	62
Tabla 13-2: Estándar de codificación.....	63
Tabla 14-2: Tabla - personas.....	70
Tabla 15-2: Sprint Backlog	71
Tabla 16-2: Sprint 1	75
Tabla 17-2: Sprint 2	76
Tabla 18-2: Sprint 3	76
Tabla 19-2: Sprint 4	77
Tabla 20-2: Sprint 5	78
Tabla 21-2: Sprint 6	78
Tabla 22-2: Sprint 7	79
Tabla 23-2: Sprint 8	80
Tabla 24-2: Sprint 9	80
Tabla 25-2: Sprint 10	81
Tabla 26-2: Sprint 11	82
Tabla 27-2: Sprint 12	82
Tabla 28-2: Sprint 13	83
Tabla 29-2: Sprint 14	84

Tabla 30-2: Sprint 15	84
Tabla 31-2: Sprint 16	85
Tabla 32-2: Sprint 17	86
Tabla 33-2: Sprint 18	86
Tabla 34-2: Historia de usuario 29 - Sprint 10.....	87
Tabla 1-3: Muestra para el pre experimento	101
Tabla 2-3: Prueba estadística T-Student para medias de dos muestras relacionadas	104
Tabla 3-3: Regla de decisión.....	105

INDICE DE FÍGURAS

Figura 1-1: Estructura de carpetas y directorios Laravel	14
Figura 2-1: Servidor web Apache y MySQL.....	17
Figura 3-1: Estructura de un archivo layout	22
Figura 4-1: Comportamiento de la arquitectura MVC y el framework Laravel	23
Figura 5-1: Esquema general de la metodología SCRUM.....	32
Figura 6-1: Estructura de la herramienta TestLink	36
Figura 7-1: Estructura de la herramienta TestLink y la metodología SCRUM	37
Figura 8-1: Características de la eficiencia.....	40
Figura 1-2: Diagrama de procesos general de facturación.....	44
Figura 2-2: Diagrama de procesos detallados de la facturación	45
Figura 3-2: Diagrama de Gantt	50
Figura 4-2: Diagrama caso de uso administrador, empleado.....	59
Figura 5-2: Diagrama de clases	60
Figura 6-2: Arquitectura del sistema	61
Figura 7-2: Página de inicio de la empresa.....	64
Figura 8-2: Página de inicio de sesión	65
Figura 9-2: Página principal - administrador.....	65
Figura 10-2: Interfaz de usuario - registro de artículos.....	66
Figura 11-2: Interfaz de usuario - registro de una venta	67
Figura 12-2: Interfaz vista móvil	67
Figura 13-2: Diagrama de base de datos - modelo lógico	69
Figura 14-2: Descargar gestor de dependencias Composer.....	88
Figura 15-2: Comprobación del funcionamiento de composer.....	89
Figura 16-2: Creación del proyecto Laravel 5.6	89
Figura 17-2: Página de bienvenida de Laravel	90
Figura 18-2: Directorio de archivos y carpetas del sistema	91
Figura 19-2: Controlador del sistema - VentaController	92
Figura 20-2: Modelo del sistema - Venta	93
Figura 21-2: Archivo de migración - 2018_09_09_033145_create_ventas_table.....	93
Figura 22-2: Vista del sistema - Venta	94
Figura 23-2: Rutas del sistema.....	95
Figura 24-2: Plan de pruebas - registro de ventas.....	96
Figura 25-2: Detalle de casos de prueba - registro de ventas.....	97

Figura 1-3: Test estadístico Kolmogorov Smirnov.....	103
---	-----

INDICE DE GRÁFICOS

Gráfico 1-2: BurnDown Chart del proyecto.....	98
Gráfico 1-3: Análisis de datos mediante la Campana de Gauss.....	105
Gráfico 2-3: Sistema manual vs. Sistema automatizado.....	106

INDICE DE ANEXOS

ANEXO A: Diccionario de datos

ANEXO B: Historias de usuario e Historias técnicas

ANEXO C: Plan de pruebas del proyecto

ANEXO D: Manual de usuario

RESUMEN

El presente trabajo de titulación tuvo como finalidad desarrollar un sistema web para la facturación de la Lavadora y Lubricadora “Los Ángeles” aplicando el framework Laravel. En la fase inicial del proyecto se determinó el proceso de facturación que realiza la empresa aplicando las técnicas de observación y entrevista. Para el desarrollo del sistema se utilizó la metodología ágil SCRUM, mediante la cual se creó un ambiente colaborativo entre el cliente y el desarrollador obteniendo 53 historias de usuario y 12 historias técnicas, además se definieron los roles de usuarios, administrador y empleado. Para el desarrollo de los módulos se utilizaron herramientas tecnológicas como el framework Laravel que permitió el desarrollo del sistema de autenticación, mapeo de objetos relacionales (ORM), plantillas Blade, sistema de rutas, líneas de comando Artisan que permitirán la interacción con la aplicación. Como gestor de base de datos se utilizó MySQL. Por otra parte, el sistema usa la arquitectura Modelo-Vista-Controlador permitiendo la mantenibilidad y escalabilidad; además, fue alojado en el servidor web Apache. También, se utilizó la herramienta Test Link con el objetivo de realizar una detección y corrección de errores de las funcionalidades que componen el sistema. Finalmente se evaluó la eficiencia de la aplicación en el proceso de facturación mediante la evaluación de los tiempos de respuesta del sistema manual y el sistema automatizado. El análisis de los datos obtenidos se realizó a través del método estadístico T- Student, obteniéndose como resultado una diferencia significativa por lo cual se concluye que el tiempo promedio del sistema manual es mayor al tiempo promedio del sistema automatizado en un 51.78%. Por los resultados obtenidos se recomienda aplicar el framework Laravel ya que permite desarrollar sistemas web robustos en menor tiempo y con un menor costo.

PALABRAS CLAVE: <DESARROLLO DE APLICACIONES WEB>, <GESTIÓN DE FACTURACION >, <MODELO VISTA CONTROLADOR (MVC)>, <FRAMEWORK LARAVEL>, <EFICIENCIA>.

ABSTRACT

The purpose of the present thesis project was to develop a web system for billing the "Los Angeles" Washer and Lubricator by applying the Laravel framework. In the initial phase of the project, the billing process performed by the company using observation and interview techniques was completed. The agile SCRUM methodology was used for the development of the system, through which a collaborative environment was created between the client and the developer, obtaining 53 user stories and 12 technical histories, as well as defining the roles of users, administrator and employee. For the development of the modules technological tools were used as framework Laravel that allowed the development of the authentication system, mapping of relations objects (ORM), Blade templates, route system, Artisan command lines that will allow interaction. As a database manager, MySQL was used. On the other hand, the system uses the Model-View-Controller architecture allowing maintainability and scalability; In addition, it was hosted on the Apache web server. Also, the Test Link tool was used in order to perform a detection and correction of errors of the functionalities that make up the system. Finally, the efficiency of the application in the billing process was evaluated by evaluating the response types of the manual system and the automated system. The analysis of the obtained data was carried out through the T-Student statistical method, obtaining as a result the significant difference, for which it is concluded that the average time of the manual system is greater than the average time of the automated system in 51.78%. Based on the results obtained, it is recommended to apply the Laravel framework, since it allows the development of robust web systems in less time and at a lower cost.

KEY WORDS: <DEVELOPMENT OF WEB APPLICATIONS>, <BILLING MANAGEMENT>, <MODEL VIEW CONTROL (MVC)>, <FRAMEWORK LARAVEL>, <EFFICIENCY>.

INTRODUCCIÓN

En la actualidad existen numerosos cambios en los mercados, competencias, organizaciones, tecnologías, sociedades y culturas, entre otros, razón por la cual se considera poco pertinente seguir maniobrando bajo el mismo enfoque tradicional. Para lograr ser competitivo dentro de este entorno tan cargado de dinamismo y turbulencia, es indispensable buscar ventajas competitivas y por ende un desarrollo económico a largo plazo, así como también desarrollar capacidad para producir, circular y utilizar correctamente la información, la comunicación y el conocimiento, por cuanto ellos constituyen la materia prima de esta nueva sociedad (Cano Pita, 2018, pp-501-502).

Hoy en día el uso de la tecnología (sistemas web) por las empresas es fundamental ya que da a conocer sus productos y/o servicios con el objetivo de ganar popularidad y prestigio ante sus clientes. Otra de las ventajas de usar estos sistemas web es que facilita al personal administrativo a llevar un control, y registro de sus compras y ventas.

En el Ecuador toda empresa que realice alguna actividad económica está obligado por la ley a emitir facturas o algún comprobante de venta para justificar sus ingresos y egresos ante en Servicio de Rentas Internas (SRI).

En este contexto el presente trabajo de titulación, ha realizado un estudio, análisis, desarrollo de un sistema web para ayude a automatizar el proceso de facturación de la Lavadora y Lubricadora “Los Ángeles” con el objetivo de disminuir el tiempo en su elaboración. Para llevar a cabo el desarrollo del sistema, se ha utilizado la metodología ágil SCRUM para obtener un software de calidad y una mejor comunicación entre los miembros involucrados, además se ha hecho el estudio y análisis previo del framework de desarrollo Laravel para determinar sus ventajas frente a otros frameworks de PHP.

Para el desarrollo del sistema web se ha utilizado las siguientes herramientas: Laravel como herramienta de desarrollo, lenguaje de programación PHP, para el diseño de las interfaces el

framework Bootstrap, como editor de texto Sublime Text, para el almacenamiento de los datos MySQL, como servidor web Apache.

El presente documento está compuesto de tres capítulos los cuales se describen a continuación:

Capítulo I: Marco teórico en este capítulo se describe definiciones y conceptos de las herramientas utilizadas en el desarrollo.

Capítulo II: Marco metodológico, en este capítulo contiene las actividades realizadas en el desarrollo del sistema siguiendo la metodología SCRUM.

Capítulo III: Marco de resultados, este capítulo contiene los resultados del sistema luego haber sido evaluado.

Antecedentes

Actualmente las tecnologías de información y comunicación a causado un gran impacto en la sociedad, estas herramientas computacionales e informáticas ayudan procesar la información y tiene como objetivo mejorar y facilitar la calidad de vida de las personas.

Los sistemas de información están cambiando en la actualidad la forma en que operan las organizaciones. Mediante su uso se obtienen grandes mejoras, ya que automatizan los procesos operativos que se pueden llevar a cabo en toda empresa, proporcionan información de apoyo al proceso de tomas de decisiones, y facilitan el logro de ventajas competitivas a través de su implantación dentro de la organización (Hamidian Fernández y Ospino Sumoza, 2015, p.164).

Los sistemas de información han llegado para quedarse por su gran utilidad como herramienta complementaria en diferentes áreas, ya sea en lo personal, empresarial (gestión de recursos humanos, procesamiento de transacciones, gerencial o administrativas, toma de decisiones) y comunicacional, entre otros (Hamidian Fernández and Ospino Sumoza, 2015, p.164). Las aplicaciones web se ha convertido en una parte esencial de toda empresa, las empresas se están adaptando con mucha rapidez a los cambios que se presentan en el entorno por causa de la alta competencia y la dinámica de un mercado cambiante.

El presente estudio se lleva a cabo en Lavadora y Lubricadora “Los Ángeles”, que es una empresa dedicada al mantenimiento y venta de lubricantes para todo tipo de vehículos. Este negocio imparte el servicio de cambio de aceite, engrasada, aspirado, pulverizada, servicio de lavado, de vehículos livianos, pesados y automotores de dos ruedas, está ubicada en las calles 18 de marzo entre Joaquín Gavilánez y Gilberto Obando en el cantón Chambo.

Actualmente la lavadora y lubricadora “Los Ángeles”, no cuenta con un sistema software que ayude con la facturación de la salida de mercadería. Ya que posee una gran demanda de clientes la cual causa la demora excesiva en la atención a los mismos.

Formulación del problema

¿El sistema web para la Lavadora y Lubricadora “Los Ángeles” reducirá el tiempo de atención a los clientes en el proceso de facturación?

Sistematización del problema

¿Cómo se maneja actualmente la facturación en el negocio?

¿Cuál es la arquitectura más adecuada para el sistema a desarrollar?

¿Cuáles son las funcionalidades que deberá ofrecer el sistema de facturación de la lavadora y lubricadora “Los Ángeles”?

¿Qué grado de eficiencia otorgará el sistema en el proceso de facturación a la lavadora y lubricadora “Los Ángeles”?

Justificación de la investigación

- **Justificación teórica**

Una aplicación web es una aplicación informática distribuida cuya interfaz de usuario es accesible desde un cliente web, normalmente un navegador web (Sevilla, 2004, p.2). Estos sistemas web tiene el aspecto de una página web estática sin embargo una de las ventajas de estos, es que no necesitan ser instalados en el ordenador, ya que se encuentran alojados en un servidor y esto facilita el acceso a los usuarios, también hace uso de una base de datos la cual permite almacenar, procesar y visualizar la información requerida por el usuario.

De acuerdo con (Mora, 2002, p.3), Las aplicaciones web permiten la generación automática de contenido, la creación de páginas personalizadas según el perfil del usuario o el desarrollo del comercio electrónico. Además, una aplicación web permite interactuar con los sistemas informáticos de gestión de una empresa, como puede ser gestión de clientes contabilidad o inventario, a través de una página web.

Un software que gestione la facturación en una empresa, permitirá ahorrar tiempo ya que las facturas no se realizan de manera manual. Una factura es un documento el cual refleja la compra o venta de un bien, productos o servicios, en este documento se indica la fecha, cantidad, valor unitario, valor total, impuesto, además de los datos del destinatario.

Actualmente la lavadora y lubricadora “Los Ángeles”, no cuenta con un sistema software para la facturación de sus productos, por lo cual hay una demora excesiva en la atención en sus clientes, además el no llevar un correcto registro de sus compras y ventas puede causar sanciones por parte del Servicio de Rentas Internas y la pérdida de clientes por ende causar pérdidas económicas para la empresa.

Con el propósito de solucionar el problema expuesto anteriormente se desarrollará una aplicación web ya que aparte de automatizar el proceso de facturación también servirá para realizar publicidad de los servicios y productos que ofrece la empresa.

El desarrollo del sistema se lo realizará usando el framework Laravel el cual permitirá al desarrollador elaborar el sistema de una manera rápida, fácil y a menor al costo para la empresa, ya que posee características como: un sistema de autenticación, ORM (Object Relational Mapping), sistema de ruteo, arquitectura MVC, línea de comando Artisan que permitirá la interacción con la aplicación, motor plantillas blade que permitirá la herencia de secciones mediante una plantilla base o layout, de esta manera evitará que el código se repita. Laravel posee una extensa documentación y tiene una curva de aprendizaje relativamente baja en relación con otros frameworks de PHP, proporciona las herramientas necesarias para la construcción de sistemas web robustas.

Para complementar el desarrollo de la aplicación se usará otras herramientas como: lenguaje de programación PHP que es libre y de código abierto, para el diseño de las interfaces se utilizará el framework Bootstrap ya que este contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript, para almacenar los datos y mantener su integridad, disponibilidad se utilizará MySQL el cual es un gestor de base de datos relacional desarrollado bajo licencia dual. Por medio de la herramienta TestLink se realizará una detección y corrección de errores del sistema,

validando que las funcionalidades estén de acuerdo a los requerimientos solicitados por el gerente de la empresa.

- **Justificación aplicativa**

En la lavadora y lubricadora “Los Ángeles” existe una concurrencia de clientes que solicitan una factura, este proceso se lo realiza manualmente lo cual causa una demora para el cliente.

Con el sistema web propuesto además de ser intuitivo para los usuarios la empresa ahorrará tiempo y recursos, ya que al automatizar este proceso se reducirá el tiempo de manera significativa al personal administrativo y a sus clientes.

A continuación, se detalla los módulos del sistema a realizar:



Realizado por: Maria Yupangui, 2019

El software que se desarrollará se lo realizará siguiendo las líneas de investigación de la EIS-ESPOCH (Programa Nacional Toda una Vida).

Objetivos

Objetivo general

Desarrollar una aplicación web para la facturación de la Lavadora y Lubricadora “Los Ángeles” aplicando el framework laravel y la metodología ágil SCRUM.

Objetivos específicos

- ✓ Analizar los procesos que realiza la lavadora y lubricadora.
- ✓ Recopilar los requerimientos necesarios, diseñar la base de datos y la arquitectura para el desarrollo del sistema de facturación.
- ✓ Desarrollar los módulos aplicando el Framework Laravel, lenguaje de programación PHP y la metodología SCRUM.
- ✓ Evaluar la eficiencia del sistema en el proceso de facturación posterior a su implementación.

CAPÍTULO I

1. MARCO TEÓRICO

1.1. Aplicaciones web

Las aplicaciones web se han convertido en una parte muy importante a nivel empresarial, educación y social ya que es una herramienta que actualmente se lo usa a nivel mundial.

Desde el punto de vista de la arquitectura se distinguen dos lados; uno es el cliente, donde se encuentra el usuario final utilizando la aplicación por medio de un navegador (como Internet Explorer o Mozilla Firefox). A través de este cliente web, el usuario interactúa con la aplicación localizada al otro lado, en el servidor, que es donde residen realmente los datos, reglas y lógica de la aplicación (Ferrer, 2012, p.18).

1.1.1. Ventajas

Las ventajas del uso de las aplicaciones web se describe a continuación:

- No necesita ser instalado en el ordenador.
- Facilita el trabajo en grupo.
- Permite ahorrar costes de hardware y software.
- Es multiplataforma
- Compatible con los navegadores
- Portabilidad
- Escalables y actualizables
- Fáciles de usar.

1.2. Sistemas de facturación

Los sistemas de facturación son una herramienta importante que aporta muchas ventajas en los negocios y grandes empresas ya que facilita en gran parte el trabajo manual para la elaboración de facturas (Velásquez Valle y Zeledón Bustillo, 2014, p.12).

Hay que tomar en cuenta que en las facturas se pueden ver todos los movimientos de dineros tanto, como entradas y salidas, también estas son un apoyo para los cálculos contable, una empresa debe de contar con el registro de todas las facturas ya que en estas están son el soporte claro de la contabilidad, por lo tanto, contar con un sistema de facturación se garantiza un registro exacto y no vulnerable a pérdidas (Velásquez Valle y Zeledón Bustillo, 2014, p.12).

1.2.1. Factura

Una factura es una relación de los objetos o artículos comprendidos en una venta, remesa u otra operación de comercio, es el documento que funciona como respaldo y prueba física de la realización de una operación económica, es común, ver este tipo de acuerdos en una compra-venta (Vergara Sandoval, 2017, p.11).

1.2.1.1. Características de una factura

Una factura posee las siguientes características la cual determina si es válida o no.

- Número del registro único de contribuyentes de la persona o empresa que emite el documento.
- Apellidos, nombres, denominación o razón social conforme conste en Registro Único de Contribuyentes (RUC), adicionalmente puede incluir el nombre de fantasía o comercial en caso de existir.
- Número telefónico del establecimiento
- La denominación del documento.
- Numeración del documento
- Dirección del establecimiento.

- Fecha de caducidad del documento expresada en día, mes y año según la autorización del Servicio de Rentas Internas (SRI).
- Número de RUC, nombres y apellidos, denominación o razón social y número otorgado por el SRI del establecimiento gráfico en el cual se realizó la impresión.

1.3. Framework de desarrollo

1.3.1. ¿Qué es un framework?

Un Framework es una plataforma de software universal y reutilizable para desarrollar aplicaciones de software, productos y soluciones. En otras palabras, podemos decir que es una especie de biblioteca, una pieza de software que proporciona a los desarrolladores web una base de código y formas consistentes y estandarizadas para crear aplicaciones web (Ortiz, 2018, <http://blog.hostdime.com.co/que-es-un-framework-informatica-programacion/>).

Según (Sánchez Pedrós, 2017, p.12) Un framework web es una estructura software compuesta de diferentes componentes que facilitan el desarrollo de una aplicación. La mayoría de los frameworks web funcionan de una manera muy similar. Los framework web tienen como objetivo que el desarrollo de estas aplicaciones o sistemas web se haga de forma elegante y simple, sin necesidad de reescribir código.

1.3.2. Framework Laravel

Laravel es el nombre de un framework creado para trabajar con PHP (Hostalia, 2016, p.3), su filosofía es desarrollar código PHP de forma elegante y simple basado en un modelo MVC (Sierra, 2018, <https://www.synergyweb.es/blog/laravel-desarrollo-medida/>).

El objetivo de Laravel es el de ser un framework que permita el uso de una sintaxis refinada y expresiva para crear código de forma sencilla, evitando el “código espagueti” y permitiendo multitud

de funcionalidades. Aprovecha todo lo bueno de otros frameworks y utiliza las características de las últimas versiones de PHP (Baquero García, 2015, <https://www.arsys.es/blog/programacion/que-es-laravel/>).

Fue desarrollado por Taylor Otwell, intentando mostrar una alternativa más completa respecto a otros frameworks del mercado. Su primer lanzamiento fue el 9 de junio de 2011. En la actualidad, Laravel es uno de los frameworks web más utilizados, contando con una gran comunidad creciente (Sánchez Pedrós, 2017, p.14).

1.3.2.1. Características

El framework laravel posee las siguientes características que se describe a continuación:

- **Arquitectura MVC:** Es un patrón de diseño de arquitectura de software usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de conceptos para que el desarrollo esté estructurado de una mejor manera, facilitando la programación en diferentes capas de manera paralela e independiente (Rivera, 2008, p.11).
- **Object relational mapping (ORM):** Utilizado para la transformación de tablas de una base datos en entidades (modelos del sistema), facilitando las tareas básicas con la base de datos (crear, modificar, consultar o eliminar). El ORM utilizado por Laravel se llama Eloquent (Sánchez Pedrós, 2017, p.13).
- **Sistema de rutas:** Laravel maneja un sistema de rutas, el cual facilita en el desarrollo estas rutas pueden ser de tipo GET y POST.
- **Motor de plantillas Blade:** Posee un sistema de platillas, el cual facilitara a la creación de vistas por medio de herencia de secciones.
- **Migraciones:** Son una serie de archivos que proveen un sistema de control de versiones para las bases de datos, utilizadas tanto para crear tablas, como hacer modificaciones o eliminar tablas, columnas, etc. a través de la programación orientada a objetos (Sánchez Pedrós, 2017, p.13).
- **Composer:** es una herramienta para la gestión de dependencias en PHP. Permite declarar las librerías y paquetes de las que un proyecto depende y las gestiona (instala o actualiza) de forma automática, quitándole trabajo al desarrollador (Sánchez Pedrós, 2017, p.39). Composer funciona mediante línea de comandos.

- **Artisan:** Entre las herramientas que Laravel nos proporciona para el desarrollo de aplicaciones se encuentra Artisan, la interfaz de línea de comandos (CLI por sus siglas en inglés de Command-line interface), la cual es un medio para la interacción con la aplicación donde los usuarios (en este caso los desarrolladores) dan instrucciones en forma de línea de texto simple o línea de comando. Artisan está basado en el componente Console de Symfony y nos ofrece un conjunto de comandos que nos pueden ayudar a realizar diferentes tareas durante el desarrollo e incluso cuando la aplicación se encuentra en producción (Rondón, 2016, <https://styde.net/artisan-interfaz-linea-comandos-de-laravel/>).
- **Autenticación simple:** Laravel incluye un módulo de autenticación basado en sesiones, facilitando al desarrollador funcionalidades como el login, el registro, comprobación de contraseñas, etc. (Sánchez Pedrós, 2017, p.13).

1.3.3. Comparativa entre frameworks de PHP

En el mercado existe una gran variedad de frameworks para PHP a continuación, se visualiza una tabla comparativa, la cual nos ayudará a entender porque el framework Laravel fue elegido en el desarrollo del sistema web de facturación. Ver **Tabla 1-1**.

Tabla 1-1: Comparativo de frameworks de PHP

Características	Laravel	Phalcon	Symfony	CodeIgniter
Object relational mapping (ORM)	X	X	-	-
Motor de plantillas	X	X	X	X
Ajax	X	X	X	-
Patrón Modelo - Vista - Controlador (MVC)	X	X	X	X
Soporte para múltiple BD	X	X	X	X
Sistema de autenticación	X	X	X	X
Peticiones HTTP (POST, GET)	X	X	X	X
Almacenamiento en cache	X	-	-	X
Manejo de sesiones	X	X	X	X

Herramienta de prueba y depuración de la aplicación	X	-	-	-
Línea de comando para la ejecución de tareas	X	-	X	X
Soporte para el uso de correo electrónico	X	-	X	X
Multi-idioma	X	X	X	X
Extensa y completa documentación	X	X	X	X

Realizado por: Yupangui María, 2019

Luego de haber realizado un estudio comparativo de los frameworks de PHP, se ha determinado que laravel cumple con todas las características ya mencionadas anteriormente lo que lo convierte en un framework adecuado para el desarrollo del sistema web.

1.3.4. Estructura del directorio de carpetas Laravel

Para conocer a profundidad el funcionamiento el framework se procede a realizar un estudio de cada uno de los archivos y directorios generados por Laravel versión 5.6 como se puede observar en la **Figura 1-1**.

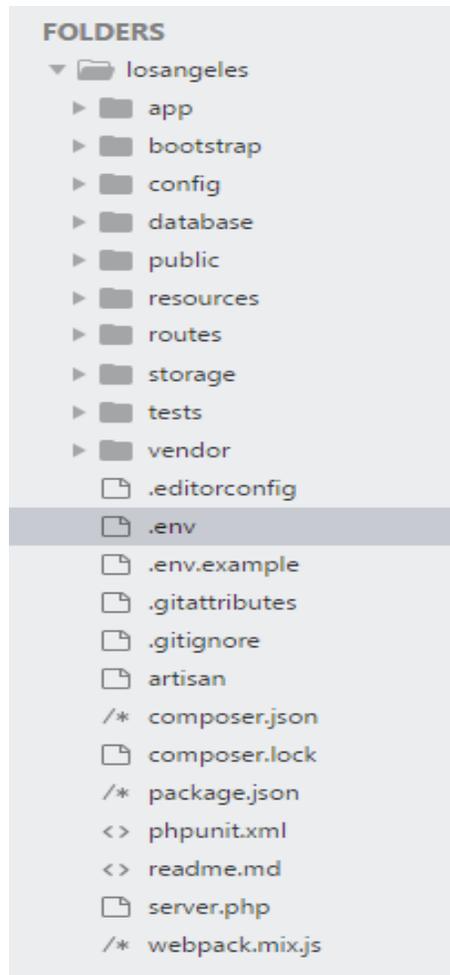


Figura 1-1: Estructura de carpetas y directorios Laravel

Realizado por: Yupangui María, 2019

- **Directorio app:** App es la carpeta principal del proyecto, esta carpeta aloja por defecto todo el código personal como clases, archivos de configuración, etc. Esta carpeta esta a su vez compuesto de varias subcarpetas (Console, Exceptions, Http, Providers, User.php) sin embargo cabe recalcar que la más utilizada es la carpeta Http, la cual contiene los Controllers y Middleware en general esta carpeta contiene toda la lógica.
- **Directorio Bootstrap:** Contiene el framework de Bootstrap y el directorio de caché donde se guarda la caché de la app (López Torralba, 2018, <http://mialtweb.es/estructura-de-laravel/>).
- **Directorio config:** Esta carpeta contiene todos los archivos de configuración del proyecto, pero entre los más usados de este directorio son los siguientes:
 - ✓ app.php: este archivo es usado para definir la zona horaria, idioma, providers, etc.
 - ✓ database.php: este archivo contiene la configuración de la base de datos.

- **Directorio database:** Este directorio contiene todos los archivos relacionados con la base de datos, en la subcarpeta migrations se encuentran todos los archivos de migración y la configuración de datos semillas se lo realiza en la subcarpeta seeds.
- **Directorio public:** Public contiene todo lo relacionado con imágenes, archivos css, js, que posteriormente serán utilizados en la creación de las vistas, además se encuentra el archivo index.php el cual es el punto de entrada de todas las solicitudes.
- **Directorio resources:** En este directorio se encuentran las subcarpetas **js** la cual contiene archivos de JavaScript, la subcarpeta **lang** en la cual se encuentra los idiomas configurados, **view** subcarpeta que contiene todas las vistas con extensión .blade.
- **Directorio routes:** Este directorio contiene subcarpetas las cuales se definirán a continuación:
 - ✓ **api.php:** contiene rutas de RouteServiceProvider.
 - ✓ **channels.php:** es utilizado si el sistema usa eventos, ahí se registra todos los canales de transmisión de eventos que admite el sistema.
 - ✓ **console.php:** aquí se define comandos de consola basados en closure.
 - ✓ **web.php:** en este archivo se define todas las rutas del sistema.
- **Directorio storage:** Este directorio contiene las plantillas .blade compiladas, sesiones, logs, y otros archivos generados por el framework Laravel, en conclusión, este directorio almacena cualquier archivo que Laravel lo puede utilizar como cache.
- **Directorio test y vendor**
 - ✓ **test:** contiene las pruebas automatizadas de la app (López Torralba, 2018, <http://mialtoweb.es/estructura-de-laravel/>).
 - ✓ **vendor:** contiene las dependencias de Laravel (Composer).
- **Archivo .env:** Este archivo se encuentra en la raíz de la aplicación, es utilizada para la configuración de la conexión de la base de datos.

1.4. Servidor web

Un servidor web es un programa que implementa el protocolo HTTP (Mcglone, 2008, p.3) el cual recibe peticiones de los clientes y responde a ellas transfiriendo documentos de tipo hipertexto.

Según (Cases, 2014; citado en Chavarria Neira y Gudiño de la A, 2017, p.43) expresa que los servidores web están alojados en un ordenador que cuenta con conexión a Internet. El web Server, se encuentra a la

espera de que algún navegador le haga alguna petición, como, por ejemplo, acceder a una página web y responde a la petición, enviando código HTML mediante una transferencia de datos en red.

1.4.1. Apache

El servidor web Apache es un servidor Web gratuito desarrollado por el Apache Server Project (Proyecto Servidor Apache) cuyo objetivo es la creación de un servidor web fiable, eficiente y fácilmente extensible con código fuente abierto gratuito (Díaz et al., 2002, p.11).

Apache es hoy por hoy, uno de los servidores web más utilizados a escala mundial, en gran parte se debe a que es Freeware bajo licencia GNU, y en parte también se debe a su robustez y sus múltiples posibilidades (Lechon, 2010, p.5). Para gestionar el servidor apache se utilizará XAMPP.

Xampp es un servidor independiente en base a software libre, con el cual podemos disponer de un servidor propio o simplemente usarlo para hacer pruebas de nuestras páginas web, bases de datos, para desarrollar aplicaciones en PHP, con conexión a base de datos SQL (LAMPP= Linux + Apache + MySQL + PHP + Perl) (Menéndez y Barzanallana, 2018, <https://www.um.es/docencia/barzana/DAWEB/Desarrollo-de-aplicaciones-web-Xampp.html>).

El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente Xampp está disponible para GNU/Linux, Microsoft Windows y MacOS X. (Menéndez y Barzanallana, 2018, <https://www.um.es/docencia/barzana/DAWEB/Desarrollo-de-aplicaciones-web-Xampp.html>).

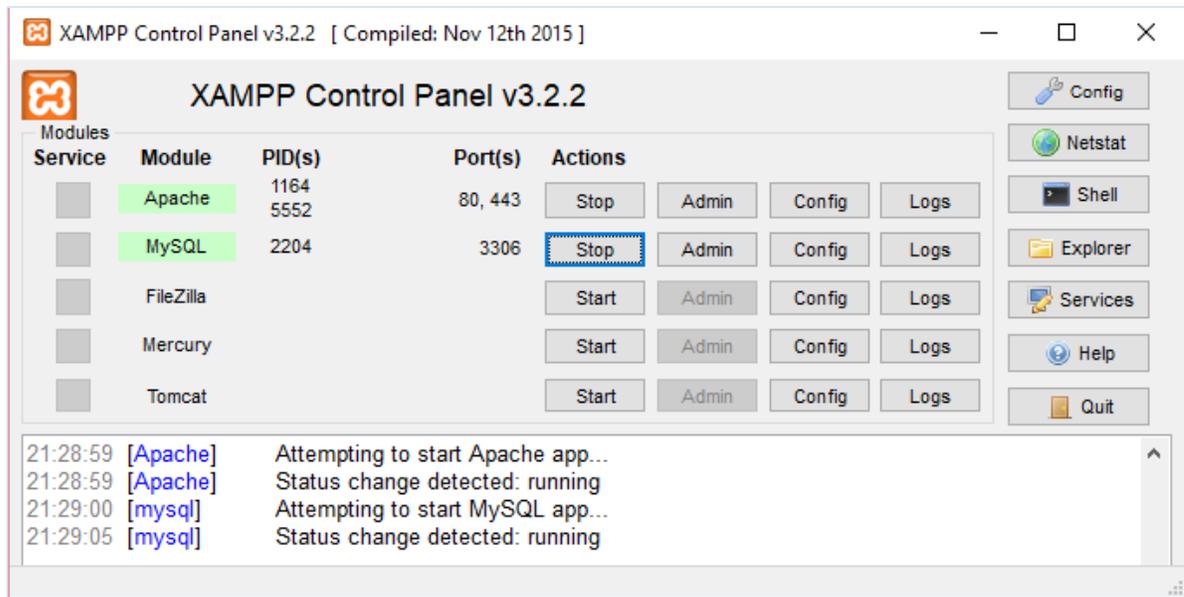


Figura 2-1: Servidor web Apache y MySQL
 Realizado por: Yupangui María, 2019

1.4.1.1. Características del servidor Apache

A continuación, se describe las características del servidor web Apache.

- Es uno de los más utilizados a nivel mundial
- Multiplataforma
- Posee infinidad de paquetes y módulos que nos permiten trabajar con gran cantidad de lenguajes de programación web, así como intérpretes de SQL y otras funciones (Lechon, 2010, p.5).
- Robusto y seguro
- Permite transacciones seguras mediante la utilización de SSL (Secure Socket Layer)
- Permite hosts virtuales

1.5. Arquitectura de software

Arquitectura de un Software es el proceso por el cual se define una solución para los requisitos técnicos y operacionales del mismo. Este proceso define qué componentes forman el software, cómo se relacionan entre ellos, y cómo mediante su interacción llevan a cabo la funcionalidad especificada, cumpliendo con los criterios previamente establecidos; como seguridad, disponibilidad, eficiencia o

usabilidad. (Almeyda Sifuentes, 2019, <http://blog.continental.edu.pe/sistemas-informatica/2013/01/10/arquitectura-de-software-que-es-y-como-funciona/>).

Según Valle y Gutiérrez (2005, pp.3) define la arquitectura de software como un entramado de componentes funcionales que, aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos, de manera que pueden ser utilizados eficazmente dentro de la organización.

Ventajas

Las ventajas que posee el uso de arquitecturas de software según Apiumhub (2017, <https://apiumhub.com/es/tech-blog-barcelona/arquitectura-de-software/>) se describen a continuación:

- ✓ Creamos una base sólida para el proyecto.
- ✓ Conseguiremos que la plataforma creada sea escalable.
- ✓ Aumenta el rendimiento de la plataforma.
- ✓ Reduce considerablemente los costes y evita duplicaciones del código.
- ✓ La arquitectura de software es una manera de garantizarte una buena IT, puesto que el arquitecto debe conservar durante toda la creación, una visión general del producto, de forma que pueda bajar a pequeñas partes del código y saber enlazarlas con el resto. Debe mantener una línea lógica y en ocasiones incluso, dar la cara para asegurar resultados exitosos.
- ✓ Ya que el código es propio, es mucho más visible y se tiene pleno conocimiento sobre él, de forma que será mucho más fácil encontrar problemas y por lo tanto soluciones, en definitiva, tenemos un mantenimiento mucho más eficaz.
- ✓ Este conocimiento, a la par nos permite la implementación de cambios en los sistemas de manera más rápida.
- ✓ Incrementa la calidad de la plataforma.
- ✓ Ayuda en las tareas más complejas.
- ✓ Nos permite hacer la plataforma de manera más rápida.
- ✓ Permite una mayor adaptabilidad. Por ejemplo, a la hora de modificar características técnicas en front end, o implementar motores de reglas. Son tareas más sencillas de adaptar cada una a su debido tiempo, ya que por otro lado el arquitecto de software establece prioridades.
- ✓ Ayuda a la gestión de riesgos reduciendo el porcentaje de fracasos.
- ✓ Reduce los tiempos de creación y de entrega de los proyectos.

Entre las arquitecturas más utilizadas se encuentra:

- Modelo– Vista– Controlador
- Cliente – Servidor
- Distribuida

1.6. Comparativa de arquitecturas

En la actualidad existen una gran variedad de arquitecturas de software, los cuales permiten estructurar de mejor manera el desarrollo de una aplicación. En la **Tabla 2-1** se observa una tabla comparativa de las arquitecturas más utilizadas.

Tabla 2-1: Comparativa de arquitecturas de software

Cuadro comparativo de arquitectura de software más utilizadas			
Característica	Cliente- Servidor	Capas	Modelo – Vista – Controlador
Posibilidad de tener varias vistas para un modelo.	–	–	X
Mantenibilidad	X	X	X
Implementación del código de forma modular	X	X	X
Orientado a objetos	X	X	X
La modificación de las vistas, no afectan al modelo	X	X	X
Separación de la lógica de negocios y la presentación de los datos	X	X	X
Permite que el sistema sea escalable	–	X	X
Permite un mejor manejo de errores	–	X	X
Vistas con información actualizada	X	X	X

Reutilización de código	X	X	X
-------------------------	---	---	---

Realizado por: Yupangui María, 2019

Luego de haber realizado un estudio comparativo de las arquitecturas de software más utilizadas por los desarrolladores, se concluye que la arquitectura más idónea para la elaboración del sistema web de facturación es la arquitectura Modelo – Vista – Controlador (MVC), ya que cumple con todas las características ya mencionadas anteriormente, por lo cual a continuación se da a conocer sus componentes, funcionamiento y ventajas.

1.7. Arquitectura MVC

MVC es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos (Álvarez, 2014, <https://desarrolloweb.com/articulos/que-es-mvc.html>). Esta arquitectura está separada en 3 capas a continuación, se describirá cada una de ellas.

1.7.1. Modelo

Es la representación de la información que maneja la aplicación; el modelo en sí son los datos puros que puestos en contexto del sistema proveen de información al usuario o a la aplicación misma (Rivera, 2008, p.11).

El modelo se puede crear manualmente, pero Laravel usa los comandos Artisan la cual permite la creación de modelos usando líneas de código.

```
php artisan :make:model NombredelModelo
```

Para la creación del modelo y su respectiva migración se puede usar de manera opcional la siguiente línea de código.

```
php artisan:make:model NombredelModelo --migration  
php artisan:make:model NombredelModelo --m
```

El uso de Eloquent permite una implementación simple ya que usa un modelo para interactuar con cada una de las tablas de la base de datos, este modelo permite la inserción, actualización, búsqueda, consulta, etc.

1.7.2. Vista

Es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación Web, la “Vista” es una página HTML con contenido dinámico sobre el cuál el usuario puede realizar operaciones (Rivera, 2008, p.11).

Las vistas en Framework Laravel son archivos de texto el cual se encuentra estructurado con código HTML e incrustado código PHP, de igual manera para la construcción de las vistas hace uso de un sistema de plantillas blade permitiendo al desarrollador la reutilización de código.

Plantillas blade:: Blade nos permite la herencia entre vistas, de esta manera podemos crear una plantilla base con nuestro esqueleto HTML y hacer que las vistas hereden de esta vista (Comas, 2014, <https://ayudalaravel.com/layouts-en-blade/>). Además, poseen una plantilla base o padre del cual se hereda alguna sección denominado layout.

Layouts: El uso de los layouts permite organizar el código de las vistas, sobre todo en elementos que permanecen de forma estática en la aplicación durante alguna petición, por ejemplo: el encabezado o header, pie de página o footer, menús, etc. Generalmente a estas partes se las denomina etiquetas o secciones.

```

<html>
<head>
<title>App Name - @yield('title') </title>
</head>
<body>
@section('sidebar')
    @show
    <div class="container">
        @yield('content')
    </div>
</body>
</html>

```

Figura 3-1: Estructura de un archivo layout
Realizado por: Yupangui María, 2019

Para extender este diseño a una vista blade se debe heredar la plantilla layout, esto se realiza agregando al inicio del archivo de la vista la siguiente línea de código `@extends('Nombre del layout')`, de igual manera para agregar una sección se hará uso `@section('Nombre de la Sección')`, para incluir una directiva se usa `@yield`.

1.7.3. Controlador

Controlador: Es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información necesaria y modificando el Modelo en caso de ser necesario (Rivera, 2008, p.12).

Los controladores al igual que el modelo se pueden crear de manera manual o mediante el uso de los comandos Artisan. La siguiente línea de código crea un controlador con recursos, es decir el controlador ya viene incluido funciones como se puede observar en la figura anterior, esto ayuda a reducir tiempo en la codificación.

php artisan :make:controller NombredelControlador --resource

1.7.4. Funcionamiento de la arquitectura MVC – Laravel

La mayoría de los frameworks utilizan la arquitectura MVC, en este caso el framework Laravel no es la excepción, en la **Figura 8-1** se visualiza el funcionamiento del framework con el patrón MVC.

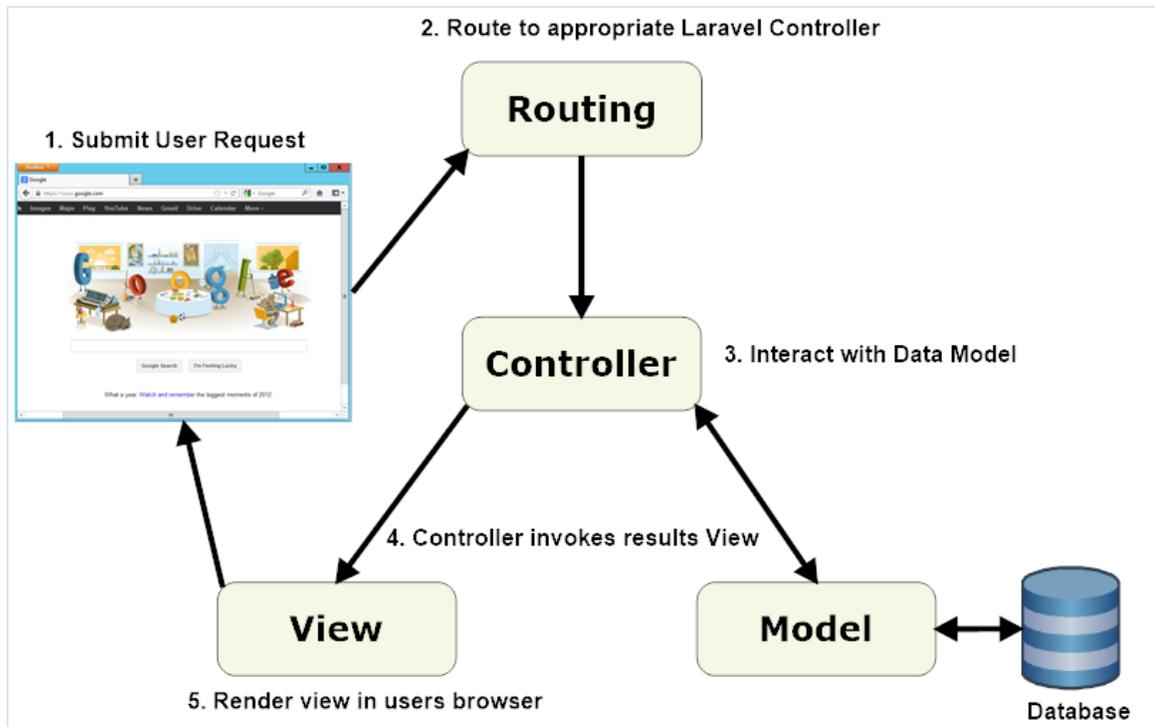


Figura 4-1: Comportamiento de la arquitectura MVC y el framework Laravel

Fuente: (Mozammel, 2013, <https://www.slideshare.net/mozammel8435/finding-laravel-from-a-lost-advanced-beginner-of-java>)

1.7.5. Ventajas

A continuación, se describe las ventajas de la arquitectura MVC.

- ✓ Brinda la facilidad de construir diferentes vistas para un mismo modelo.
- ✓ Proporciona un mecanismo de configuración para componentes complejos mucho más tratable que el puramente basado en eventos (el modelo puede verse como una representación estructurada del estado de la interacción) (Peinado, 2010, p.4).
- ✓ Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas (Rivera, 2008, p.13).
- ✓ Facilita el mantenimiento en las diferentes capas.
- ✓ Permite el escalamiento de la aplicación.

1.8. Base de datos

Una base de datos es una colección de información organizada de tal modo que sea fácilmente accesible, gestionada y actualizada. En una sola vista, las bases de datos pueden ser clasificadas de acuerdo con los tipos de contenido: bibliográfico, de puro texto, numéricas y de imágenes (Rouse, 2015, <https://searchdatacenter.techtarget.com/es/definicion/Base-de-datos>).

En informática, las bases de datos se clasifican de acuerdo a su enfoque organizativo. El enfoque más frecuente es la base de datos relacional, una base de datos tabular en la que los datos se definen de manera que puede ser reorganizada y se accede en un número de maneras diferentes (Rouse, 2015, <https://searchdatacenter.techtarget.com/es/definicion/Base-de-datos>).

1.8.1. MySQL

El sistema de base de datos operacional MySQL es hoy en día uno de los más importantes en lo que hace al diseño y programación de base de datos de tipo relacional. Cuenta con millones de aplicaciones y aparece en el mundo informático como una de las más utilizadas por usuarios del medio. El programa MySQL se usa como servidor a través del cual pueden conectarse múltiples usuarios y utilizarlo al mismo tiempo (Bembibre, 2009, <https://www.definicionabc.com/tecnologia/mysql.php>). Es un sistema de gestión de base de datos relacional, multihilo y multiusuario (Pérez García, 2007, p.13).

1.8.1.1. Características:

Las características del sistema gestor de base de datos se describen a continuación:

- Velocidad
- Facilidad de uso de la herramienta
- Capacidad ya que muchos usuarios pueden realizar peticiones simultáneamente.
- Multiplataforma.
- Multihilo

- Transacciones ACID (Atomic, Consistent Isolated, Durable), que permite construir aplicaciones más seguras haciendo uso de commit, rollback, crash recovery y bloqueo por registro.
- Gran cantidad de tipos de datos.
- Gestión de usuarios y password, manteniendo un muy buen nivel de seguridad en los datos (Acosta Cons et al., 2013: <https://es.slideshare.net/Wario1025/my-sql-26811089>).
- Tiene la capacidad de soportar hasta 32 índices por tabla.
- Permite el uso de los triggers para mejorar las reglas del negocio.
- Usa Information Schema para obtener un fácil acceso a los metadatos.
- Cuenta con transacciones distribuidas para que pueda soportar transacciones en múltiples ambientes de base de datos.
- Utiliza vistas para que la información sensible sea más segura.

1.9. Otras herramientas para el desarrollo de la aplicación

1.9.1. Navegadores web

Un navegador es un software que permite el acceso a Internet, interpretando la información de archivos y sitios web para que éstos puedan ser leídos. La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto con recursos multimedia incrustados (Sánchez Soto y Rodríguez Rivero, 2016, p.9).

1.9.2. Sublime Text

Es un editor de texto ligero, pensado desde un inicio en la velocidad, haciéndolo uno de los editores de texto más rápido y fácil de usar. Además de la velocidad se tiene más de 1000 de plugins adicionales y todas de código abierto, con una comunidad de desarrolladores que día a día contribuyen con nuevos plugins (Villasante, 2014, <https://enboliviacom.wordpress.com/2014/02/21/desarrollo-web-con-sublime-text/>).

1.9.3. Lenguaje de programación

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones, y es utilizado para controlar el comportamiento físico y lógico de una máquina (Flujo, 2018, p.4).

Un lenguaje de programación permite especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural, tal como sucede con el lenguaje Léxico (Flujo, 2018, p.4).

1.9.3.1. PHP

PHP es un lenguaje interpretado del lado del servidor que surge dentro de la corriente denominada código abierto (open source). Se caracteriza por su potencia, versatilidad, robustez y modularidad. Al igual que ocurre con tecnologías similares, los programas son integrados directamente dentro del código HTML (Cobo *et al.*, 2005, p.23).

El lenguaje puede ejecutarse en prácticamente todos los sistemas operativos actuales y en múltiples servidores web. Este también soporta una amplia variedad de bases de datos y cuenta con múltiples librerías para ejecutar procesos comunes (Arce, 2018, p.3).

PHP 7.- El 03 de diciembre 2015 se anuncia el lanzamiento de la nueva versión de PHP, PHP7 incluye numerosas mejoras y nuevas características que a continuación se describe en el siguiente apartado.

1.9.3.2. Características

PHP7, viene integrado nuevas características las cuales son descritas a continuación:

- Soporte: soporta archivos de 64 bits, por lo que funciona sin problemas al sistema operativo Windows.

- Fácil manejo de los errores: hace uso de engine exceptions la cual permite manejar los errores fatales a través de excepciones, de esta manera se logrará manejar los errores fácilmente y sin problemas.
- Rendimiento: permiten la ejecución del doble de solicitudes por segundo en el sistema. Es muy significativa las mejoras en relación al rendimiento (en comparación con PHP 5.6, por ejemplo, que posee casi la mitad de latencia) (Macedo, 2018, <https://www.workana.com/blog/freelanceando/php7-razones-migrar/>).
- Bajos requerimientos en memoria RAM.
- Generador de números aleatorios criptográficamente más seguros (Zend Technologies, 2018, <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/caracteristicas-y-ventajas-de-php7/>).
- Seguridad

1.9.4. JavaScript

JavaScript fue desarrollado por Netscape, a partir del lenguaje Java, el cual sigue una filosofía similar, aunque va más allá. Java es un lenguaje de programación por sí mismo, como lo puedan ser C, Pascal o VisualBasic. Esto quiere decir, que se puede ejecutar un programa Java fuera de un navegador. Pero, repetimos, que la diferencia fundamental es que Java es un lenguaje completo, que puede ser utilizado para crear aplicaciones de todo tipo, mientras que JavaScript sólo “funciona” dentro de una página HTML (Navarrete, 2007, p.1).

Javascript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con Javascript podemos crear diferentes efectos e interactuar con nuestros usuarios. Este lenguaje posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros (Pérez Valdés, 2007, <http://www.maestrosdelweb.com/que-es-javascript/>).

1.9.5. Ajax

Ajax es una forma de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente (en este caso el navegador de los usuarios), y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma (Alegsa, 2010, <http://www.alegsa.com.ar/Dic/ajax.php>).

Las tecnologías que forman Ajax de acuerdo con (Pérez, 2008, p.5) son:

- HTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, sirve para unir las tecnologías anteriormente mencionadas.

1.9.6. Chart.js

Chart.js es una herramienta potente y simple para la creación de gráficos, es una de las librerías de Javascript más usadas para la creación de gráficos (Carita, 2018, <http://gidahatari.com/ih-es/tutorial-de-creacion-de-graficos-con-chartjs>).

La librería te permite mezclar diferentes tipos de gráficos y trazar datos en escalas fecha tiempo, logarítmica, o personalizada con facilidad. La librería también soporta animaciones que pueden ser aplicadas cuando se cambian los datos o se actualizan colores (Shokeen, 2017, <https://code.tutsplus.com/es/tutorials/getting-started-with-chartjs-introduction--cms-28278>).

1.9.7. HTML5

HTML5 es un lenguaje markup (de hecho, las siglas de HTML significan Hyper Text Markup Language) usado para estructurar y presentar el contenido para la web. Es uno de los aspectos fundamentales para el funcionamiento de los sitios, pero no es el primero. Es de hecho la quinta revisión del estándar que fue creado en 1990 (Barbarapvn, 2013, <https://hipertextual.com/archivo/2013/05/entendiendo-html5-guia-para-principiantes/>).

A diferencia de otras versiones de HTML, los cambios en HTML5 comienzan añadiendo semántica y accesibilidad implícitas. Establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos de ellos son técnicamente similares a las etiquetas <div> y , pero tienen un significado semántico, como por ejemplo <nav> (bloque de navegación del sitio web) o <footer>. Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos <audio> y <video> (Aubry, 2011, p.2).

1.9.8. CSS3

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes (Eguíluz, 2008, p.5).

Css3 (Cascading Style Sheets Level 3) llega al mercado con numerosas novedades, que permitirá desarrollar interfaces más dinámicas y llamativas para el usuario. Brinda soporte a la web, eliminando la necesidad de recurrir a algún truco o lenguaje de programación.

1.9.9. Bootstrap

Bootstrap es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una pantalla de escritorio, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “Responsive Design” o diseño adaptativo (Curso, 2016, p.48).

Los diseños creados con Bootstrap son simples, limpios e intuitivos, esto le da agilidad a la hora de cargar y al adaptarse a otros dispositivos. El Framework trae varios elementos con estilos predefinidos fáciles de configurar: Botones, Menús desplegables, Formularios incluyendo todos sus elementos e integración jQuery para ofrecer ventanas y tooltips dinámicos (Curso, 2016, p.48).

1.9.9.1. Ventajas

A continuación, se describe las ventajas del framework Bootstrap.

- **Mobile First:** Todo el framework está desarrollado teniendo en cuenta la filosofía mobile first que es la más recomendable para el desarrollo de sitios web responsive (Sánchez Velamazán, 2015, p.30).
- Utiliza HTML5, CSS3, JQuery, GirHub, etc.
- Compatible con los navegadores
- Amplia comunidad tras el proyecto.
- Sus plantillas son fáciles de adaptarlos.
- Utiliza LESS es un potente preprocesador que funciona con lenguaje de programación y permite el uso de variables, funciones entre otras para acelerar y enriquecer los estilos de los sitios web (Sánchez Velamazán, 2015, p.30).
- Cada vez utiliza más plugins al usuario.
- **Mantenimiento:** Cuenta con el mantenimiento y las actualizaciones del equipo de Twitter, por lo que está en constante mejora (Sánchez Velamazán, 2015, p.30).

El sistema web de facturación se basó en el framework Bootstrap para la creación de sus vistas, ya que como anteriormente se mencionaba, este framework nos ayuda a que las interfaces sean

adaptables a cualquier tipo de dispositivo, eliminando la necesidad de crear un estilo para cada tipo de pantalla.

Para el desarrollo de software se utilizó la metodología que a continuación se encuentra descrita, y cada una de las fases que compone la metodología, obteniendo en cada una de ellas su respectivo resultado.

1.10. Metodología de desarrollo SCRUM

Scrum es un marco de trabajo que nos permite encontrar prácticas emergentes en dominios complejos, como la gestión de proyectos de innovación. No es un proceso completo, y mucho menos, una metodología. En lugar de proporcionar una descripción completa y detallada de cómo deben realizarse las tareas de un proyecto, genera un contexto relacional e iterativo, de inspección y adaptación constante para que los involucrados vayan creando su propio proceso. Esto ocurre debido a que no existen ni mejores ni buenas prácticas en un contexto complejo. Es el equipo de involucrados quien encontrará la mejor manera de resolver sus problemáticas. Este tipo de soluciones serán emergentes (Martin Alaimo, 2015, p.21).

Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo (Schwaber y Sutherland, 2014, p.4).

A continuación, se visualiza de forma esquemática el proceso de desarrollo de la metodología SCRUM.

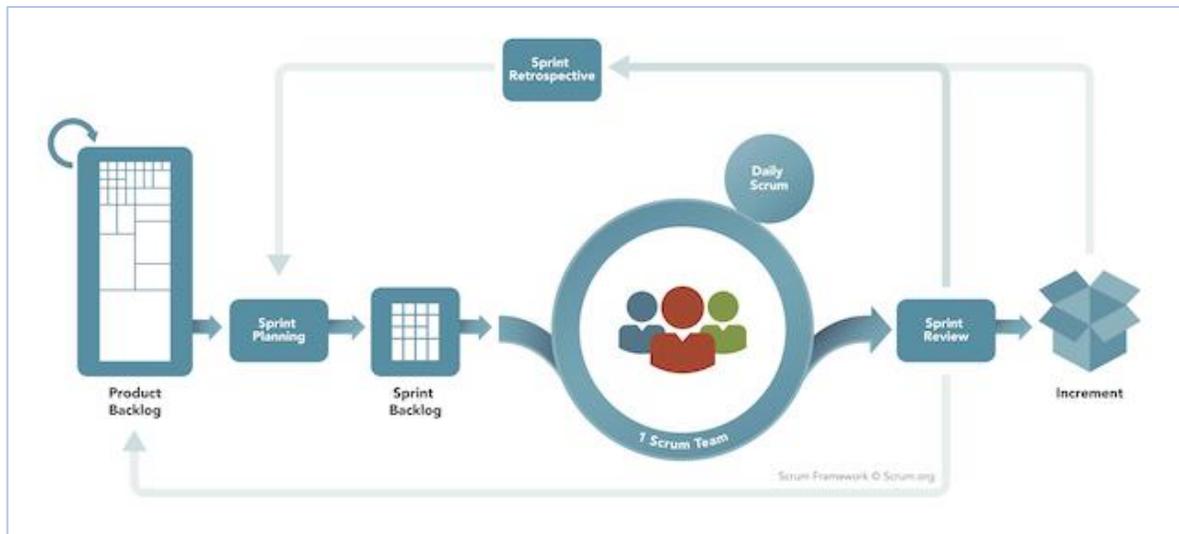


Figura 5-1: Esquema general de la metodología SCRUM

Fuente: (Peralta, 2003, p.3)

1.10.1. Beneficios

A continuación, se describe los beneficios del uso de la metodología SCRUM

- Prioriza las tareas
- Fomentar el trabajo colaborativo
- Adaptación a los cambios
- Equipos autodirigidos
- Fijar tiempos de trabajo para las tareas.
- Fomenta la comunicación entre los miembros
- Divide roles y asigna tareas a cada miembro del equipo.

1.10.2. Roles

En la metodología SCRUM se desarrolla 3 roles los cuales se describen a continuación:

- **Scrum master:** Es el encargado de dirigir el proyecto y de que cada Sprint se cumpla con éxito, también se encarga de que el equipo funcione correctamente en forma armoniosa, funcional y productiva (Yanqui, 2014, p.90).

- **Product Owner:** Es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. El cómo se lleva a cabo esto podría variar ampliamente entre distintas organizaciones, Equipos Scrum e individuos. El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto (Product Backlog) (Schwaber y Sutherland, 2014, p.6).
- **Scrum Team:** Típicamente es un equipo de entre 5 y 10 personas cada una especializada en algún elemento que conforma los objetivos a cumplir (Peralta, 2003, p.9).

Además, cabe recalcar que cada miembro puede intercambiar roles durante el sprint.

1.10.3. Artefactos

Los artefactos utilizados por la metodología se describen a continuación:

Pila del producto (Product Backlog): Es una lista priorizada que define el trabajo que se va a realizar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto. Sin embargo, suelen surgir los más importantes que casi siempre son más que suficientes para un Sprint. (Peralta, 2003, p.4).

La pila del producto no es un documento sino una guía de información de las actividades priorizadas, detalladas y estimadas. Se usa un formato denominado historias de usuario el cual contiene información como:

- Descripción de la funcionalidad
- Prioridad
- Estimación del esfuerzo
- Observaciones criterio de validación
- Número de sprint en el que se realiza
- Módulo del sistema al que pertenece

Pila del sprint. (sprint backlog): Es el punto de entrada de cada Sprint. Es una lista que tiene los ítems de la Product Backlog List que van a ser implementados en el siguiente Sprint (Peralta, 2003, p.5). En otras palabras, es una lista de tareas necesarias para construir la historia de usuario.

Incremento: El incremento tiene como característica el estar completamente terminada y operativa en condiciones de ser entregadas al cliente.

BurnDown chart: Es un gráfico que nos muestra el avance de Sprint y nos ayuda a medir la cantidad de requisitos en la pila del producto.

1.10.4. Reuniones

A continuación, se describe los tipos de reuniones utilizados por SCRUM.

- **Reunión de planificación de Sprint:** Esta reunión se lleva a cabo al inicio de la reunión y tiene una duración de 8 horas, en esta reunión se definen los objetivos de cada proyecto o Sprint, como también se definen las tareas y roles del proyecto (Yanqui, 2014, p.91).
- **Seguimiento del sprint:** Esta reunión se realiza en el mismo lugar y tiene una duración de 15 minutos en la cual se realizan tres preguntas ¿Qué hiciste ayer?, ¿Qué harás hoy?, ¿Qué obstáculo vez en tu camino?, esta reunión es un elemento de presión para que cada integrante haga lo que dijo que va a hacer.
- **Revisión del sprint:** Es una reunión informal que tiene como regla que su preparación no puede tomar más de 2 horas. En ella el equipo presenta lo que ha logrado durante el Sprint. Generalmente toma la forma de una demostración de las nuevas características o la arquitectura (Peralta, 2003, p.8).
- **Retrospectiva:** Solo es una reunión entre los miembros del equipo de trabajo del Sprint es una reunión para obtener feedback y cada miembro del equipo muestra lo que estaba bien o mal, también señalan que puede ser mejorado y como puede implementarse estas mejoras (Pham, 2012; citado en Yanqui, 2014, p.91).

1.11. Herramienta de gestión de pruebas

1.11.1. *TestLink*

Es una herramienta gratuita que permite crear y gestionar casos de pruebas y organizarlos en planes de prueba. Estos planes permiten a los miembros del equipo ejecutar casos de test y registrar los resultados dinámicamente, generar informes, mantener la trazabilidad con los requerimientos, así como priorizar y asignar tareas (Terrera, 2013, <https://testingbaires.com/2013/04/10/testlink-autoaprendizaje-parte-i/>).

De acuerdo con (Peño Sánchez, 2015, p.109), TestLink es una herramienta libre que permite gestionar y crear casos de prueba, organizados en planes de prueba [WEB36]. Esta herramienta se utiliza a través de una interfaz WEB.

1.11.2. *Unidades básicas utilizadas por TestLink*

Los conceptos o unidades básicas con lo que trabaja la herramienta de TestLink se describe a continuación:

- **Test Project:** Es un proyecto, es decir, cualquier aplicativo con un código de aplicación. Para proyectos complejos o de grandes dimensiones es probable que se desgrane como varios proyectos (Ejie, 2013, p.1).
- **Test-suite:** Carpetas que agrupan las pruebas (test-case) con una tipología similar. Es análogo a una especificación, pero para pruebas. Una test-suite se define con un nombre y descripción (Ejie, 2013, p.1).
- **Test-case:** Un caso de prueba es la definición de la prueba a ejecutar y debe pertenecer a una test-suite (Ejie, 2013, p.1). En el test-case se definen los pasos para ejecutar la prueba, además se puede versionar la prueba ya que se pueden crear varias pruebas debido a algún cambio en el requerimiento, también se elige el tipo de ejecución (manual / automática) y se adjunta ficheros u otras evidencias.
- **Test-plan:** Un test-plan agrupa y lista las pruebas que se van a ejecutar. Un test-plan se define mediante un título, descripción, activo (sí o no) y público (sí o no) (Ejie, 2013, p.2). Luego de haber

creado el test-plan se añade el test-case, se asigna las pruebas al usuario y se crean las Build o reléase, se ejecuta el plan es decir se anota los resultados de las pruebas.

1.11.3. Estructura de TestLink

En la **Figura 6-1**, se observa la estructura general de la herramienta TestLink, del cómo se organizan los test y la asignación de test a un test plan y a una Build.

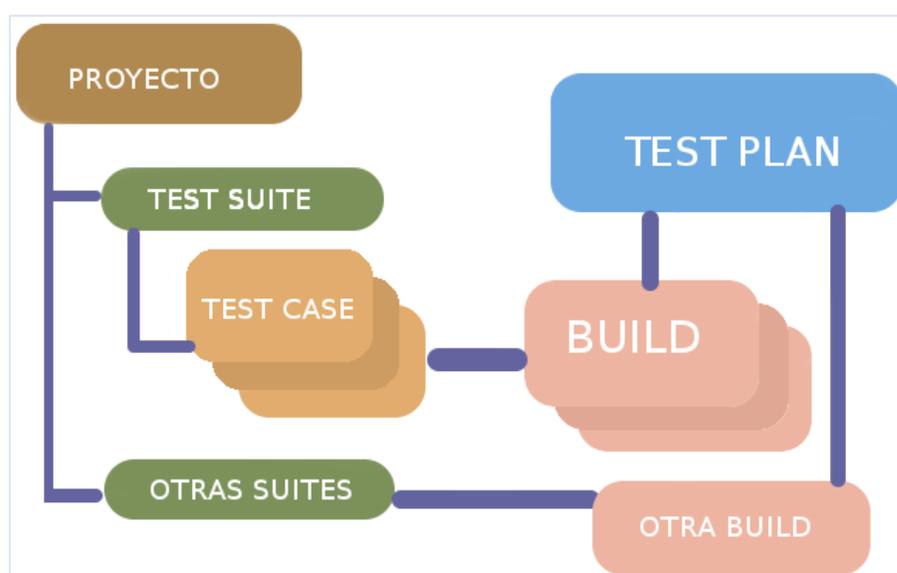


Figura 6-1: Estructura de la herramienta TestLink

Fuente: (Rodríguez, 2013, <https://www.adictosaltrabajo.com/2013/06/05/trabajando-con-test-link/>)

1.11.4. Estructura de TestLink con la metodología SCRUM

Para gestionar el proyecto usando metodologías ágiles, en este caso SCRUM la cual está compuesta por iteraciones y dentro de esta, se encuentra las historias de usuario. La forma de gestionar las pruebas de aceptación por TestLink es la siguiente:

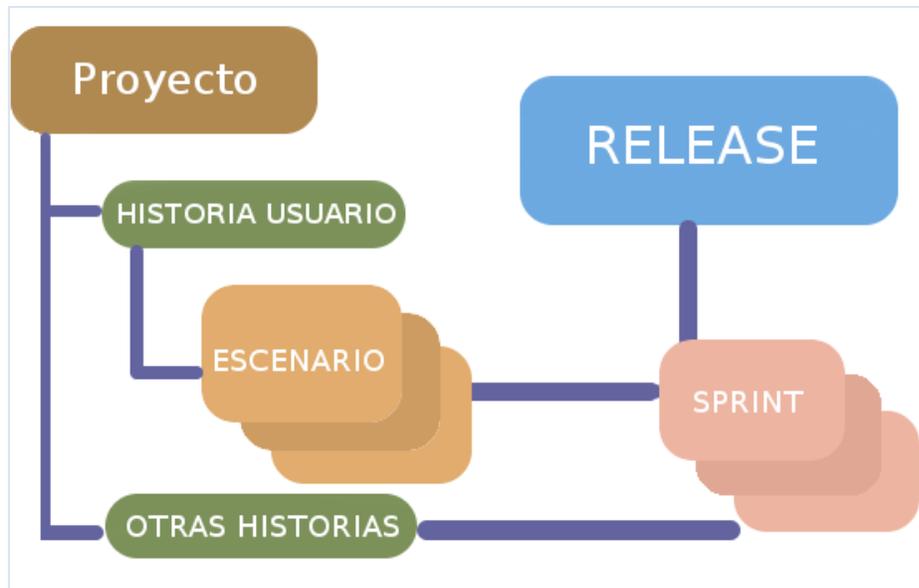


Figura 7-1: Estructura de la herramienta TestLink y la metodología SCRUM

Fuente: (Rodríguez, 2013, <https://www.adictosaltrabajo.com/2013/06/05/trabajando-con-test-link/>)

1.11.5. Beneficios

A continuación, se describe según (Eje, 2013, p.1) los beneficios que ofrece la herramienta.

- Creación de diferentes roles con distintos permisos para los integrantes de un equipo de desarrollo.
- Creación ilimitada de carpetas en forma de árbol (llamadas requeriment-specification) para una mejor organización y agrupamiento de requerimientos.
- Creación ilimitada de carpetas en forma de árbol (llamadas test-suites) para una mejor organización y agrupamiento de los casos de prueba.
- Relacionar los casos de prueba con los requerimientos para indicar el grado de cobertura.
- Versionado de casos de prueba.
- Ejecución de los casos de prueba por versión del software bajo prueba.
- Creación de test-plan para la ejecución y control de las pruebas.
- Integración con herramientas de gestión de incidencias como Mantis.
- Generación de distintos tipos de reportes: listados de pruebas, requerimientos, resultados.

1.12. ISO/IEC 9126

Esta norma Internacional fue publicada en 1992, la cual es usada para la evaluación de la calidad de software, llamado “Information technology-Software product evaluation-Quality characteristics and guidelines for their use”; o también conocido como ISO 9126 (o ISO/IEC 9126) (Borbón Ardilla, 2013, <http://actividadreconocimiento-301569-8.blogspot.com/2013/03/norma-de-evaluacion-isoiec-9126.html>).

1.12.1. Características de ISO/IEC 9126

A continuación, en la **Tabla 2-1** se detalla las características de la norma ISO/IEC 9126.

Tabla 3-1: Características de la norma ISO/IEC 9126

Característica	Descripción	Sub características
Funcionalidad	La capacidad del producto software para proporcionar funciones declaradas e implícitas cuando se usa bajo condiciones.	<ul style="list-style-type: none">• Adecuación• Exactitud• Interoperabilidad• Seguridad de acceso• Cumplimiento funcional
Fiabilidad	La capacidad del producto software para mantener un nivel especificado de prestaciones cuando se usa bajo condiciones especificadas.	<ul style="list-style-type: none">• Madurez• Tolerancia a fallos• Capacidad de recuperación.• Cumplimiento de la fiabilidad
Usabilidad	La capacidad del producto software para ser entendido, aprendido, usado, y ser atractivo para el usuario, cuando se usa bajo condiciones específicas.	<ul style="list-style-type: none">• Capacidad para ser entendido.• Capacidad para ser aprendido.• Capacidad para ser operado.• Capacidad de atracción

		<ul style="list-style-type: none"> • Cumplimiento de la usabilidad.
Eficiencia	Capacidad del producto software para proporcionar prestaciones apropiadas, relativas a la cantidad de recursos usados, bajo condiciones determinadas.	<ul style="list-style-type: none"> • Comportamiento de tiempos • Utilización de recursos • Cumplimiento de la eficiencia
Mantenibilidad	Capacidad del producto software para ser modificado. Las modificaciones podrían incluir correcciones, mejoras o adaptación del software a cambios en el entorno, requisitos y especificaciones funcionales.	<ul style="list-style-type: none"> • Capacidad para ser analizado • Capacidad para ser cambiado • Estabilidad • Capacidad para ser probado • Cumplimiento de la mantenibilidad
Portabilidad	La capacidad del producto software para ser transferido de un entorno a otro	<ul style="list-style-type: none"> • Adaptabilidad • Instalabilidad • Coexistencia • Capacidad para reemplazar • Cumplimiento de la portabilidad

Fuente: (Sarmiento, 2010, pp-6-38)

Realizado por: Yupangui María, 2019

1.12.1.1. *Eficiencia*

La eficiencia es el conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas (Medina Prieto, 2017, <http://unidad4rociomp.blogspot.com/2017/07/46.html>). En la **Figura 8-1** se visualiza las características para medir la eficiencia de un producto software.

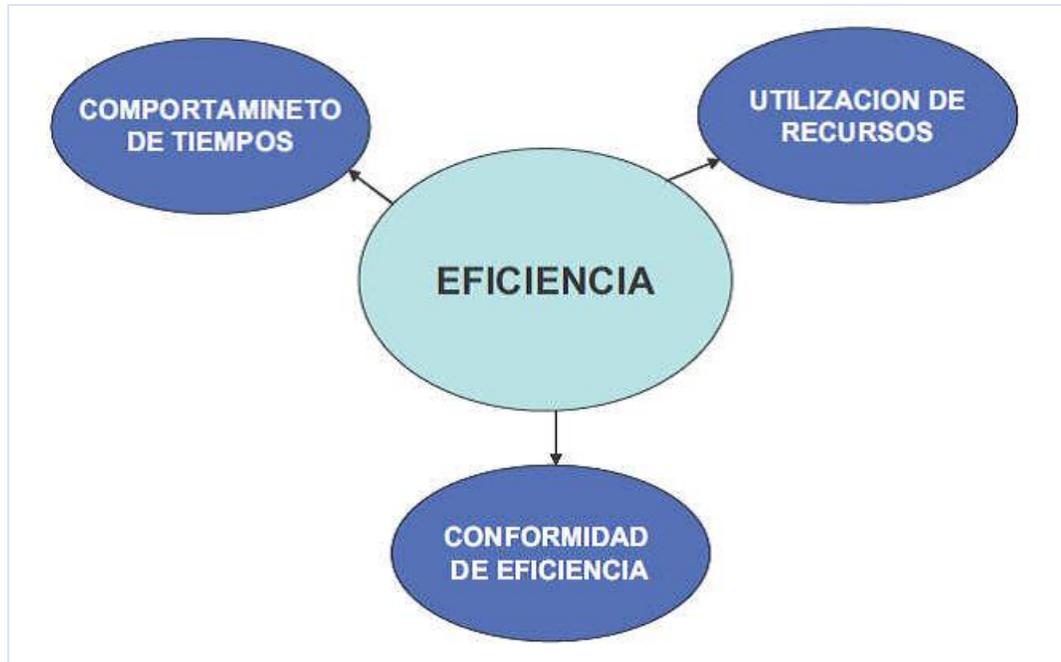


Figura 8-1: Características de la eficiencia

Fuente: (Borbón Ardilla, 2103, <http://actividadreconocimiento-301569-8.blogspot.com/2013/03/norma-de-evaluacion-isoiec-9126.html>)

La eficiencia está compuesta de 3 características que se detallan a continuación:

- **Comportamiento de tiempos:** La capacidad del producto software para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados bajo condiciones determinadas (Sarmiento, 2010, p.24).
- **Utilización de recursos:** la capacidad del producto software para usar las cantidades y tipos de recursos (Sarmiento, 2010, p.25).
- **Cumplimiento de la eficiencia:** la capacidad del producto software para adherirse a normas o convenciones relacionadas con la eficiencia (Sarmiento, 2010, p.26).

En el presente proyecto se va a analizar el comportamiento de tiempos, tomando en consideración que se establecerá escenarios de pruebas con el sistema y sin el sistema. En lo referente a la utilización de recursos no será analizado ya que por la naturaleza del análisis no es pertinente su estudio.

CAPÍTULO II

2. MARCO METODOLÓGICO

En el presente capítulo se describe el tipo de investigación, métodos, técnicas y procesamiento de información, a través de la utilización de la metodología ágil SCRUM, en el cual se evidenciará sus fases en el desarrollo del sistema web de facturación.

2.1. Metodología de la investigación

En este apartado se describe el tipo, métodos y técnicas de investigación utilizadas para un estudio y análisis de los requerimientos del sistema.

2.1.1. *Tipo, métodos y técnicas de investigación*

2.1.1.1. *Tipo de investigación*

El presente trabajo de titulación se utilizará la investigación aplicada, con la finalidad de aplicar los conocimientos teóricos y prácticos adquiridos en nuestra vida estudiantil superior, los mismos que contribuirán para el desarrollo del sistema web, satisfaciendo al cliente con un producto funcional y eficiente.

2.1.1.2. *Métodos de investigación*

Los métodos de investigación que serán de gran ayuda para el desarrollo del sistema web, se describen a continuación:

- **Método análisis:** Con este método se analiza los procesos de facturación de la lavadora y lubricadora “Los Ángeles” con el objetivo de comprender como se lleva a cabo dichos procesos.
- **Método de síntesis:** En este método se utiliza para la construcción del sistema web partir de los elementos percibidos en la etapa de análisis.
- **Método Inductivo:** Mediante este método se obtiene las conclusiones de las pruebas aplicadas, los mismos que permiten comprobar la funcionalidad en los procesos de facturación para el cumplimiento de los objetivos y requerimientos del sistema.
- **Método deductivo:** Este método es usado para emitir las recomendaciones necesarias en el sistema web, luego de haber obtenido las conclusiones en el método inductivo.
- **Método bibliográfico:** Se utilizará este método para fundamentar los conceptos utilizados en el desarrollo del proyecto, la información puede venir de diferentes fuentes como: libros, páginas web, documentos PDF, entre otros.

2.1.1.3. Técnicas de investigación

Para la recolección de información necesaria, para la ejecución del desarrollo del sistema web, se determinó el uso de las siguientes técnicas:

- **Entrevista:** Se realiza la entrevista al gerente propietario de la empresa en la cual se entabla la comunicación con la finalidad de obtener información general del funcionamiento de dicha empresa.
- **Observación:** a través de la observación se pudo percatar de qué manera se realiza los procesos de facturación dentro de la lavadora y lubricadora “Los Ángeles” y como procesa la información para su posterior análisis, además de los cambios que se generen a partir de la implementación del sistema web.

2.2. Metodología para el desarrollo de software SCRUM

La metodología para el desarrollo del sistema web de facturación es SCRUM, ya que es una metodología que permite obtener software de calidad. Además, esta metodología nos permite fomentar una buena comunicación dentro del grupo de desarrollo, que es una característica de las metodologías ágiles a la cual pertenece SCRUM.

También permite realizar cambios de último momento ya que el proyecto se lo planifica y posteriormente se lo divide en iteraciones (Sprint) para su desarrollo, mediante reuniones de entrega del producto se verifica las funcionalidades conjuntamente con el cliente.

2.3. Fase de planificación

En la fase de planificación se realiza un análisis para definir las funcionalidades del sistema, de igual manera se describen los requerimientos del usuario mediante acuerdos que serán establecidos en la reuniones y entrevistas realizadas al gerente.

2.3.1. Reuniones

SCRUM, establece que los miembros del equipo de desarrollo deben mantener reuniones, para establecer los requerimientos funcionales y no funcionales de la aplicación además de planificar las iteraciones o Sprint y entrega de funcionalidades al Product Owner. En la **Tabla 1-2**, se detalla las reuniones que se llevaron a cabo en la fase de planificación.

Tabla 1-2: Reuniones en la fase de planificación

Fecha	Personas	Actividad	Resultado
30/07/2018	Sr. Ernesto Yupangui Vanessa Yupangui	Análisis de las actividades de la empresa en el proceso de facturación de sus productos.	Se estableció los módulos del sistema
10/08/2018	Sr. Ernesto Yupangui Vanessa Yupangui	Definir los requerimientos del sistema.	Listado de los requerimientos del sistema.
15/08/2018	Sr. Ernesto Yupangui Ing. Leticia Vaca Vanessa Yupangui	Revisión de los módulos del sistema, y el listado de requerimientos.	Aprobación de los módulos y requerimientos del sistema.
10/01/2019	Sr. Ernesto Yupangui Vanessa Yupangui	Revisión de avances del sistema web.	Se expuso la interfaz de usuario de la página de inicio.

			Se dio a conocer la interfaz del administrador y empleado. Se mostro la interfaz de la facturación.
10/01/2019	Ing. Leticia Vaca Vanessa Yupangui	Revisión de avances del sistema web.	Se dio a conocer la interfaz del sistema web y la gestión de facturación

Realizado por: Yupangui María. 2019

2.3.2. Análisis de procesos

Luego de las reuniones con el gerente general de la empresa, aplicando la técnica de la observación y entrevista se logró determinar el proceso de facturación el cual se realiza de forma manual.

Dentro del proceso de facturación se presentan varios subprocesos, siendo estos la gestión de artículos, compras y facturación como se puede observar en la **Figura 1-2**.

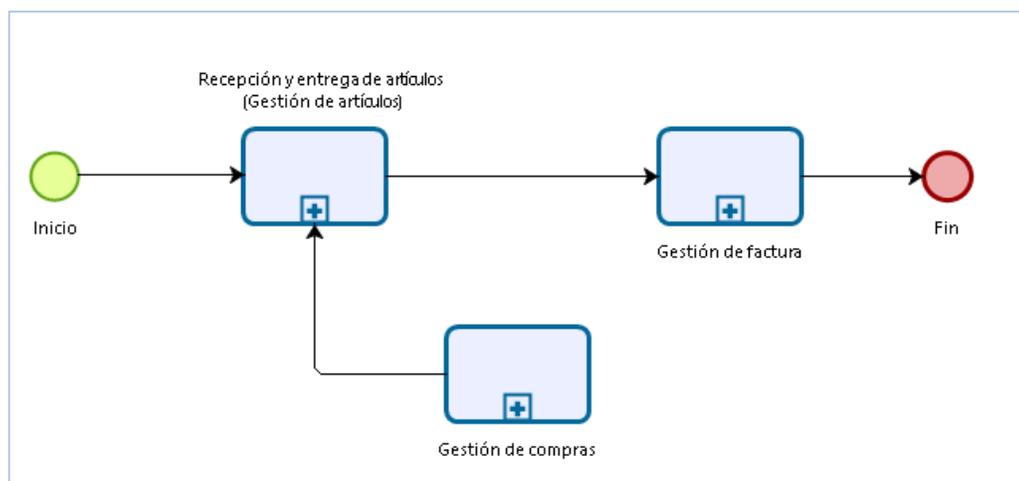


Figura 1-2: Diagrama de procesos general de facturación

Realizado por: Yupangui María, 2019

En la **Figura 2-2** se observa el diagrama de proceso de facturación con sus respectivos subprocesos y los actores involucrados, el administrador del negocio, empleado, proveedor quien suministra los artículos, y finalmente el cliente el cual solicita artículos.

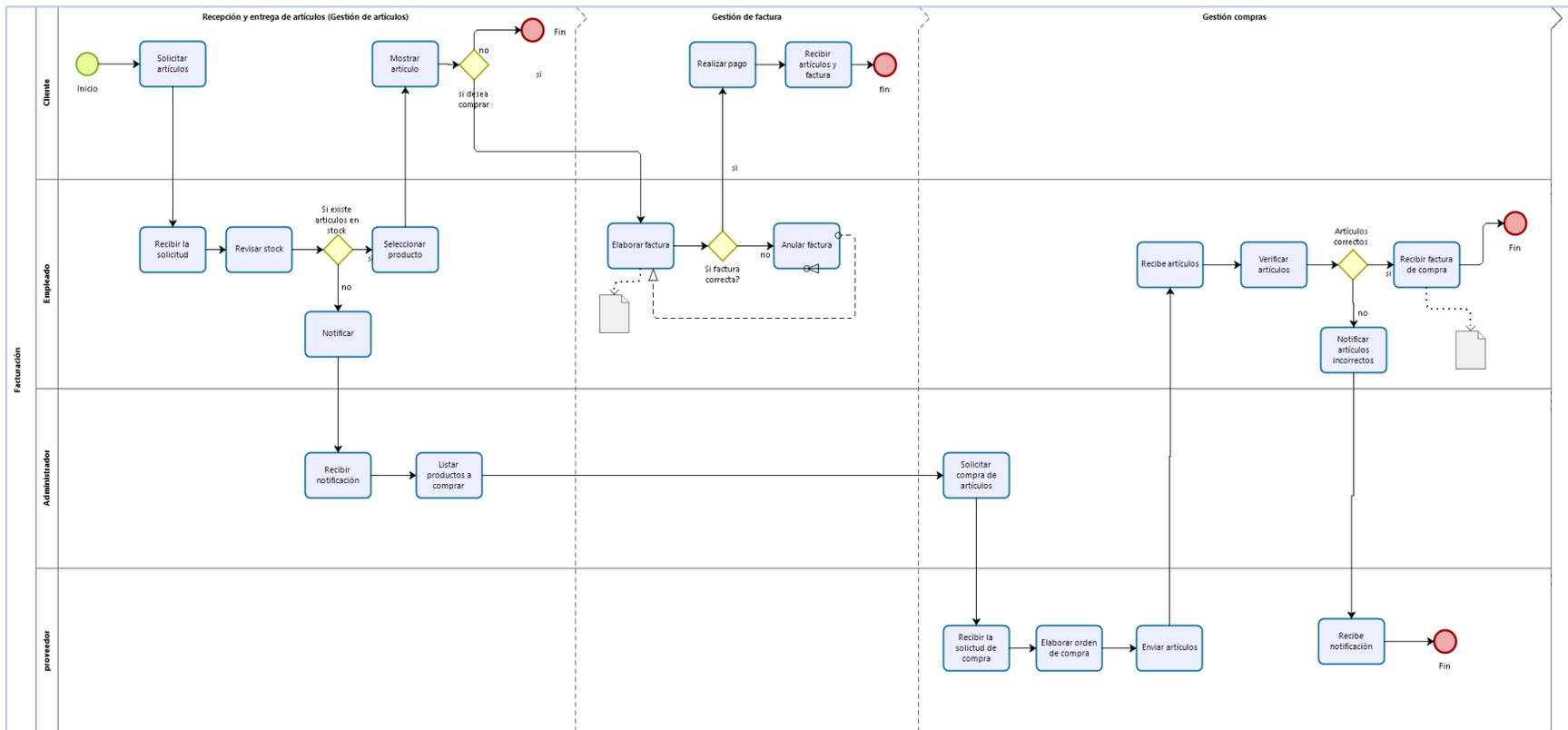


Figura 2-2: Diagrama de procesos detallados de la facturación
 Realizado por: Yupangui María. 2019

2.3.3. Especificación de requerimientos

Este apartado contiene los requerimientos para el desarrollo del sistema web, el mismo que gestionara la información del proceso de facturación realizado por la empresa, de modo que toda la información se encuentre ordenada, centralizada, integra y disponible para el usuario.

2.3.3.1. Propósito

El propósito de este apartado es definir los requerimientos funcionales y no funcionales para el desarrollo del sistema web, el cual permitirá gestionar el proceso de facturación, el sistema será utilizado por el gerente general de la empresa y el vendedor.

2.3.3.2. Alcance

Realizar los respectivos CRUD (Create, Read, Update, Delete) de la información que maneja la empresa como son: categorías, artículos, compras, proveedores, ventas, clientes, usuarios, además el sistema permitirá al usuario gestionar la información que posteriormente serán utilizados en las facturas. El sistema proporcionara reportes para la gestión de compras y ventas, todo esto distribuido en una aplicación web.

2.3.3.3. Restricciones

- El sistema se diseña según el modelo MVC (Modelo-Vista-Controlador)
- Interfaz para ser usada mediante internet
- Uso de un dominio
- Servidor local que contenga el sistema conectado a la red.
- Tecnologías a utilizar: framework Laravel, Ajax, JavaScript, MySQL, PHP

2.3.3.4. *Requerimientos funcionales*

Mediante las reuniones y la información recolectada se realizó un análisis de las funcionalidades y requerimientos del sistema, en base a las actividades que realiza la empresa se determinó 8 requerimientos funcionales los cuales se detallan a continuación:

Sistema web

3. Gestionar la información de los artículos.
4. Registro de compras de la empresa.
5. Gestionar información de los proveedores de la empresa.
6. Registro de ventas
7. Gestionar la información de los clientes.
8. Generar una factura en formato PDF.
9. Gestionar la información de usuarios.
10. Generar reportes de compras y ventas en formato PDF.

Para desarrollar el sistema web para automatizar el proceso de facturación de la Lavadora y Lubricadora “Los Ángeles”, se lo dividió en 7 módulos los cuales se encuentran detallados a continuación:

Módulo usuario

- ✓ Gestión de la información de la empresa: Esta gestión permite visualizar la información de contactos, servicios y productos que ofrece la empresa.

Módulo de autenticación

- ✓ Inicio de sesión (Administrador/Empleado): Este módulo permitirá el ingreso al sistema a través de sus credenciales (nombre de usuario y contraseña).

Módulo de inventario

- ✓ Gestión de las categorías: Esta gestión permitirá el registro, modificación, eliminación y búsqueda de una categoría.

- ✓ Gestión de los artículos. Esta gestión permitirá el registro, modificación, eliminación y búsqueda de un artículo.

Módulo de compras

- ✓ Gestión de compras: La gestión de compras permitirá al usuario registrar las compras de artículos que realice la empresa.
- ✓ Gestión de proveedores: Esta gestión permite registrar, modificar, eliminar y buscar un proveedor de la empresa.

Módulo de compras

- ✓ Gestión de ventas: Mediante esta gestión se puede realizar el registro o anulación de una venta, además de generar su respectiva factura.
- ✓ Gestión de clientes: La gestión de clientes permitirá el registro, modificación y búsqueda de un cliente de la empresa.

Módulo de promociones

- ✓ Gestión de promociones: Esta gestión permitirá al administrador del negocio publicar en la página de inicio las promociones que ofrece la empresa.

Módulo de acceso:

- ✓ Gestión de usuarios: La gestión de usuarios permitirá al administrador registrar nuevos usuarios (empleado o administrador) en el sistema.
- ✓ Gestión de roles: Esta gestión permite visualizar los roles creados para el sistema.

Módulo de reportes

- ✓ Reporte de todas las compras realizadas: Este reporte permite visualizar al administrador todas las compras registradas en el sistema.
- ✓ Reporte de todas las ventas realizadas por la empresa: Este reporte permite visualizar al administrador todas las ventas registradas en el sistema.
- ✓ Reporte de compras en un intervalo de fechas: Este reporte permite visualizar al administrador todas las compras que se encuentren dentro de un rango de fechas especificadas.
- ✓ Reporte de ventas en un intervalo de fechas: Este reporte permite visualizar todas las ventas que se encuentren dentro de un rango de fechas especificadas por el administrador de la empresa.

- ✓ Reporte de compras y ventas de los últimos meses mediante un gráfico estadístico: Este reporte permitirá visualizar dos gráficos, el primero se visualiza el gráfico de compras de los 4 últimos meses y el segundo se observa el gráfico de ventas de los 4 últimos meses. Estos gráficos son dinámicos ya que se actualizará según los datos ingresados en la gestión de compras y ventas.

2.3.3.5. Requerimientos no funcionales

Para brindar al cliente una aplicación web de calidad se define los requerimientos no funcionales del sistema que a continuación, se describe.

- ✓ **Seguridad:** Se brindará seguridad al sistema mediante el módulo de autenticación (usuario, contraseña) el cual permitirá o denegará el acceso a la misma.
- ✓ **Usabilidad:** Mediante el diseño de interfaces amigables e intuitivas, de esta manera facilitará el uso del software por el administrador y empleado.
- ✓ **Mantenibilidad:** El sistema permitirá el desarrollo de nuevos requerimientos de una manera fácil ya que es desarrollado bajo la arquitectura MVC.
- ✓ **Disponibilidad:** El sistema estará disponible las 24 horas del día.

2.3.4. Planificación

La información de los requerimientos definidos en el plan de trabajo, se los visualiza en la **Figura 3-2** en un Diagrama Gantt.

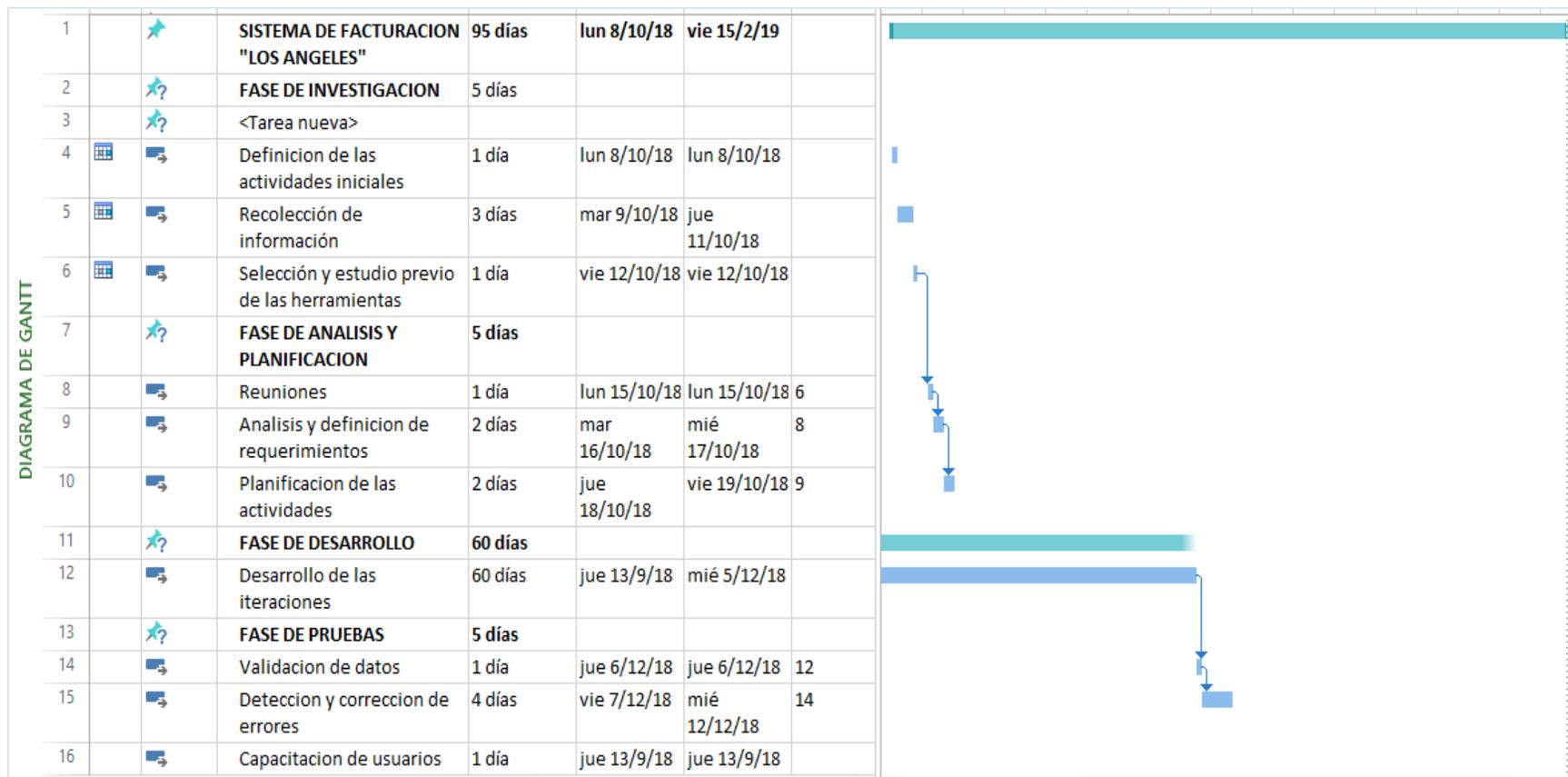


Figura 3-2: Diagrama de Gantt

Realizado por: Yupangui María, 2019

2.3.5. Roles y personas

En el desarrollo del sistema web se involucran a varias personas, a continuación, se detalla a cada una de ellas y sus funciones a desempeñar.

Tabla 2-2: Roles y personas

Persona	Rol
Sr. Ernesto Yupangui	Product Owner
Ing. Alejandra Oñate	Scrum Master
Srta. María Yupangui	Team Development

Realizado por: Yupangui María. 2019

2.3.6. Tipos y roles de usuarios

En el desarrollo del sistema se definió dos tipos de roles de usuarios los cuales se detallan en la **Tabla 3-2.**

Tabla 3-2: Tipos y roles de usuarios

Tipo de usuario	Rol
Administrador	Es la persona que tiene el control total del sistema y puede acceder a todas las funcionalidades.
Empleado	Es el encargado de registrar compras y ventas del negocio además de generar una factura.

Realizado por: Yupangui María. 2019

2.3.7. Pila de producto

La pila de producto o Product Backlog es el conjunto requerimientos que el cliente determina, del sistema web a desarrollar. A través de la lista de requerimientos se puede priorizar y estimar, estos requisitos se los realizara en cada iteración hasta obtener el sistema requerido.

Por medio de reuniones y entrevistas al cliente se obtuvo los requerimientos técnicos y requerimientos funcionales para la aplicación. Para estimar se usó la técnica “talla de la camiseta”, proyectando de esta manera el tiempo aproximado en el desarrollo del sistema. A través de esta técnica también se puede obtener y comparar el tiempo estimado y tiempo real al finalizar el desarrollo. En la siguiente **Tabla 4-2** se puede visualizar la técnica “talla de la camiseta” con sus respectivas equivalencias.

Tabla 4-2: Talla de la camiseta

Talla de la Camiseta		
1/8 Iteración	S	3 puntos
1/4 Iteración	M	5 puntos
1/2 iteración	L	10 puntos
1 iteración	XL	20 puntos

Realizado por: Yupangui María, 2019

Cada punto estimado tiene una equivalencia de 2 horas, por lo tanto, la equivalencia de cada uno de los puntos se encuentra descrita a continuación.

- ✓ 2 puntos equivalen a 4 horas de trabajo
- ✓ 3 puntos equivalen a 6 horas de trabajo
- ✓ 5 puntos equivalen a 10 horas de trabajo
- ✓ 10 puntos equivalen a 20 horas de trabajo
- ✓ 20 puntos equivalen a 40 horas de trabajo

Consecuentemente se ha establecido un horario de trabajo de lunes a viernes de 8 horas diarias de trabajo.

En cuanto a los requerimientos, se encuentran detallados en la **Tabla 5-2**, los cuales poseen un identificador, donde las historias de usuario se encuentran representadas por HU y las historias técnicas con HT, en la descripción se detalla su funcionalidad y los puntos de esfuerzo estimados.

Tabla 5-2: Product Backlog

ID	Descripción	Esfuerzo (Puntos estimados)
HT-01	Como desarrollador deseo recolectar la información necesaria para el desarrollo del sistema.	8
HT-02	Como desarrollador deseo instalar las herramientas y tecnologías necesarias para el desarrollo del sistema.	4
HT-03	Como desarrollador deseo conocer los requerimientos del sistema	8
HT-04	Como desarrollador deseo diseñar la arquitectura del sistema	4
HT-05	Como desarrollador deseo definir un estándar de codificación	4
HT-06	Como desarrollador deseo diseñar la base de datos	8
HT-07	Como desarrollador deseo diseñar la interfaz de usuario	4
HU-01	Como usuario, deseo visualizar una página principal	8
HU-02	Como administrador, empleado deseo ingresar al sistema utilizando mis credenciales (usuario y contraseña)	6
HU-03	Como administrador, empleado, deseo tener una interfaz principal donde se observen las opciones que se puedan realizar	6
HU-04	Como administrador, deseo registrar una categoría	6
HU-05	Como administrador, deseo actualizar una categoría	6
HU-06	Como administrador, deseo eliminar una categoría	4
HU-07	Como administrador, deseo buscar una categoría por nombre o descripción	4
HU-08	Como administrador, deseo listar las categorías	4
HU-09	Como administrador, deseo ingresar un nuevo artículo	4
HU-10	Como administrador, deseo actualizar un artículo	4
HU-11	Como administrador, deseo eliminar un artículo	4
HU-12	Como administrador, deseo buscar un artículo por nombre o código.	4
HU-13	Como administrador, deseo listar los artículos	4
HU-14	Como administrador, deseo obtener un reporte de todos los artículos en inventario en formato PDF	8
HU-15	Como administrador, empleado, deseo registrar una compra	8
HU-16	Como administrador, empleado, deseo listar los registros de compras	4

HU-17	Como administrador, empleado, deseo visualizar el detalle de una compra	4
HU-18	Como administrador, empleado, deseo anular una compra	4
HU-19	Como administrador, empleado, deseo buscar una compra por número de comprobante, tipo de comprobante.	4
HU-20	Como administrador, empleado, deseo registrar un nuevo proveedor	4
HU-21	Como administrador, empleado, deseo actualizar los datos de un proveedor	6
HU-22	Como administrador, empleado, deseo buscar un proveedor por nombre o número de documento.	6
HU-23	Como administrador, empleado, deseo eliminar un proveedor	4
HU-24	Como administrador, empleado, deseo listar todos los proveedores.	4
HU-25	Como administrador, empleado, deseo registrar una venta	12
HU-26	Como administrador, empleado, deseo listar las ventas	4
HU-27	Como administrador, empleado, deseo visualizar el detalle de una venta	4
HU-28	Como administrador, empleado, deseo anular una venta	4
HU-29	Como administrador, empleado, deseo generar una factura en formato PDF	8
HU-30	Como administrador, empleado, deseo buscar una venta por número de comprobante, tipo de comprobante.	4
HU-31	Como administrador, empleado, deseo, registrar un cliente	4
HU-32	Como administrador, empleado, deseo actualizar los datos de un cliente	4
HU-33	Como administrador, empleado, deseo listar todos los clientes	4
HU-34	Como administrador, empleado, deseo buscar un cliente por nombre o número de documento.	4
HU-35	Como administrador, deseo registrar una promoción.	4
HU-36	Como administrador, deseo eliminar una promoción	4
HU-37	Como administrador, deseo actualizar una promoción	4
HU-38	Como administrador, deseo listar las promociones.	4
HU-39	Como administrador, deseo buscar una promoción por nombre o descripción.	4
HU-40	Como administrador, deseo listar los roles existentes	4
HU-41	Como administrador, deseo buscar un rol por nombre o descripción	4
HU-42	Como administrador, deseo registrar un nuevo usuario en el sistema	4

HU-43	Como administrador, deseo listar todos los usuarios del sistema	4
HU-44	Como administrador, deseo actualizar los datos de un usuario	4
HU-45	Como administrador, deseo eliminar un usuario del sistema	4
HU-46	Como administrador, deseo buscar un usuario por nombre o número de documento.	4
HU-47	Como administrador, deseo visualizar un reporte de todas las compras	6
HU-48	Como administrador, deseo visualizar un reporte de todas las ventas	6
HU-49	Como administrador, deseo visualizar un reporte de compras en un intervalo de fechas	8
HU-50	Como administrador, deseo visualizar un reporte de ventas en un intervalo de fechas.	8
HU-51	Como administrador, deseo visualizar un gráfico estadístico de compras y ventas de los últimos meses.	8
HU-52	Como administrador, empleado, deseo visualizar un panel de notificaciones mostrando el número de compras y ventas registradas desde su ultimo ingreso al sistema.	4
HT-08	Como desarrollador, deseo realizar la detección y corrección de errores	20
HT-09	Como desarrollador, deseo analizar los resultados	16
HT-10	Como desarrollador, deseo determinar las conclusiones y recomendaciones	4
HT-11	Como desarrollador, deseo realizar el manual de usuario	4
HT-12	Como desarrollador, deseo realizar el documento del trabajo de titulación	12
HU-53	Como desarrollador, deseo dar capacitación a los usuarios	4

Realizado por: Yupangui María, 2019

Finalmente se obtuvo 53 historias de usuario (HU) y 12 historias técnicas (HT), las cuales serán asignadas a los Sprint.

2.3.8. *Análisis económico*

Para la ejecución del proyecto la mayoría de las herramientas son de licencia libre, por lo cual existe un costo mínimo para el desarrollo del sistema.

A continuación, en la **Tabla 6-2**, se detalla los valores presupuestados del hardware a utilizar, y en la **Tabla 7-2** se detalla el presupuesto de software.

Tabla 6-2: Presupuesto de hardware

Equipo	Cantidad	Precio unitario	Precio total
Laptop DELL Windows 10 Pro Procesador: Intel(R)Core (TM) i5-M450 CPU @ 2,4GHZ Memoria RAM: 4.00GB Disco Duro: 1 TB	1	\$700,00	\$700,00
Disco duro externo 1 TB Toshiba Basic	1	\$60,00	\$60,00
Impresora Epson L210	1	\$200,00	\$200,00
Total			\$960,00

Realizado por: Yupangui María, 2019

Tabla 7-2: Presupuesto de software

Software	Cantidad	Precio unitario	Precio total
Microsoft Windows 10 Pro	1	\$199,00	\$199,00
Microsoft Office 2016	1	\$149,00	\$149,00
Sublime Text 3	1	\$0,00	\$0,00
MySQL	1	\$0,00	\$0,00
PhpMyAdmin	1	\$0,00	\$0,00
StarUML	1	\$0,00	\$0,00
Servidor XAMPP	1	\$0,00	\$0,00
Framework Laravel 5.6	1	\$0,00	\$0,00
Gestor de dependencias Composer	1	\$0,00	\$0,00
Framework Bootstrap	1	\$0,00	\$0,00
Total			\$348,00

Realizado por: Yupangui María, 2019

Además de los recursos hardware y software, en la **Tabla 8-2** se detalló el presupuesto de materiales adicionales que intervinieron en el desarrollo del software.

Tabla 8-2: Presupuesto materiales adicionales

Materiales	Precio unitario (mensual)	Precio total (4 meses)
Suministros de oficina	\$10,00	\$40,00
Transporte y viáticos	\$50,00	\$200,00
Servicio de internet	\$20,00	\$80,00
Total		\$320,00

Realizado por: Yupangui María, 2019

Para el presupuesto de la mano de obra, se tomó en consideración el sueldo básico de un programador junior según la tabla de salarios mínimos sectoriales del Ministerio de Trabajo, en la cual indica el valor de \$419,81 por 8 horas de trabajo diario. En la **Tabla 9-2** se especifica los valores por cuatro meses de trabajo en el desarrollo del software.

Tabla 9-2: Presupuesto mano de obra

Número de desarrolladores	Actividad	Sueldo	Sueldo total (4 meses)
1	Desarrollador de software	\$419,81	\$1.645,12
Total			\$1.645,12

Realizado por: Yupangui María, 2019

Nota: No se emitirá ningún costo en la mano de obra, ya que es contribución de la tesista a la empresa.

Resumen de presupuesto del proyecto

En la **Tabla 10-2** se realiza un resumen del presupuesto en el desarrollo del sistema web con sus respectivos valores.

Tabla 10-2: Resumen de presupuesto del proyecto

Nombre del recurso	Presupuesto
Presupuesto hardware	\$960,00
Presupuesto software	\$348,00
Presupuesto materiales adicionales	\$320,00
Total	\$1.628,00

Realizado por: Yupangui María, 2019

2.4. Fase de diseño

En esta fase se detalla las actividades realizadas para empezar con el desarrollo del sistema, entre las actividades se encuentra el diseño de la arquitectura del sistema, diseño de pantallas, diseño de la base de datos, definición del estándar de codificación y los diagramas UML para reflejar los procesos.

2.4.3. Diagrama de casos de uso

Los diagramas de caso de uso permiten hacer una representación gráfica entre los usuarios y los procesos que se encuentran dentro del sistema, la siguiente figura muestra la relación del administrador y empleado con el sistema, además de la forma, tipo y orden en como los elementos interactúan. En la **Figura 4-2** se describe las funcionalidades dentro del sistema, además se especifica los roles y acciones que pueden realizar cada uno de los usuarios. Estos dos tipos de usuarios deben iniciar sesión para acceder a sus funciones, el administrador del negocio tiene la posibilidad de acceder a todas las opciones implementadas en el sistema, el empleado tiene la posibilidad de registrar compras y ventas.



Figura 4-2: Diagrama caso de uso administrador, empleado.
 Realizado por: Yupangui María, 2019

2.4.4. Diagrama de clases

Para el correcto desarrollo y funcionamiento del sistema se realizó el diagrama de clases la cual se puede visualizar en la **Figura 5-2** con un total de 14 clases con sus respectivos atributos y métodos.

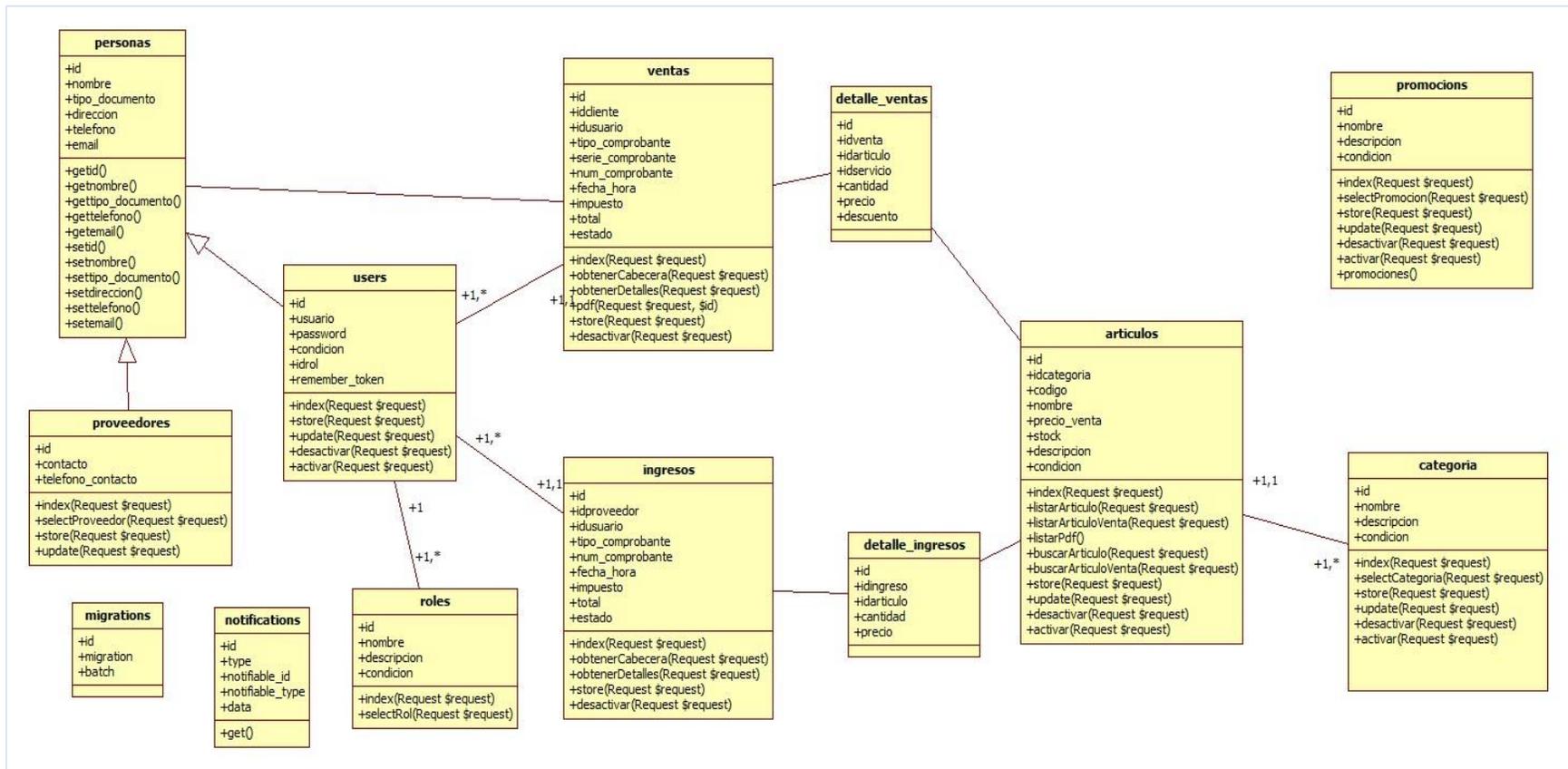


Figura 5-2: Diagrama de clases
 Realizado por: Yupangui María, 2019

2.4.5. Diagrama de componentes

En **Figura 6-2**, se visualiza el diagrama de componentes la cual proporciona una visión física de la construcción del sistema. Muestra la organización de los componentes software, sus interfaces y las dependencias entre ellos, el diseño se lo realizó en base a la arquitectura del framework Laravel.

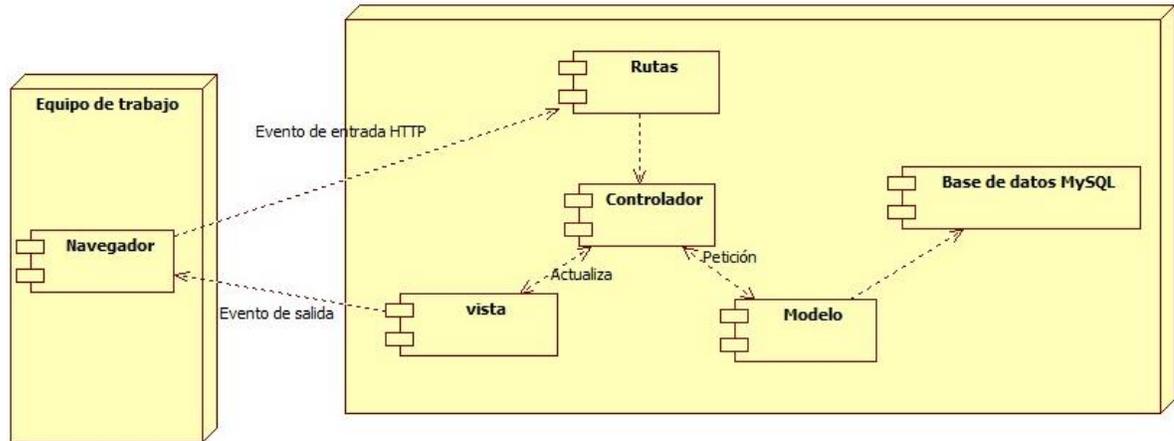


Figura 6-2: Arquitectura del sistema

Realizado por: Yupangui María, 2019

En la figura se puede observar que el navegador ingresa a una ruta, la ruta define el controlador y método solicitado, el controlador se comunicará con el modelo realizando la petición y el modelo accederá a la base de datos devolviendo los datos solicitados por el controlador para que este actualice la vista y presente los datos.

2.4.6. Recursos necesarios

Para el desarrollo del sistema web se utilizó los siguientes recursos tanto hardware y software los cuales se detallan a continuación.

2.4.6.1. Recursos hardware

Los recursos hardware que se ha utilizado para el desarrollo del sistema web de facturación se encuentra descrita en la **Tabla 11-2** con sus respectivas características.

Tabla 11-2: Recursos hardware

Equipo	Descripción	Estado
Laptop	Laptop DELL Procesador: Intel(R)Core (TM) i5-M450 CPU @ 2,4GHZ Memoria RAM: 4.00GB Disco Duro: 1 TB.	Disponible
Impresora Epson L210	Multifuncional Resolución: Hasta 5760 x 1440 dpi de resolución. Velocidad de Impresión: Máxima: 27 ppm en texto negro y 15 ppm en texto a color Normal: 6,0 ISO ppm en negro y 3,0 ISO ppm a color.	Disponible
Disco duro externo	1 TB Toshiba Basic.	Disponible

Realizado por: Yupangui María, 2019

2.4.6.2. Recursos software

Para el desarrollo del sistema web de facturación se requirió de lo siguiente, lo cual se encuentra detallado en la **Tabla 12-2**.

Tabla 12-2: Recursos software

Nombre del software	Funcionalidad
Sublime Text	Editor de texto, el cual es usado para la codificación del sistema web.
Gestor de dependencias Composer	Instalación de librerías para el desarrollo del sistema
Laravel	Framework que ayudara con el desarrollo del sistema.
Bootstrap	Framework para el diseño de interfaces
Windows 10	Sistema operativo
MySQL	Servidor de base de datos el cual almacenara la información.
phpMyAdmin	Administrador de base de datos MySQL
XAMMP	Servidor donde se alojará el sistema
StarUML	Diagramas de procesos del negocio

Microsoft Office	Gestión de documentos
------------------	-----------------------

Realizado por: Yupangui María, 2019

2.4.7. Estándar de codificación

Se estableció un estándar de codificación para las clases, métodos y variables necesarias en el desarrollo del sistema web, con el objetivo de establecer un código fuente uniforme en todo el proyecto y que este facilite el desarrollo en el equipo de trabajo y ayude en el mantenimiento del mismo.

A continuación, en la **Tabla 13-2** se describe los nombres de los objetos, clases, métodos y variables usados en la aplicación.

Tabla 13-2: Estándar de codificación

Componente	Nombre	Descripción
Clases	<NombreClase> <Articulo>	El nombre de la clase se definirá, haciendo referencia de lo que se trata, la primera letra de cada palabra será mayúscula.
Métodos	<nombreMetodo> < buscarArticuloVenta>	El nombre del método hará referencia al proceso que cumplirá, la primera palabra comenzara en minúscula, si el nombre del método se compone de varias palabras su primera letra será mayúscula.
Variables	<variable> <\$pdf>	Las variables se nombrarán de acuerdo a la función a desempeñar y será escrito toda la palabra en minúscula.
Paquetes	<paquete.paquete> <composer.json>	los nombres de los paquetes se escribirán toda la palabra en minúscula, si se compone de varias palabras todas empezaran en minúscula separadas por un punto.

Realizado por: Yupangui María, 2019

2.4.8. Diseño de la interfaz de usuario

Para el diseño de la interfaz, se realizó una reunión con el Product Owner el cual manifestó que el sistema web debe ser amigable, intuitiva y fácil de manejar.

Una vez diseñada las pantallas y realizando sus respectivas pruebas de aceptación, se obtuvo las siguientes interfaces.

2.4.8.1. *Página de inicio del sistema web*

La página de inicio cuenta con un menú el cual permitirá a los usuarios conocer los productos, servicios, promociones, contactos, inicio de sesión e información de la empresa.

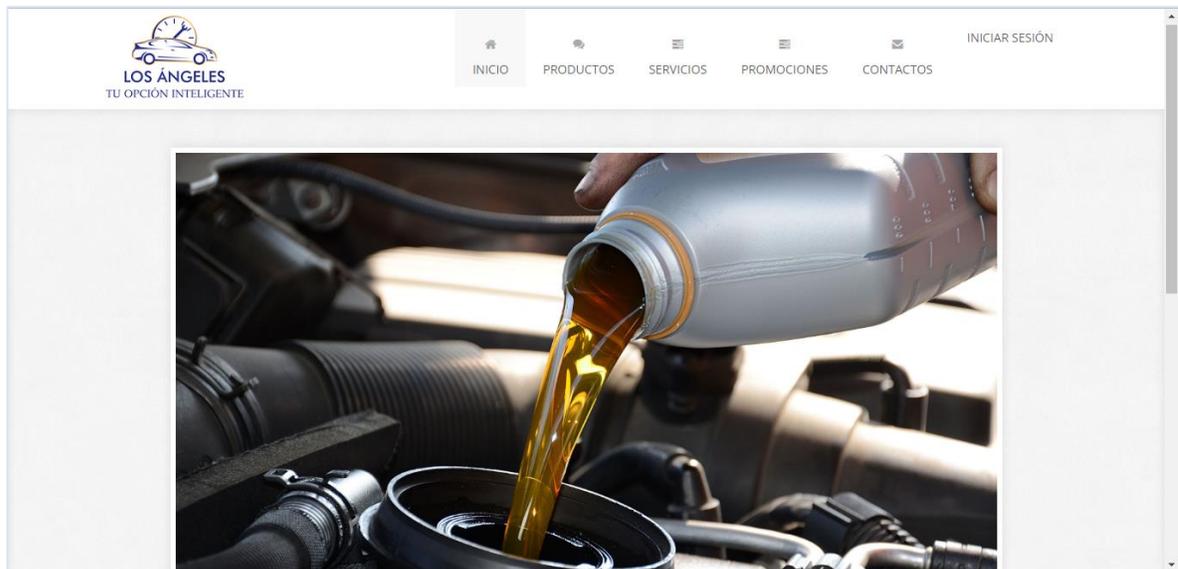


Figura 7-2: Página de inicio de la empresa
Realizado por: Yupangui María, 2019

2.4.8.2. *Pantalla de inicio de sesión*

En la **Figura 8-2** se visualiza la interfaz de inicio de sesión a la cual solo podrá autenticarse solo el administrador y empleados o el personal autorizado por el administrador del sistema.

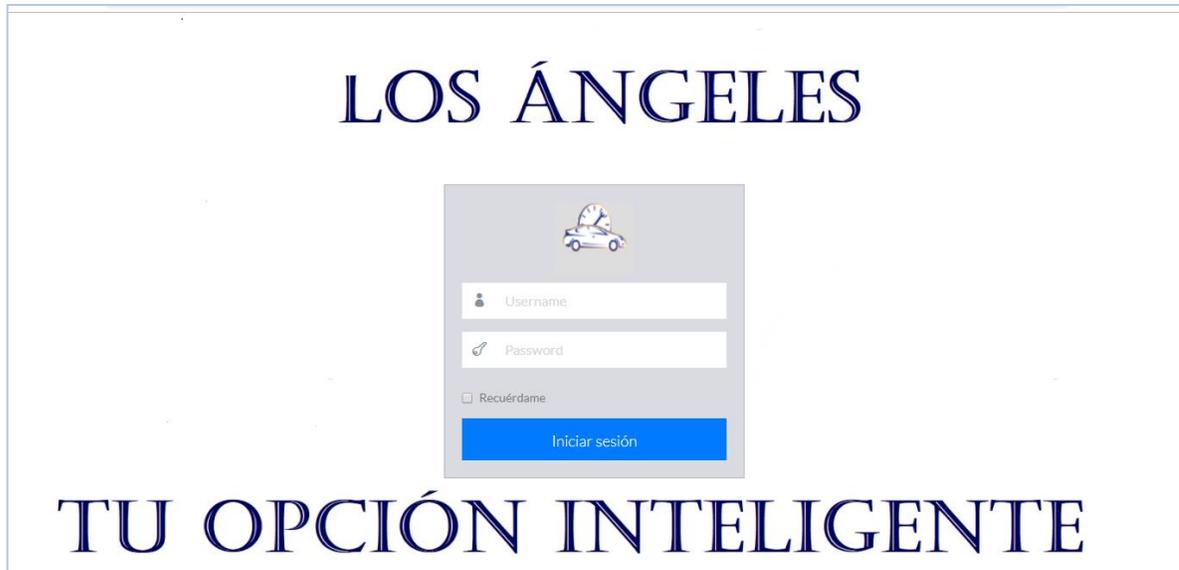


Figura 8-2: Página de inicio de sesión
Realizado por: Yupangui María, 2019

2.4.8.3. *Página principal administrador*

La página principal del administrador cuenta con todas las funcionalidades y reportes del sistema, en el menú de la izquierda se puede visualizar todas las funciones implementadas.

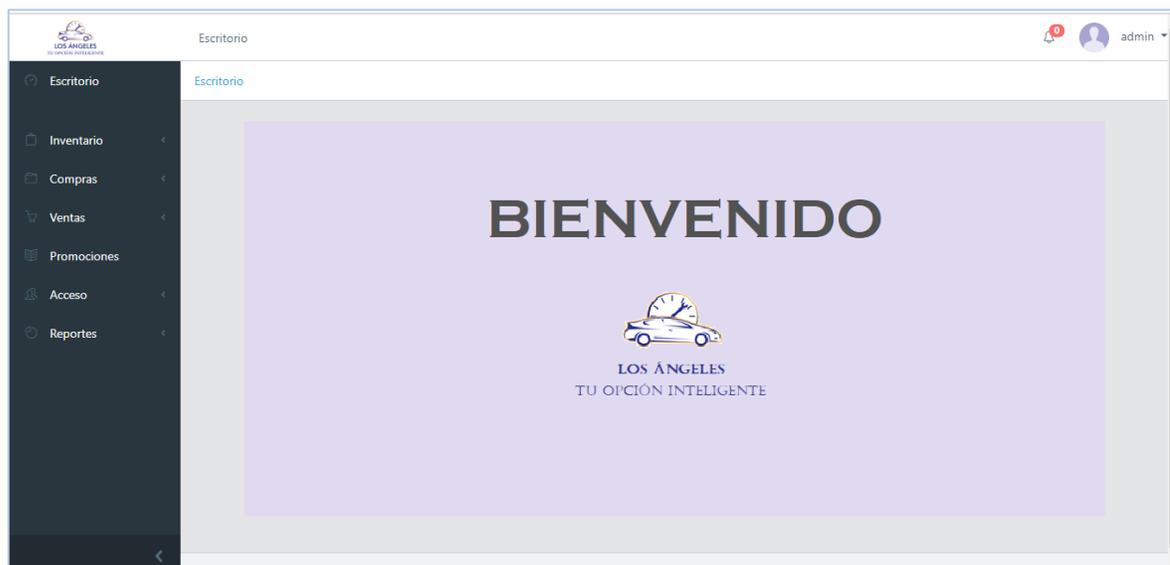


Figura 9-2: Página principal - administrador
Realizado por: Yupangui María, 2019

2.4.8.4. Interfaz de formularios de ingreso de datos

En la **Figura 10-2** se visualiza la interfaz de los formularios para el registro, cada formulario cuenta con los campos y botones necesarios para cada transacción.

The screenshot shows a web application interface with a sidebar menu on the left containing items like 'Escritorio', 'Inventario', 'Categorías', 'Artículos', 'Ingresos', 'Ventas', 'Venta', 'Clientes', 'Promociones', 'Acceso', and 'Reportes'. The main content area is partially obscured by a modal window titled 'Registrar Artículo'. The modal contains the following fields and elements:

- Categoría:** A dropdown menu with the text 'Seleccione una categoría'.
- Código:** A text input field with the placeholder 'Ingrese el código'. Below it, the text 'Generando código de barras.' is displayed.
- Nombre:** A text input field with the placeholder 'Ingrese el nombre del artículo'.
- Precio Venta:** A text input field with the value '0'.
- Stock:** A text input field with the value '0'.
- Presentación:** A text input field with the placeholder 'Ingrese la presentación del artículo'.
- At the bottom right of the modal are two buttons: 'Cerrar' (grey) and 'Guardar' (blue).

In the background, a table is visible with the following data:

Stock	Presentación	Estado
30	GALON	Activo
24	MEDIANO	Activo
19	LITRO	Activo
21	LITRO	Activo
21	LITRO	Activo
10	LITRO	Activo
11	LITRO	Activo

Figura 10-2: Interfaz de usuario - registro de artículos

Realizado por: Yupangui María, 2019

2.4.8.5. Interfaz de registro de ventas

Para el registro de ventas se ha elaborado una interfaz similar a una factura física, para una mejor experiencia del usuario.

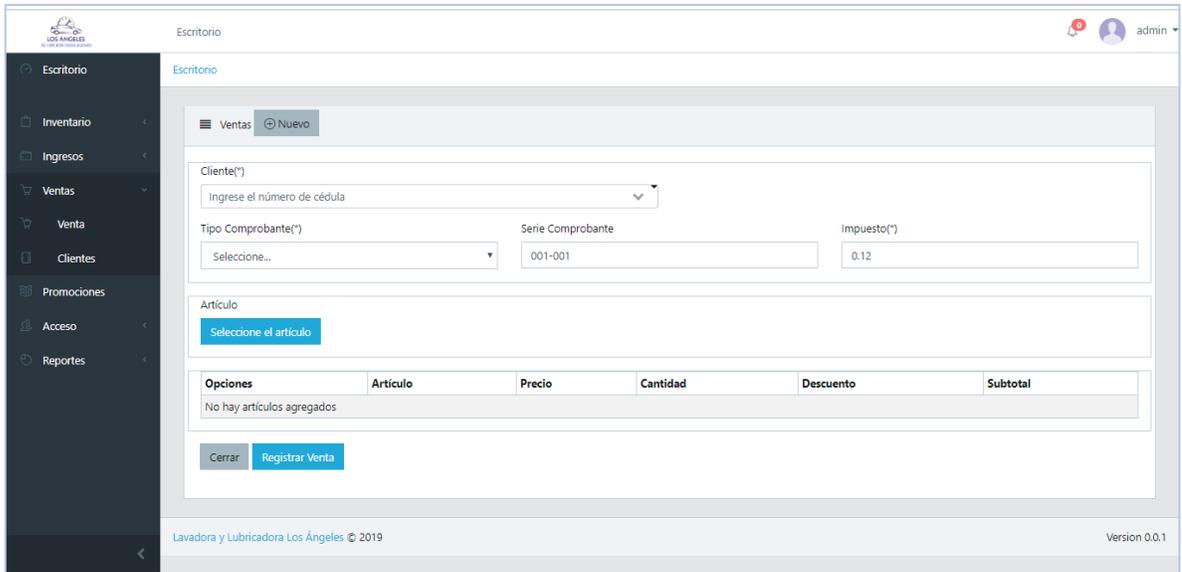


Figura 11-2: Interfaz de usuario - registro de una venta
Realizado por: Yupangui María, 2019

2.4.8.6. Vista de la Interfaz listado de compras dispositivo móvil

En la **Figura 12-2** se observa un interfaz adaptado a un dispositivo móvil del listado de registro de compras.

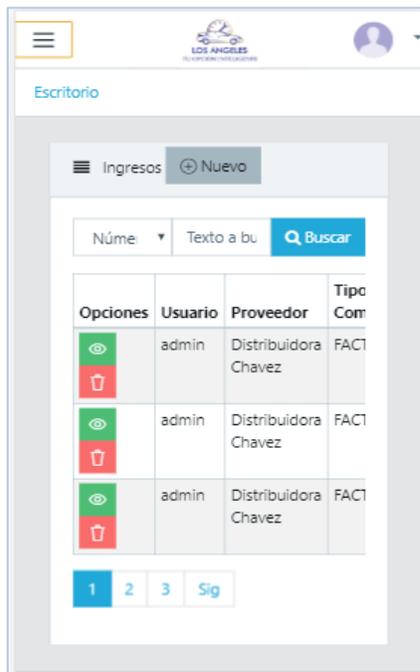


Figura 12-2: Interfaz vista móvil
Realizado por: Yupangui María, 2019

Cabe recalcar que las todas las interfaces des sistema son completamente responsive, es decir que su contenido se adapta a cualquier tipo de dispositivo, esto se lo realizo haciendo uso del framework Bootstrap.

2.4.9. *Diseño de la base de datos*

Con la finalidad de obtener acceso a la información que maneja el negocio se diseñó la base de datos, la cual permitirá tener acceso a los diferentes usuarios y podrá gestionar las acciones de acuerdo al rol que posea. Para identificar las entidades, atributos y relaciones de acuerdo al funcionamiento del negocio se procedió a realizar el diagrama Entidad – Relación. En base a este diagrama se obtiene el diseño del modelo lógico de la base de datos que cuenta con un total de 14 tablas, la cual almacenará la información referente al negocio y permitirá y correcto almacenamiento y funcionamiento del sistema.

Las tablas se encuentran identificadas por el nombre de la entidad o funcionalidad que cumplirá dentro del sistema, además se encuentran definidas sus respectivas claves primarias y claves foráneas, una particularidad de las llaves primarias es que todas son de tipo int y auto incrementables, esta clave está presente en todas las tablas. El resto de los datos que contienen las tablas cuentan con los siguientes tipos de datos:

- Tipo de dato varchar: para especificar nombres, descripciones, etc.
- Tipo de dato tinyint para especificar la condición de los usuarios, roles, promociones, artículos y categorías.
- Tipo de dato decimal se usó para especificar valores monetarios
- Tipo de dato datetime para especificar fechas

Además, en la **Figura 13-2**, se puede visualizar las relaciones entre las tablas (uno a uno, uno a muchos y muchos a muchos). Cabe recalcar que la tabla migrations es creada por framework Laravel para llevar un registro de todas las migraciones y controlar cual ejecutar y cual no, dependiendo del comando utilizado por el desarrollador. A continuación, se muestra el diagrama lógico de la base de datos implementada en el sistema web.

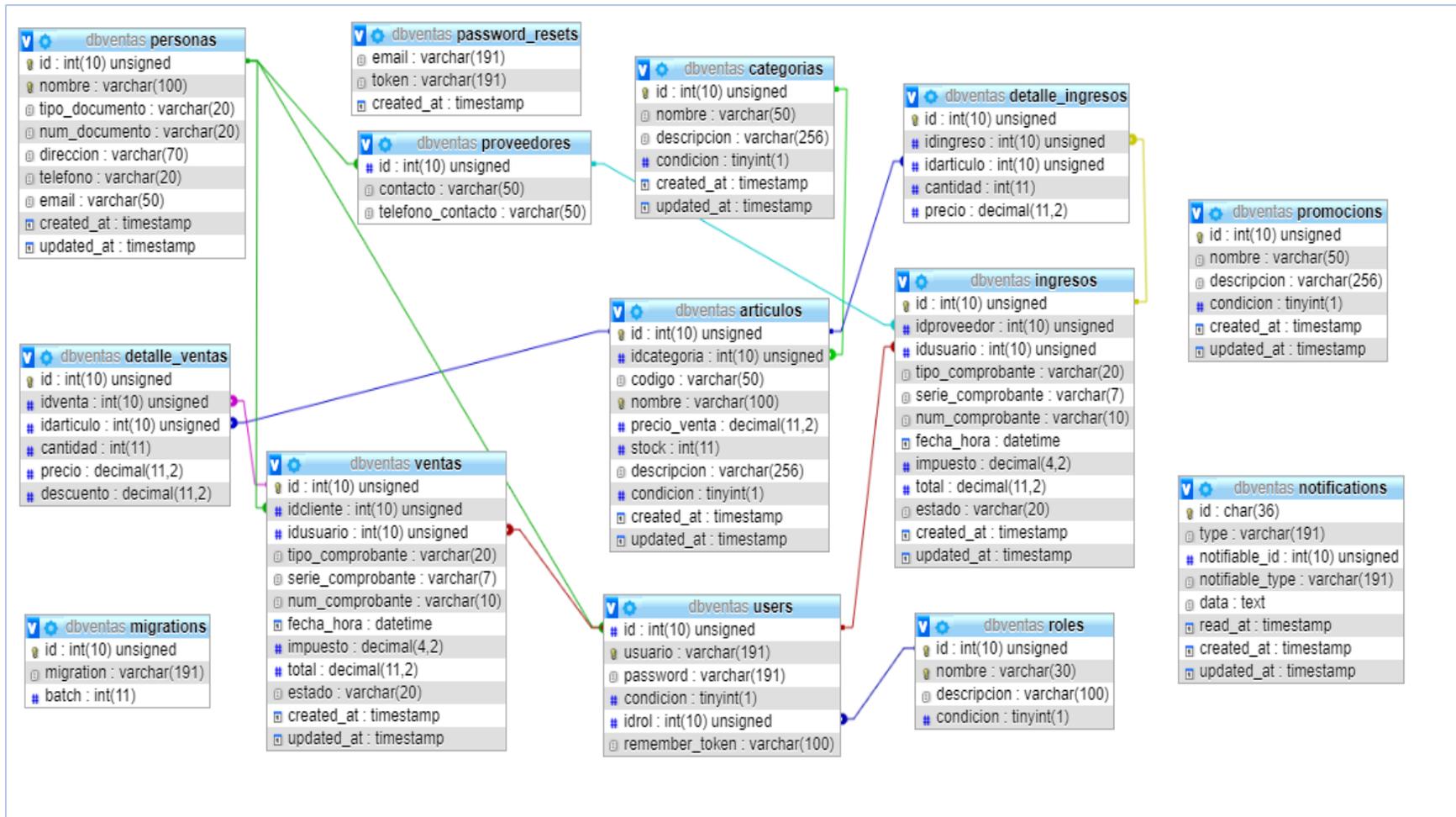


Figura 13-2: Diagrama de base de datos - modelo lógico
 Realizado por: Yupangui María, 2019

2.4.9.1. Diccionario de datos

El diccionario de datos es utilizado para describir cada una de las tablas que componen la base de datos, en esta se describe el nombre del campo, claves primarias, claves foráneas, tipo de dato, tamaño y si el campo es nulo.

Tabla 14-2: Tabla - personas

Nombre del campo	Clave primaria	Clave foránea	Tipo de dato	Tamaño	Nulo
Id	Si	No	Int	10	No
Nombre	No	No	varchar	100	No
tipo_documento	No	No	varchar	20	No
num_documento	No	No	varchar	15	No
Dirección	No	No	varchar	70	No
Teléfono	No	No	varchar	10	No
Email	No	No	varchar	50	No
created_at	No	No	timestamp		Si
updated_at	No	No	timestamp		Si

Realizado por: Yupangui María, 2019

El total de tablas del diccionario de datos se encuentra en el **Anexo A**.

2.5. Fase de desarrollo

2.5.6. *Sprint backlog*

El sprint backlog es el conjunto de historias de usuario e historias técnicas las cuales van a ser desarrollados para la creación del sistema.

Para el desarrollo del sistema se cuenta con 18 sprint, cada sprint tiene una duración de una semana es decir 40 horas (20 puntos estimados). El plan de entrega de cada sprint se detalla a continuación.

Tabla 15-2: Sprint Backlog

N° de sprint	Id	Requerimientos	Puntos estimados	Puntos Estimados totales
1	HT-01	Como desarrollador deseo recolectar la información necesaria para el desarrollo del sistema.	8	20
	HT-02	Como desarrollador deseo instalar las herramientas y tecnologías necesarias para el desarrollo del sistema.	4	
	HT-03	Como desarrollador deseo conocer los requerimientos del sistema	8	
2	HT-04	Como desarrollador deseo diseñar la arquitectura del sistema	4	20
	HT-05	Como desarrollador deseo definir un estándar de codificación	4	
	HT-06	Como desarrollador deseo diseñar la base de datos	8	
	HT-07	Como desarrollador deseo diseñar la interfaz de usuario	4	
3	HU-01	Como usuario, deseo visualizar una página principal	8	20
	HU-02	Como administrador, empleado deseo ingresar al sistema utilizando mis credenciales (usuario y contraseña)	6	
	HU-03	Como administrador, empleado, deseo tener una interfaz principal donde se observen las opciones que se puedan realizar	6	
4	HU-04	Como administrador, deseo registrar una categoría	6	20
	HU-05	Como administrador, deseo actualizar una categoría	6	
	HU-06	Como administrador, deseo eliminar una categoría	4	
	HU-07	Como administrador, deseo buscar una categoría por nombre o descripción	4	
5	HU-08	Como administrador, deseo listar las categorías	4	20

	HU-09	Como administrador, deseo ingresar un nuevo artículo	4	
	HU-10	Como administrador, deseo actualizar un artículo	4	
	HU-11	Como administrador, deseo eliminar un artículo	4	
	HU-12	Como administrador, deseo buscar un artículo por nombre o código.	4	
6	HU-13	Como administrador, deseo listar los artículos	4	20
	HU-14	Como administrador, deseo obtener un reporte de todos los artículos en inventario en formato PDF	8	
	HU-15	Como administrador, empleado, deseo registrar una compra	8	
7	HU-16	Como administrador, empleado, deseo listar los registros de compras	4	20
	HU-17	Como administrador, empleado, deseo visualizar el detalle de una compra	4	
	HU-18	Como administrador, empleado, deseo anular una compra	4	
	HU-19	Como administrador, empleado, deseo buscar una compra por número de comprobante, tipo de comprobante.	4	
	HU-20	Como administrador, empleado, deseo registrar un nuevo proveedor	4	
8	HU-21	Como administrador, empleado, deseo actualizar los datos de un proveedor	6	20
	HU-22	Como administrador, empleado, deseo buscar un proveedor por nombre o número de documento.	6	
	HU-23	Como administrador, empleado, deseo eliminar un proveedor	4	
	HU-24	Como administrador, empleado, deseo listar todos los proveedores.	4	
9	HU-25	Como administrador, empleado, deseo registrar una venta	12	20

	HU-26	Como administrador, empleado, deseo listar las ventas	4	
	HU-27	Como administrador, empleado, deseo visualizar el detalle de una venta	4	
10	HU-28	Como administrador, empleado, deseo anular una venta	4	20
	HU-29	Como administrador, empleado, deseo generar una factura en formato PDF	8	
	HU-30	Como administrador, empleado, deseo buscar una venta por número de comprobante, tipo de comprobante.	4	
	HU-31	Como administrador, empleado, deseo, registrar un cliente	4	
11	HU-32	Como administrador, empleado, deseo actualizar los datos de un cliente	4	20
	HU-33	Como administrador, empleado, deseo listar todos los clientes	4	
	HU-34	Como administrador, empleado, deseo buscar un cliente por nombre o número de documento.	4	
	HU-35	Como administrador, deseo registrar una promoción.	4	
	HU-36	Como administrador, deseo eliminar una promoción	4	
12	HU-37	Como administrador, deseo actualizar una promoción	4	20
	HU-38	Como administrador, deseo listar las promociones.	4	
	HU-39	Como administrador, deseo buscar una promoción por nombre o descripción.	4	
	HU-40	Como administrador, deseo listar los roles existentes	4	
	HU-41	Como administrador, deseo buscar un rol por nombre o descripción.	4	

13	HU-42	Como administrador, deseo registrar un nuevo usuario en el sistema	4	20
	HU-43	Como administrador, deseo listar todos los usuarios del sistema	4	
	HU-44	Como administrador, deseo actualizar los datos de un usuario	4	
	HU-45	Como administrador, deseo eliminar un usuario del sistema	4	
	HU-46	Como administrador, deseo buscar un usuario por nombre o número de documento.	4	
14	HU-47	Como administrador, deseo visualizar un reporte de todas las compras	6	20
	HU-48	Como administrador, deseo visualizar un reporte de todas las ventas.	6	
	HU-49	Como administrador, deseo visualizar un reporte de compras en un intervalo de fechas.	8	
15	HU-50	Como administrador, deseo visualizar un reporte de ventas en un intervalo de fechas.	8	20
	HU-51	Como administrador, deseo visualizar un gráfico estadístico de compras y ventas de los últimos meses.	8	
	HU-52	Como administrador, empleado, deseo visualizar un panel de notificaciones mostrando el número de compras y ventas registradas desde su ultimo ingreso al sistema.	4	
16	HT-08	Como desarrollador, deseo realizar la detección y corrección de errores	20	20
17	HT-09	Como desarrollador, deseo analizar los resultados	16	20
	HT-10	Como desarrollador, deseo determinar las conclusiones y recomendaciones	4	
18	HT-11	Como desarrollador, deseo realizar el manual de usuario	4	20

	HT-12	Como desarrollador, deseo realizar el documento del trabajo de titulación	12	
	HU-53	Como desarrollador, deseo dar capacitación a los usuarios	4	

Realizado por: Yupangui María, 2019

Sprint 1: Mediante reuniones con el gerente general de la Lavadora y Lubricadora “Los Ángeles” se consiguió recolectar los requerimientos e información necesaria para el desarrollo del sistema web.

Además, se seleccionó e instaló las herramientas y tecnologías necesarias para el inicio del desarrollo a continuación, se describe cada una de las herramientas instaladas.

- Gestor de dependencias composer versión 1.7.2.
- Framework Laravel versión 5.6.
- Editor de texto Sublime Text 3.
- Gestor de base de datos MySQL
- Lenguaje de programación PHP versión 7.2.4, ya que es un requisito propio de Laravel que la versión a utilizar sea PHP 7 o superior.

Tabla 16-2: Sprint 1

Sprint 1					
Inicio: 07/01/2019		Fin: 11/01/2019		Esfuerzo estimado: 40	Esfuerzo real: 40
Pila del sprint					
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable	
HT-01	Como desarrollador deseo recolectar la información necesaria para el desarrollo del sistema.	8	Análisis	María Yupangui	
HT-02	Como desarrollador deseo instalar las herramientas y tecnologías necesarias para el desarrollo del sistema.	4	Análisis	María Yupangui	
HT-03	Como desarrollador deseo conocer los requerimientos del sistema	8	Análisis	María Yupangui	

Realizado por: Yupangui María, 2019

Sprint 2: En este sprint se desarrolla el diseño de la arquitectura del sistema, se define un estándar para la codificación, se realiza el diseño de la base de datos con el fin de almacenar la información del sistema web y se concluye el sprint con el diseño de la interfaz de usuario.

Tabla 17-2: Sprint 2

Sprint 2				
Inicio: 14/01/2019		Fin: 18/01/2019	Esfuerzo estimado: 40	Esfuerzo real: 40
Pila del sprint				
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable
HT-04	Como desarrollador deseo diseñar la arquitectura del sistema	4	Análisis Diseño	María Yupangui
HT-05	Como desarrollador deseo definir un estándar de codificación	4	Análisis	María Yupangui
HT-06	Como desarrollador deseo diseñar la base de datos	8	Análisis Diseño	María Yupangui
HT-07	Como desarrollador deseo diseñar la interfaz de usuario	4	Análisis	María Yupangui

Realizado por: Yupangui María, 2019

Sprint 3: En este sprint se inicia con la codificación de las funcionalidades del sistema, se diseña una página de inicio la cual servirá para dar a conocer información de los productos y servicios que ofrece la empresa, además, se implementa el módulo de autenticación de los usuarios y se realiza la página principal del administrador/empleador con sus respectivas opciones en el sistema.

Tabla 18-2: Sprint 3

Sprint 3				
Inicio: 21/01/2019		Fin: 25/01/2019	Esfuerzo estimado: 40	Esfuerzo real: 44
Pila del sprint				
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable

HU-01	Como usuario, deseo visualizar una página principal	8	Codificación	María Yupangui
HU-02	Como administrador, empleado deseo ingresar al sistema utilizando mis credenciales (usuario y contraseña)	6	Codificación	María Yupangui
HU-03	Como administrador, empleado, deseo tener una interfaz principal donde se observen las opciones que se puedan realizar	6	Codificación	María Yupangui

Realizado por: Yupangui María, 2019

Sprint 4: En este sprint se realiza la codificación para el registro, actualización, eliminación y búsqueda de una categoría por algún criterio.

Tabla 19-2: Sprint 4

Sprint 4				
Inicio: 28/01/2019		Fin: 01/02/2019		Esfuerzo estimado: 40
Esfuerzo real: 40				
Pila del sprint				
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable
HU-04	Como administrador, deseo registrar una categoría	6	Codificación	María Yupangui
HU-05	Como administrador, deseo actualizar una categoría	6	Codificación	María Yupangui
HU-06	Como administrador, deseo eliminar una categoría	4	Codificación	María Yupangui
HU-07	Como administrador, deseo buscar una categoría por nombre o descripción	4	Codificación	María Yupangui

Realizado por: Yupangui María, 2019

Sprint 5: En el sprint 5 se codifica el listado de categorías y se continua con el registro, actualización, eliminación y búsqueda de un artículo.

Tabla 20-2: Sprint 5

Sprint 5				
Inicio: 04/02/2019		Fin: 08/02/2019		Esfuerzo estimado: 40
Esfuerzo real: 40				
Pila del sprint				
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable
HU-08	Como administrador, deseo listar las categorías	4	Codificación	María Yupangui
HU-09	Como administrador, deseo ingresar un nuevo artículo	4	Codificación	María Yupangui
HU-10	Como administrador, deseo actualizar un artículo	4	Codificación	María Yupangui
HU-11	Como administrador, deseo eliminar un artículo	4	Codificación	María Yupangui
HU-12	Como administrador, deseo buscar un artículo por nombre o código.	4	Codificación	María Yupangui

Realizado por: Yupangui María, 2019

Sprint 6: En este sprint se codifica el listado de los artículos y un reporte en archivo PDF de los mismos. Además, se continua con la codificación para el registro de una compra.

Tabla 21-2: Sprint 6

Sprint 6				
Inicio: 11/02/2019		Fin: 15/02/2019		Esfuerzo estimado: 40
Esfuerzo real: 40				
Pila del sprint				
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable
HU-13	Como administrador, deseo listar los artículos	4	Codificación	María Yupangui
HU-14	Como administrador, deseo obtener un reporte de todos los artículos en inventario en formato PDF	8	Codificación	María Yupangui

HU-15	Como administrador, empleado, deseo registrar una compra	8	Codificación	María Yupangui
--------------	--	---	--------------	-------------------

Realizado por: Yupangui María, 2019

Sprint 7: En el sprint 7 se realiza la codificación para el listado de compras hechas por la empresa, además, se codifica la opción de visualizar el detalle una compra, anular y buscar una compra mediante un criterio de búsqueda. Para culminar el sprint se genera el código para el registro de un nuevo proveedor de la empresa.

Tabla 22-2: Sprint 7

Sprint 7					
Inicio: 18/02/2019		Fin: 22/02/2019		Esfuerzo estimado: 40	Esfuerzo real: 40
Pila del sprint					
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable	
HU-16	Como administrador, empleado, deseo listar los registros de compras	4	Codificación	María Yupangui	
HU-17	Como administrador, empleado, deseo visualizar el detalle de una compra	4	Codificación	María Yupangui	
HU-18	Como administrador, empleado, deseo anular una compra	4	Codificación	María Yupangui	
HU-19	Como administrador, empleado, deseo buscar una compra por número de comprobante, tipo de comprobante.	4	Codificación	María Yupangui	
HU-20	Como administrador, empleado, deseo registrar un nuevo proveedor	4	Codificación	María Yupangui	

Realizado por: Yupangui María, 2019

Sprint 8: En este sprint se codifica para que el administrador o empleado tenga la opción de actualizar datos, eliminación, búsqueda de un proveedor y listado de todos los proveedores de la empresa.

Tabla 23-2: Sprint 8

Sprint 8					
Inicio: 25/02/2019		Fin: 01/03/2019		Esfuerzo estimado: 40	Esfuerzo real: 40
Pila del sprint					
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable	
HU-21	Como administrador, empleado, deseo actualizar los datos de un proveedor	6	Codificación	María Yupangui	
HU-22	Como administrador, empleado, deseo buscar un proveedor por nombre o número de documento.	6	Codificación	María Yupangui	
HU-23	Como administrador, empleado, deseo eliminar un proveedor	4	Codificación	María Yupangui	
HU-24	Como administrador, empleado, deseo listar todos los proveedores.	4	Codificación	María Yupangui	

Realizado por: Yupangui María, 2019

Sprint 9: En este sprint se realiza la codificación para el registro de una venta y la opción de visualizar el detalle, además el sistema lista todas las ventas registradas cada registro cuenta con la opción de modificar, anular, visualizar detalle, etc.

Tabla 24-2: Sprint 9

Sprint 9					
Inicio: 04/03/2019		Fin: 08/03/2019		Esfuerzo estimado: 40	Esfuerzo real: 40
Pila del sprint					
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable	
HU-25	Como administrador, empleado, deseo registrar una venta	12	Codificación	María Yupangui	
HU-26	Como administrador, empleado, deseo listar las ventas	4	Codificación	María Yupangui	

HU-27	Como administrador, empleado, deseo visualizar el detalle de una venta	4	Codificación	María Yupangui
--------------	--	---	--------------	-------------------

Realizado por: Yupangui María, 2019

Sprint 10: En este sprint se continua con la codificación para la opción de anular una venta y generar la factura en formato PDF. También se codifica la búsqueda de ventas realizadas y el registro de un nuevo cliente al sistema.

Tabla 25-2: Sprint 10

Sprint 10					
Inicio: 11/03/2019		Fin: 15/03/2019		Esfuerzo estimado: 40	Esfuerzo real: 42
Pila del sprint					
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable	
HU-28	Como administrador, empleado, deseo anular una venta	4	Codificación	María Yupangui	
HU-29	Como administrador, empleado, deseo generar una factura en formato PDF	8	Codificación	María Yupangui	
HU-30	Como administrador, empleado, deseo buscar una venta por número de comprobante, tipo de comprobante.	4	Codificación	María Yupangui	
HU-31	Como administrador, empleado, deseo, registrar un cliente	4	Codificación	María Yupangui	

Realizado por: Yupangui María, 2019

Sprint 11: En este sprint se codifica la actualización y búsqueda de un cliente y por su puesto el listado de todos los clientes registrados en el sistema. Además, se realiza la codificación para el registro y eliminación de una promoción.

Tabla 26-2: Sprint 11

Sprint 11				
Inicio: 18/03/2019		Fin: 22/03/2019		Esfuerzo estimado: 40
Esfuerzo real: 40				
Pila del sprint				
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable
HU-32	Como administrador, empleado, deseo actualizar los datos de un cliente	4	Codificación	María Yupangui
HU-33	Como administrador, empleado, deseo listar todos los clientes	4	Codificación	María Yupangui
HU-34	Como administrador, empleado, deseo buscar un cliente por nombre o número de documento.	4	Codificación	María Yupangui
HU-35	Como administrador, deseo registrar una promoción.	4	Codificación	María Yupangui
HU-36	Como administrador, deseo eliminar una promoción	4	Codificación	María Yupangui

Realizado por: Yupangui María, 2019

Sprint 12: En este sprint se codifica la actualización, búsqueda y listado de la gestión promoción, además se realiza el listado y búsqueda de los roles existentes.

Tabla 27-2: Sprint 12

Sprint 12				
Inicio: 25/03/2019		Fin: 29/03/2019		Esfuerzo estimado: 40
Esfuerzo real: 40				
Pila del sprint				
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable
HU-37	Como administrador, deseo actualizar una promoción	4	Codificación	María Yupangui
HU-38	Como administrador, deseo listar las promociones.	4	Codificación	María Yupangui

HU-39	Como administrador, deseo buscar una promoción por nombre o descripción.	4	Codificación	María Yupangui
HU-40	Como administrador, deseo listar los roles existentes	4	Codificación	María Yupangui
HU-41	Como administrador, deseo buscar un rol por nombre o descripción	4	Codificación	María Yupangui

Realizado por: Yupangui María, 2019

Sprint 13: En este sprint se realiza la codificación para el registro, actualización, eliminación, búsqueda de un usuario del sistema, además de generar el listado de todos los usuarios del sistema.

Tabla 28-2: Sprint 13

Sprint 13					
Inicio: 01/04/2019		Fin: 05/04/2019		Esfuerzo estimado: 40	Esfuerzo real: 40
Pila del sprint					
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable	
HU-42	Como administrador, deseo registrar un nuevo usuario en el sistema	4	Codificación	María Yupangui	
HU-43	Como administrador, deseo listar todos los usuarios del sistema	4	Codificación	María Yupangui	
HU-44	Como administrador, deseo actualizar los datos de un usuario	4	Codificación	María Yupangui	
HU-45	Como administrador, deseo eliminar un usuario del sistema	4	Codificación		
HU-46	Como administrador, deseo buscar un usuario por nombre o número de documento.	4			

Realizado por: Yupangui María, 2019

Sprint 14: En este sprint se codifico el reporte de todas las compras y ventas registradas en el sistema, además se realizó la codificación para generar un reporte de compras en un rango de fechas con sus respectivos archivos en formato PDF.

Tabla 29-2: Sprint 14

Sprint 14					
Inicio: 08/04/2019		Fin: 12/04/2019		Esfuerzo estimado: 40	Esfuerzo real: 44
Pila del sprint					
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable	
HU-47	Como administrador, deseo visualizar un reporte de todas las compras	6	Codificación	María Yupangui	
HU-48	Como administrador, deseo visualizar un reporte de todas las ventas	6	Codificación	María Yupangui	
HU-49	Como administrador, deseo visualizar un reporte de compras en un intervalo de fechas	8	Codificación	María Yupangui	

Realizado por: Yupangui María, 2019

Sprint 15: En este sprint se realiza la codificación para generar un reporte de ventas en un rango de fechas, además se genera dos gráficos estadísticos de compras y de ventas de los últimos meses, indicando el mes y el total de compra o venta. Y como requerimiento final se elabora un panel de notificación para indicar cuantas compras y ventas se han realizado desde el ultimo ingreso al sistema.

Tabla 30-2: Sprint 15

Sprint 15					
Inicio: 15/04/2019		Fin: 19/04/2019		Esfuerzo estimado: 40	Esfuerzo real: 40
Pila del sprint					
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable	

HU-50	Como administrador, deseo visualizar un reporte de ventas en un intervalo de fechas.	8	Codificación	María Yupangui
HU-51	Como administrador, deseo visualizar un gráfico estadístico de compras y ventas de los últimos meses.	8		
HU-52	Como administrador, empleado, deseo visualizar un panel de notificaciones mostrando el número de compras y ventas registradas desde su ultimo ingreso al sistema.	4		

Realizado por: Yupangui María, 2019

Sprint 16: En este sprint se realiza la detección y corrección de las pruebas realizadas al sistema.

Tabla 31-2: Sprint 16

Sprint 16				
Inicio: 22/04/2019		Fin: 26/04/2019		Esfuerzo estimado: 40
Esfuerzo real: 40				
Pila del sprint				
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable
HT-08	Como desarrollador, deseo realizar la detección y corrección de errores	20		María Yupangui

Realizado por: Yupangui María, 2019

Sprint 17: En este sprint se analiza los resultados para observar si el sistema cumple y ayuda a resolver la problemática planteada en el presente trabajo de titulación para posteriormente determinar las conclusiones y recomendaciones respectivas.

Tabla 32-2: Sprint 17

Sprint 17				
Inicio: 29/04/2019		Fin: 03/05/2019		Esfuerzo estimado: 40
Esfuerzo real: 40				
Pila del sprint				
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable
HT-09	Como desarrollador, deseo analizar los resultados	16		María Yupangui
HT-10	Como desarrollador, deseo determinar las conclusiones y recomendaciones	4		María Yupangui

Realizado por: Yupangui María, 2019

Sprint 18: En este sprint se realiza el manual de usuario del sistema, el documento del trabajo de titulación y finalmente se procede con la capacitación respectiva a los usuarios.

Tabla 33-2: Sprint 18

Sprint 18				
Inicio: 06/05/2019		Fin: 10/05/2019		Esfuerzo estimado: 40
Esfuerzo real: 40				
Pila del sprint				
Product backlog Id	Descripción	Esfuerzo (H)	Tipo	Responsable
HT-11	Como desarrollador, deseo realizar el manual de usuario	4		María Yupangui
HT-12	Como desarrollador, deseo realizar el documento del trabajo de titulación	12		María Yupangui
HU-53	Como desarrollador, deseo dar capacitación a los usuarios	4		María Yupangui

Realizado por: Yupangui María, 2019

2.5.7. Historias de usuario

La metodología SCRUM utiliza las historias de usuario para describir los requerimientos del sistema este posee información como el ID, Nombre de la historia, el usuario que va a ejecutar la funcionalidad, el número de sprint asignado, la prioridad que tiene en el proyecto, la fecha de inicio y fin, los puntos estimados y reales del desarrollo y una pequeña descripción.

Además, en la **Tabla 34-2** se especifica las pruebas de aceptación las cuales ayudaran a verificar que las funcionalidades implementadas cumplan con los requerimientos. Por último, se especifica las tareas de las cuales son un conjunto de actividades que ayudaran a desarrollar una funcionalidad del sistema.

Tabla 34-2: Historia de usuario 29 - Sprint 10

HISTORIA DE USUARIO			
ID: HU-29		Nombre de la historia: Como administrador, empleado, deseo generar una factura en formato PDF	
Usuario: Administrador, empleado		Sprint: 10	
Prioridad en el negocio: Alta		Puntos estimados: 8	
(Alta/Media/Baja)		Puntos reales: 8	
Fecha inicio: 12/03/2019		Fecha fin: 13/03/2019	
Descripción: Como administrador, empleado, deseo generar una factura en formato PDF con el objetivo de emitir al cliente.			
Pruebas de aceptación			
ID_PA	Nombre	Resultado	Responsable
HU-29 PA_01	Verificar que la información sea correcta	Exitosa	María Yupangui
Tareas de ingeniería			
ID_TI	Descripción	Esfuerzo	
HU-29 T1-01	Creación de la vista para emitir la factura	2	
HU-29 TI-02	Recuperar los datos del cliente, usuario y detalle de la venta realizada.	3	
HU-29 TI-03	Descargar el documento generado en formato PDF	3	

Realizado por: Yupangui María, 2019

En la historia de usuario HU-29, perteneciente al sprint 10, se observa que el usuario es el administrador(es) y empleado (s), posee 1 prueba de aceptación la cual verifica la funcionalidad y 3 tareas de ingeniería. Luego de realizar todas las historias de usuario en el formato de la **Tabla 34-2** se ha obtenido un total de 53 historias de usuario y 13 historias técnicas. Mediante el uso de las pruebas de aceptación se concluyó que todas obtuvieron una evaluación exitosa y esto quiere decir que las funcionalidades implementadas cumplen con los requerimientos establecidos por el Product Owner.

El total de historias de usuario e historias técnicas se encuentran en el **Anexo B**.

2.6. Instalación, configuración y codificación

Para la codificación se utilizó la herramienta Sublime Text el cual permitirá la edición de código, Composer el cual permitirá la descarga automática de las librerías y paquetes necesarios para el proyecto y el sistema gestor MySQL para el almacenamiento de la información. Cabe mencionar que para el desarrollo del sistema se utiliza el framework de desarrollo Laravel y el servidor Apache el cual permitirá la funcionalidad correcta del código PHP.

Instalación del framework Laravel

1. Instalar el gestor de dependencias Composer para Windows desde su página oficial en la sección de descargas mediante el siguiente link <https://getcomposer.org/download/> como se puede visualizar en la **Figura 14-2**.

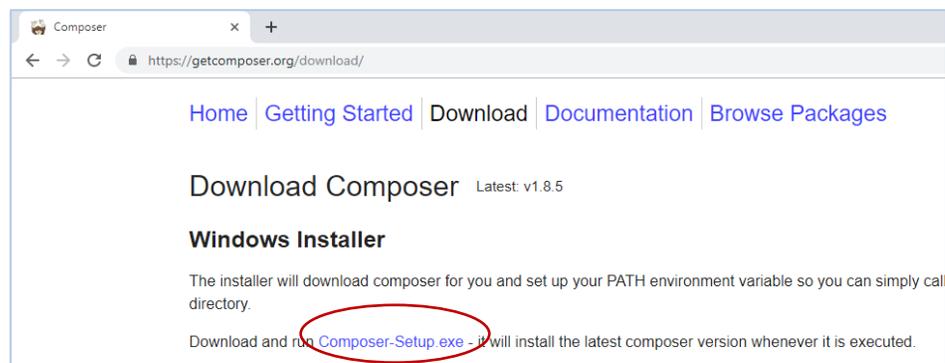
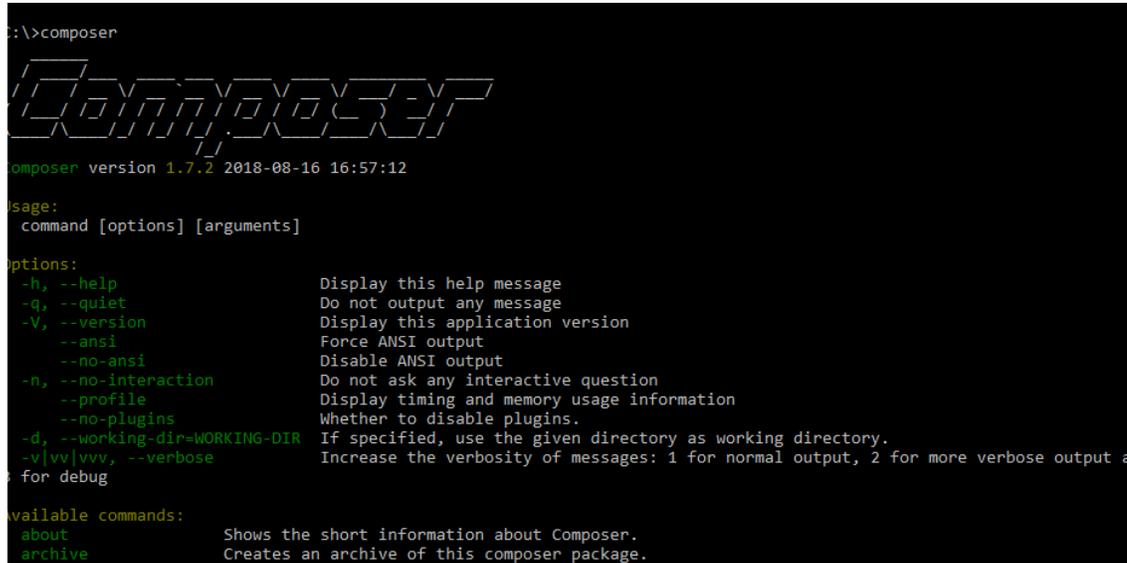


Figura 14-2: Descargar gestor de dependencias Composer
Realizado por: Yupangui María, 2019

2. Para verificar el funcionamiento de Composer, se ingresa al símbolo del sistema y se introduce la siguiente línea de código en la consola `c:/> composer` (tecla enter). La **Figura 15-2** se observa que Composer está funcionando de una manera correcta.



```
C:\>composer

Composer

Composer version 1.7.2 2018-08-16 16:57:12

Usage:
  command [options] [arguments]

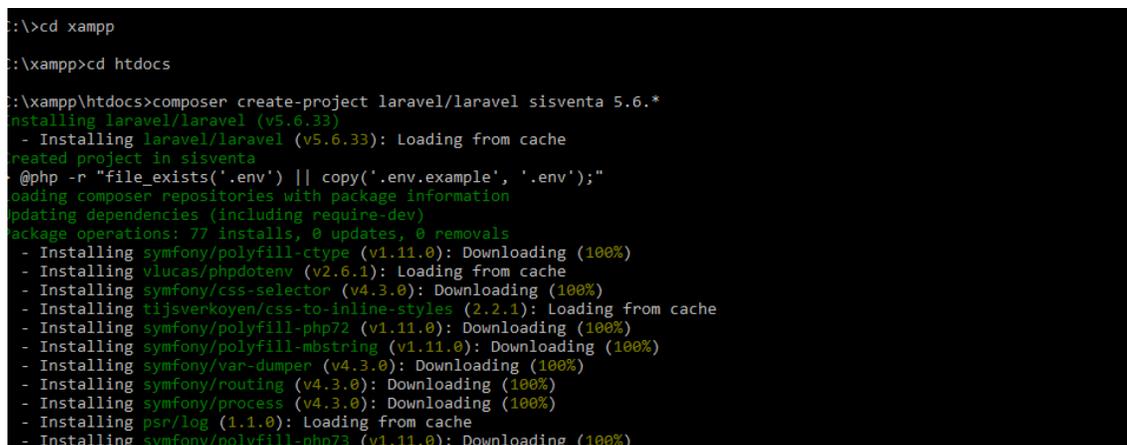
Options:
  -h, --help                Display this help message
  -q, --quiet                Do not output any message
  -V, --version              Display this application version
      --ansi                 Force ANSI output
      --no-ansi              Disable ANSI output
  -n, --no-interaction      Do not ask any interactive question
      --profile               Display timing and memory usage information
      --no-plugins           Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  -v|vv|vvv, --verbose      Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3
                             for debug

Available commands:
  about      Shows the short information about Composer.
  archive    Creates an archive of this composer package.
```

Figura 15-2: Comprobación del funcionamiento de composer

Realizado por: Yupangui María, 2019

3. Posterior a la instalación y verificación de Composer, se continua con la creación del proyecto. Para esto se abre la consola y mediante líneas de comando se ubica en la carpeta `xampp\htdocs`, que es el directorio en la cual se guardan los proyectos.
4. Para la creación del proyecto se ejecuta la línea de comando `composer create-project laravel/laravel sisventa 5.6.*`. En la **Figura 16-2** se observa la descarga de paquetes y librerías, además de la creación del esqueleto del proyecto.



```
C:\>cd xampp
C:\xampp>cd htdocs
C:\xampp\htdocs>composer create-project laravel/laravel sisventa 5.6.*
Installing laravel/laravel (v5.6.33)
- Installing laravel/laravel (v5.6.33): Loading from cache
Created project in sisventa
@php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 77 installs, 0 updates, 0 removals
- Installing symfony/polyfill-ctype (v1.11.0): Downloading (100%)
- Installing vlucas/phpdotenv (v2.6.1): Loading from cache
- Installing symfony/css-selector (v4.3.0): Downloading (100%)
- Installing tijsverkoyen/css-to-inline-styles (2.2.1): Loading from cache
- Installing symfony/polyfill-php72 (v1.11.0): Downloading (100%)
- Installing symfony/polyfill-mbstring (v1.11.0): Downloading (100%)
- Installing symfony/var-dumper (v4.3.0): Downloading (100%)
- Installing symfony/routing (v4.3.0): Downloading (100%)
- Installing symfony/process (v4.3.0): Downloading (100%)
- Installing psr/log (1.1.0): Loading from cache
- Installing symfony/polyfill-php73 (v1.11.0): Downloading (100%)
```

Figura 16-2: Creación del proyecto Laravel 5.6

Realizado por: Yupangui María, 2019

5. Para verificar que el proyecto haya sido creado correctamente nuevamente se ubica en la consola y mediante comando se ingresa a la carpeta `xampp/htdocs/sisventas` a continuación para levantar el servidor se ingresa la línea de comando `php artisan serve`.
6. En el navegador se debe ingresar a la dirección <http://localhost:8000/> ya que el proyecto se encuentra alojado en un servidor local XAMPP. En la **Figura 17-2** se visualiza la página de bienvenida del framework Laravel.

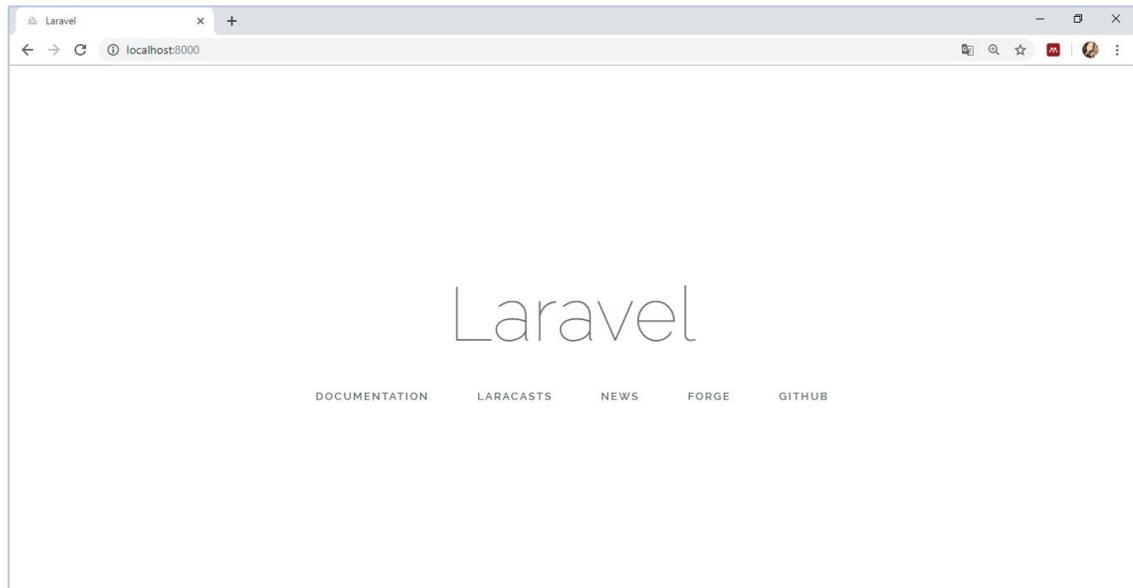


Figura 17-2: Página de bienvenida de Laravel
Realizado por: Yupangui María, 2019

En la **Figura 18-2** se observa la estructura carpetas y archivos del sistema de facturación, los cuales permitirán el correcto funcionamiento de la aplicación.

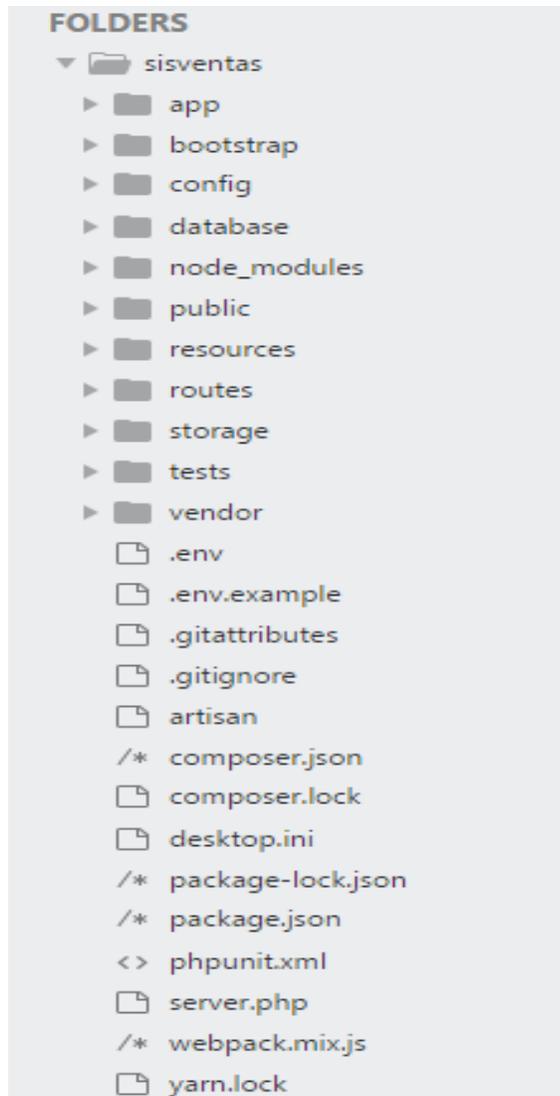


Figura 18-2: Directorio de archivos y carpetas del sistema

Realizado por: Yupangui María, 2019

En la carpeta Http/Controller se ubica todos los controladores del sistema, en la siguiente figura se observa el controlador VentaController, el cual muestra cada una de las opciones desarrolladas que va a tener la gestión de ventas.

The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure for 'sisventas' with subdirectories 'app', 'Console', 'Exceptions', 'Http', 'Controllers', 'Middleware', 'Requests', 'Notifications', 'Providers', and 'Rules'. The 'Controllers' directory is expanded, showing various controller files, with 'VentaController.php' selected. The code editor shows the following PHP code:

```

1 <?php
2 namespace App\Http\Controllers;
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\DB;
5 use Carbon\Carbon;
6 use App\Venta;
7 use App\DetalleVenta;
8 use App\User;
9 use App\Notifications\NotifyAdmin;
10
11 class VentaController extends Controller
12 {
13     public function index(Request $request)
14     {
15         if (!$request->ajax()) return redirect('/');
16
17         $buscar = $request->buscar;
18         $criterio = $request->criterio;
19
20         if ($buscar==''){
21             $ventas = Venta::join('personas','ventas.idcliente','=','personas.id')
22                 ->join('users','ventas.idusuario','=','users.id')
23                 ->select('ventas.id','ventas.tipo_comprobante','ventas.serie_comprobante',
24                     'ventas.num_comprobante','ventas.fecha_hora','ventas.impuesto','ventas.total',
25                     'ventas.estado','personas.nombre','users.usuario')
26                 ->orderBy('ventas.id', 'desc')->paginate(7);
27         }
28         else{
29             $ventas = Venta::join('personas','ventas.idcliente','=','personas.id')
30                 ->join('users','ventas.idusuario','=','users.id')
31                 ->select('ventas.id','ventas.tipo_comprobante','ventas.serie_comprobante',
32                     'ventas.num_comprobante','ventas.fecha_hora','ventas.impuesto','ventas.total',
33                     'ventas.estado','personas.nombre','users.usuario')
34                 ->where('ventas.'.$criterio, 'like', '%'. $buscar . '%')
35                 ->orderBy('ventas.id', 'desc')->paginate(7);
36         }
37
38         return [
39             'pagination' => [
40                 'total' => $ventas->total(),
41                 'current_page' => $ventas->currentPage(),
42                 'per_page' => $ventas->perPage(),
43                 'last_page' => $ventas->lastPage(),
44                 'from' => $ventas->firstItem(),
45

```

Figura 19-2: Controlador del sistema - VentaController

Realizado por: Yupangui María, 2019

En la raíz de la carpeta App se encuentran todos los modelos del sistema, estos modelos al momento de crear por defecto se guardan en esta ubicación, En la **Figura 20-2** se observa el modelo Venta, en este archivo además de especificar los atributos también se especifica la el tipo de relación con otras tablas.

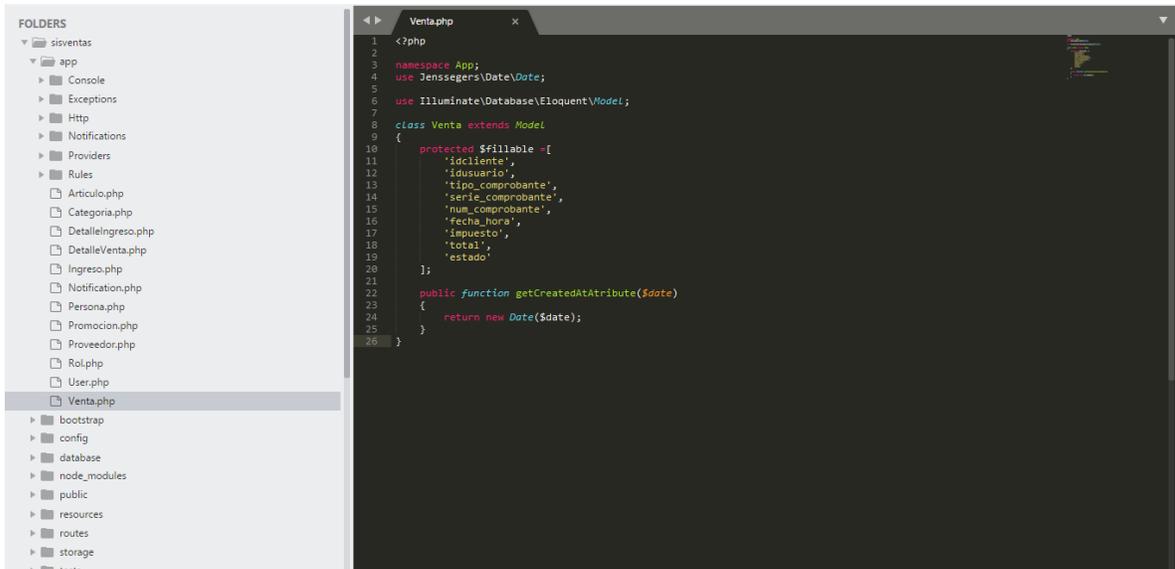


Figura 20-2: Modelo del sistema - Venta

Realizado por: Yupangui María, 2019

Laravel usa archivos de migraciones las cuales permitirá crear las tablas de la base de datos, cada uno de los modelos del sistema posee su respectivo archivo de migración los cuales se encuentran en la carpeta database/migrations. En la **Figura 21-2**, se observa el archivo de migración 2018_09_09_033145_create_ventas_table para la Venta.

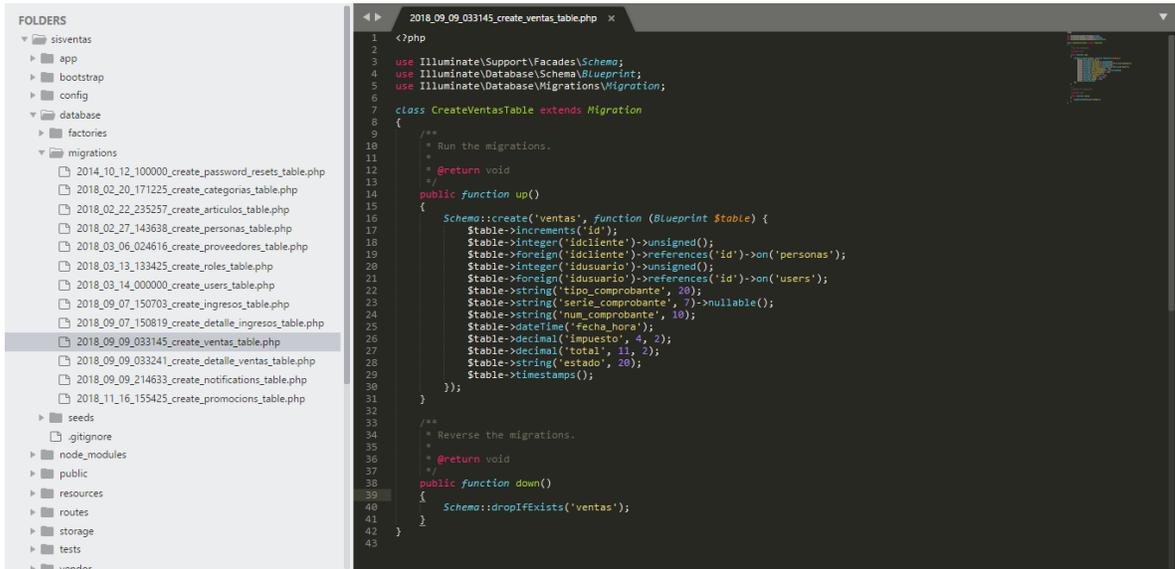
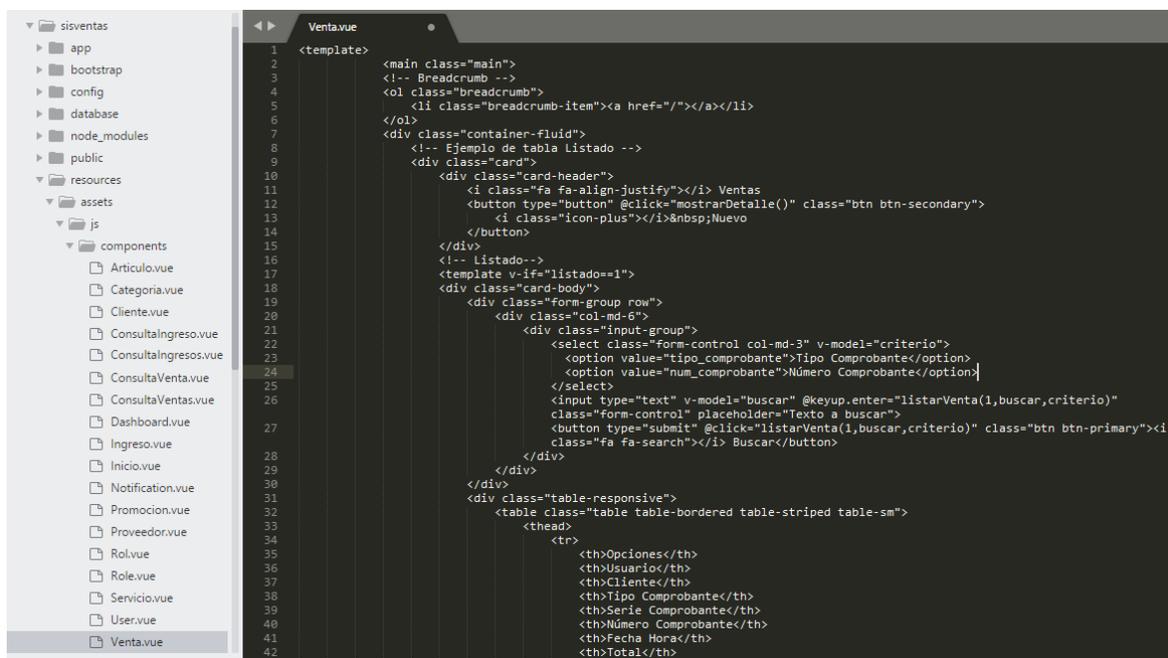


Figura 21-2: Archivo de migración - 2018_09_09_033145_create_ventas_table

Realizado por: Yupangui María, 2019

En la carpeta resource/assets/components se ubican las vistas o interfaces del sistema, en la **Figura 22-2**, se observa la vista de la gestión de Ventas.



The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure for 'sisventas' with subdirectories 'app', 'bootstrap', 'config', 'database', 'node_modules', 'public', 'resources', and 'assets'. The 'assets' directory contains a 'components' subdirectory with various Vue components like 'Articulo.vue', 'Categoria.vue', 'Cliente.vue', etc. The 'Venta.vue' component is selected. The code editor shows the following code:

```
1 <template>
2 <main class="main">
3 <!-- Breadcrumb -->
4 <ol class="breadcrumb">
5 <li class="breadcrumb-item"><a href="/"></a></li>
6 </ol>
7 <div class="container-fluid">
8 <!-- Ejemplo de tabla listado -->
9 <div class="card">
10 <div class="card-header">
11 <i class="fa fa-align-justify"></i> Ventas
12 <button type="button" @click="mostrarDetalle()" class="btn btn-secondary">
13 <i class="icon-plus"></i>&nbsp;&nbsp;&nbsp;Nuevo
14 </button>
15 </div>
16 <!-- Listado -->
17 <template v-if="listado=1">
18 <div class="card-body">
19 <div class="form-group row">
20 <div class="col-md-6">
21 <div class="input-group">
22 <select class="form-control col-md-3" v-model="criterio">
23 <option value="tipo_comprobante">Tipo Comprobante</option>
24 <option value="num_comprobante">Número Comprobante</option>
25 </select>
26 <input type="text" v-model="buscar" @keyup.enter="listarVenta(1,buscar,criterio)"
27 class="form-control" placeholder="Texto a buscar">
28 <button type="submit" @click="listarVenta(1,buscar,criterio)" class="btn btn-primary"><i
29 class="fa fa-search"></i> Buscar</button>
30 </div>
31 </div>
32 <div class="table-responsive">
33 <table class="table table-bordered table-striped table-sm">
34 <thead>
35 <tr>
36 <th>Opciones</th>
37 <th>Usuario</th>
38 <th>Cliente</th>
39 <th>Tipo Comprobante</th>
40 <th>Serie Comprobante</th>
41 <th>Número Comprobante</th>
42 <th>Fecha Hora</th>
43 <th>Total</th>
44 </tr>
45 </thead>
46 </table>
47 </div>
48 </div>
49 </div>
50 </div>
51 </main>
52 </template>
```

Figura 22-2: Vista del sistema - Venta
Realizado por: Yupangui María, 2019

Laravel 5.6 cuenta con un sistema de rutas el cual se encuentra en la carpeta routes/web.php, estas rutas además de establecer la URL, también definen el método con el cual va a realizar la petición estas pueden ser de tipo GET, POST, PUT. En la siguiente figura se muestra las rutas del sistema.

```

13 Auth::routes();
14 Route::get('/home', 'HomeController@index')->name('home');
15 Route::get('/', function () {
16     return view('welcome');
17 });
18 Route::get('/productos', 'ArticuloController@productos');
19 Route::get('/promociones', 'PromocionController@promociones');
20 Route::get('/contactos', function () {
21     return view('contactos');
22 });
23 Route::get('/quienes-somo', function () {
24     return view('quienes-somo');
25 });
26
27 Route::group(['middleware'=>['auth']],function(){
28
29     Route::post('/logout', 'Auth\LoginController@logout')->name('logout');
30     Route::get('/dashboard', 'DashboardController');
31     //Notificaciones
32     Route::post('/notification/get', 'NotificationController@get');
33
34     Route::get('/main', function () {
35         return view('contenido/contenido');
36     }->name('main');
37
38     Route::group(['middleware' => ['Almacenero']], function () {
39         Route::get('/categoria', 'CategoriaController@index');
40         Route::post('/categoria/registrar', 'CategoriaController@store');
41         Route::put('/categoria/actualizar', 'CategoriaController@update');
42         Route::put('/categoria/desactivar', 'CategoriaController@desactivar');

```

Figura 23-2: Rutas del sistema

Realizado por: Yupangui María, 2019

A continuación, se describe algunos de los comandos Artisan utilizados en el desarrollo del sistema, los cuales permitieron realizar el desarrollo de una manera más rápida.

- **php artisan serve:** Para levantar el servidor
- **php artisan app:name NombreNamespace):** Para cambiar el namespace de la aplicación.
- **php artisan migrate:** Para la creación de migraciones
- **php artisan make:auth:** Para brindar seguridad a la información de la empresa se generó el módulo de autenticación.
- **php artisan make: model NombreModelo:** Para generar un modelo
- **php artisan make:controller NombreControlador --resource):** Para generar un controlador
- **php artisan make:middleware NombreMiddleware:** Para crear un middleware.

Cabe recalcar que cada una de las gestiones que componen los módulos se realizaron de la misma manera descrita anteriormente, evidenciando el uso de la arquitectura MVC y el framework Laravel.

2.7. Plan de pruebas

Para la validación del software se ha elaborado un plan de pruebas para todo el proyecto mediante la utilización de la herramienta TestLink.

En la **Figura 24-2** se observa los planes de pruebas creados para el requerimiento de registro de una venta.

Suite de Pruebas : Venta	
Suite de Pruebas : HU-25: Como administrador, empleado, deseo registrar una venta	
Caso de Prueba SGP-100: Registrar venta - Flujo normal [Versión : 1]	
<u>Resumen:</u>	
Caso de prueba para validar el registro de una venta siguiendo un flujo normal.	
<u>Tipo de ejecución:</u>	Automatizada
<u>Duración estimada de la ejec. (min):</u>	0.35
<u>Prioridad:</u>	Media
Caso de Prueba SGP-101: Registrar venta - campos vacíos [Versión : 1]	
<u>Resumen:</u>	
Caso de prueba para validar el registro de una venta cuando no haya ingresado ningún dato en los campos.	
<u>Tipo de ejecución:</u>	Automatizada
<u>Duración estimada de la ejec. (min):</u>	0.15
<u>Prioridad:</u>	Media

Figura 24-2: Plan de pruebas - registro de ventas

Realizado por: Yupangui María, 2019

En la **Figura 25-2** se observa el detalle de cada uno de los casos de prueba creados para el requerimiento. En esta figura se observa un conjunto de pasos y los resultados esperados al realizar cada paso, además se puede visualizar el estado de la ejecución, prioridad en el negocio, el usuario encargado de realizar la prueba, la versión del sistema y el tiempo de ejecución del requerimiento.

Suite de Pruebas : HU-25: Como administrador, empleado, deseo registrar una venta			
Caso de Prueba SGP-100: Registrar venta - Flujo normal (Versión : 1)			
<u>Resumen:</u>			
Caso de prueba para validar el registro de una venta siguiendo un flujo normal.			
Nº	Pasos:	Resultados Esperados:	Estado de la ejecución:
1	El usuario debe dirigirse al menú de la parte izquierda, y desplegar el menú de "Ventas".	Debe desplegarse el menú de Ventas.	Pasado
2	Dar click en "Venta"	Debe visualizarse la interfaz de "Venta"	Pasado
3	Dar click en "Nuevo"	El sistema debe abrir una ventana de registro de ventas.	Pasado
4	En los campos ingresar la siguiente información: En el campo "Cliente" ingresar "0603059908" En el campo "Tipo Comprobante" seleccionar "FACTURA" Seleccionar los artículos a vender.		Pasado
5	Dar click en "Registrar Venta"	El sistema emite un mensaje de "Registrado La venta ha sido registrado con éxito."	Pasado
6	Dar click en "OK"	El sistema redireccionara a la interfaz de ventas	Pasado
<u>Tipo de ejecución:</u> Automatizada			
<u>Duración estimada de la ejec. (min):</u> 0.35			
<u>Prioridad:</u> Media			
Detalles de la ejecución			
<u>Build:</u> Versión 1.0			
<u>Tester:</u> admin			
<u>Resultado de la Ejecución:</u> Pasado			
<u>Modo de Ejecución:</u> Manual			
<u>Duración de la ejecución (min):</u>			

Figura 25-2: Detalle de casos de prueba - registro de ventas
Realizado por: Yupangui María, 2019

Para la validación de los requerimientos de todo el sistema se han elaborado 176 casos de prueba. El total de informes de plan de pruebas de las demás iteraciones se encuentran en el **Anexo C**.

2.8. Gestión del proyecto

2.8.6. *BurnDown Chart*

SCRUM utiliza el gráfico BurnDown Chart o gráfico de quemado, para la gestión del proyecto, este gráfico muestra el seguimiento del proyecto y da a conocer la velocidad con la que se efectúan los requerimientos y si se está cumpliendo en los tiempos planificados.

En el **Gráfico 1-2** se observa que el número de sprint se encuentra ubicados en el eje X, mientras que los puntos estimados se encuentran en el eje Y.

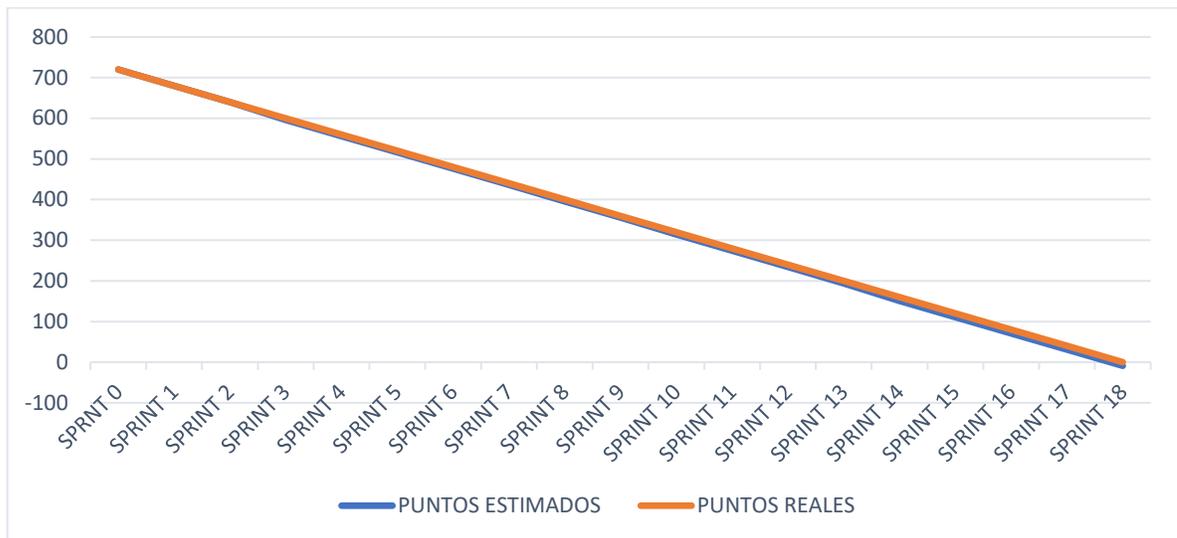


Gráfico 1-2: BurnDown Chart del proyecto

Realizado por: Yupangui María, 2019

Además, en el **Gráfico 1-2**, se visualiza que el tiempo real (línea naranja), varía con respecto al tiempo estimado (línea azul), esto se debe a que el desarrollo de las funcionalidades que se encuentran específicamente del sprint 3, 10 y 14 llevaron más tiempo desarrollar por falta de conocimiento del framework.

Con respecto al desarrollo de todo el proyecto se cumplió los tiempos estimados con los reales en la mayoría de los sprints.

2.9. Métodos utilizados para evaluar la eficiencia del sistema web

2.9.6. Estudio comparativo sistema manual vs. sistema automatizado

Para analizar y evaluar la eficiencia del sistema web se realiza un estudio comparativo de tiempos de respuesta en el proceso de facturación de los productos de la empresa denominada Lavadora y Lubricadora “Los Ángeles”. Los tiempos se medirán desde que un cliente llega a solicitar su factura hasta que reciba el documento, el estudio comparativo se realizara midiendo el tiempo de un sistema manual vs. un sistema automatizado.

2.9.6.1. Procesos a evaluar la eficiencia

Para evaluar la eficiencia se tomó en cuenta la duración de tiempo del proceso antes de desarrollar el sistema web y el tiempo después de desarrollar el sistema web. El proceso a evaluar la eficiencia es el siguiente:

Facturación: Este proceso consiste en registrar una venta, por medio de esta interfaz se seleccionará un cliente y los artículos solicitados, al mismo tiempo generar su factura. Este proceso involucra otros subprocesos los cuales se detallan a continuación.

- Registro de compras
- Registro del artículo

CAPÍTULO III

3. MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS RESULTADOS

En este capítulo se realiza un análisis para medir la eficiencia del sistema en cuanto al comportamiento del tiempo en el proceso de facturación, usando una distribución estadística la cual ayudara a establecer los resultados y permitirán determinar si el sistema cumple con las expectativas y los objetivos planteados anteriormente previo al desarrollo del sistema.

3.4. Diseño del pre experimento

De acuerdo a la norma ISO/IEC 9126 la cual es usado para la evaluar la calidad del software, este estándar detalla 6 características la Funcionalidad, Fiabilidad, Usabilidad, Eficiencia, Mantenibilidad, Portabilidad, para este estudio se analizará la Eficiencia, dicha característica basa su criterio en el comportamiento de tiempos y utilización de recursos. Por lo tanto, al medir tiempos de respuesta de un proceso este es un valor cuantificable, se determinó comparar el tiempo promedio entre el sistema manual y el sistema automatizado.

Para el experimento se analiza el proceso de generar una factura, este a su vez utiliza otros subprocesos como: el registro de una compra y artículos.

La población determinada para el experimento fue de 73 datos recogidos durante un periodo de 15 días de trabajo, para el análisis se aplicó el muestreo por conveniencia, donde la población es igual a la muestra considerando 73 datos recogidos durante 15 días. La eficiencia del sistema se evaluó mediante la relación de tiempos de respuesta, para esto se tomó dos conjuntos de datos que reflejan los tiempos en segundos en que se demora el proceso. El análisis se realizó mediante la utilización de la prueba estadística T-student para muestras relacionadas la cual se utiliza cuando se tiene una población normalmente distribuida y se determinó haciendo uso del test estadístico de Kolmogorov Smirnov.

3.5. Desarrollo del pre experimento

Los conjuntos de datos tomados mediante la técnica utilizada para comparar la diferencia de tiempos del proceso de facturación del sistema manual y el sistema automatizado se detallan en la **Tabla 1-3** y representados en el **Gráfico 1-3** que se visualizan a continuación.

Tabla 1-3: Muestra para el pre experimento

Número de datos	Tiempo sistema manual (seg)	Tiempo sistema automatizado (seg)
1	252	126
2	250	130
3	248	128
4	250	130
5	260	131
6	242	128
7	247	129
8	250	126
9	240	132
10	248	128
11	241	129
12	253	131
13	244	126
14	252	129
15	249	131
16	248	132
17	247	127
18	251	130
19	249	129
20	241	130
21	249	128
22	252	130
23	250	127
24	249	128
25	252	129
26	252	130
27	248	126
28	255	127
29	247	129

30	250	126
31	247	126
32	248	127
33	249	130
34	252	128
35	247	127
36	250	130
37	248	129
38	245	126
39	252	130
40	262	128
41	252	129
42	249	130
43	251	127
44	247	128
45	250	127
46	246	125
47	245	128
48	252	129
49	252	130
50	247	128
51	240	130
52	246	131
53	248	130
54	246	129
55	250	130
56	248	128
57	246	131
58	251	129
59	249	130
60	241	129
61	250	130
62	249	129
63	251	130
64	247	131
65	241	129
66	248	128
67	250	130
68	249	127
69	252	129
70	248	127

71	251	129
72	249	131
73	252	132
Media	248,753	128,808
Varianza	14,966	2,740

Realizado por: Yupangui María, 2019

Para comprobar que los datos de la muestra sigan una distribución normal se utiliza el test estadístico de Kolmogorov Smirnov, el cual se utiliza cuando se tiene una muestra mayor a 30 datos y se obtiene los siguientes resultados.



Figura 1-3: Test estadístico Kolmogorov Smirnov
Realizado por: Yupangui María, 2019

Por medio del test estadístico Kolmogorov Smirnov se determina el siguiente criterio el cual permitirá conocer si los datos siguen una distribución Normal.

Criterio para determinar la Normalidad

P-valor $\geq \alpha$ Aceptar H_0 = Los datos provienen de una distribución Normal

P-valor $< \alpha$ Aceptar H_1 = Los datos no provienen de una distribución Normal

Porcentaje de error o nivel de significancia $\alpha = 0.05 = 5\%$

Intervalo de confianza = 95%

Como P-valor=0.0694 es mayor que $\alpha=0.05$ entonces se rechaza la hipótesis alternativa, por lo cual se puede afirmar que los datos provienen de una población normalmente distribuida.

3.6. Análisis de prueba T-Student para muestras relacionadas

3.6.6. Planteamiento de hipótesis

Hipótesis nula (H_0). – El tiempo promedio del sistema manual es menor o igual al tiempo promedio del sistema automatizado para el proceso de solicitar una factura.

Hipótesis alternativa (H_1). – El tiempo promedio del sistema manual es mayor al tiempo promedio del sistema automatizado para el proceso de solicitar una factura.

3.6.7. Nivel de significancia

Se ha determinado el nivel de significancia de $\alpha = 0.05$ (5% de error) ya que es el más utilizado y permitirá obtener un nivel de confianza del 95%.

3.6.8. Estadístico de prueba

Se utilizó la prueba estadística T-Student para muestras relacionadas, ya que los dos conjuntos de datos provienen de una distribución normal lo cual se analizó mediante el test de Kolmogorov Smirnov. A continuación, en la Tabla 2-3 se visualiza los resultados obtenidos mediante Microsoft Excel.

Tabla 2-3: Prueba estadística T-Student para medias de dos muestras relacionadas

Prueba estadística T Student	
Grados de libertad	72
Estadístico t	247,767
P(T<=t) una cola	0,000

Realizado por: Yupangui María, 2019

3.6.9. Regla de decisión

Mediante la campana de Gauss se puede observar los resultados emitidos para la muestra T-Student para muestras relacionadas, por lo cual se visualiza la zona de aceptación y rechazo.

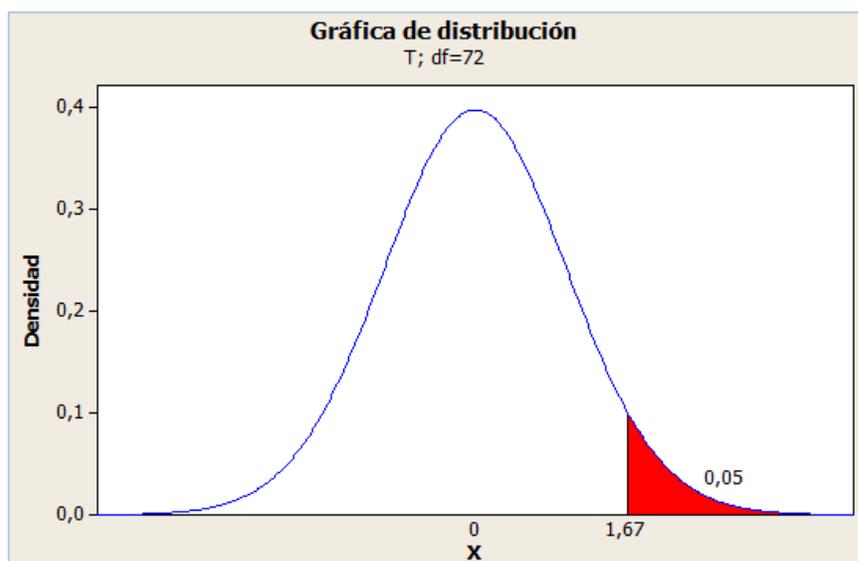


Gráfico 1-3: Análisis de datos mediante la Campana de Gauss
Realizado por: Yupangui María, 2019

En base a los datos de la **Tabla 3-3** los cuales son utilizados en la toma de decisión y se determinara si se acepta o se rechaza la hipótesis nula. Se puede observar los valores de T tratado=1.67; el T calculado=247.767; P-valor=0.000 y el nivel de significancia $\alpha=0.05$

Tabla 3-3: Regla de decisión

Regla de decisión		
Tc	>	Tt
247.767	>	1.66
P-valor	<	A
0.000	<	0.05

Realizado por: Yupangui María, 2019

3.6.10. Toma de decisión

Dado que el valor $t=247.767$ es mayor que el intervalo de confianza (1.66) y P-valor=0.000 es menor que el nivel de significancia $\alpha=0.05$ ya que existe diferencias estadísticamente significativas por lo

tanto se concluye que se rechaza la hipótesis nula, es decir El tiempo promedio del sistema manual es mayor al tiempo promedio del sistema automatizado para el proceso de solicitar una factura.

3.7. Análisis de medias

En el **Gráfico 1-3**, se observa la diferencia de tiempos de respuesta entre el sistema manual y el sistema automatizado.

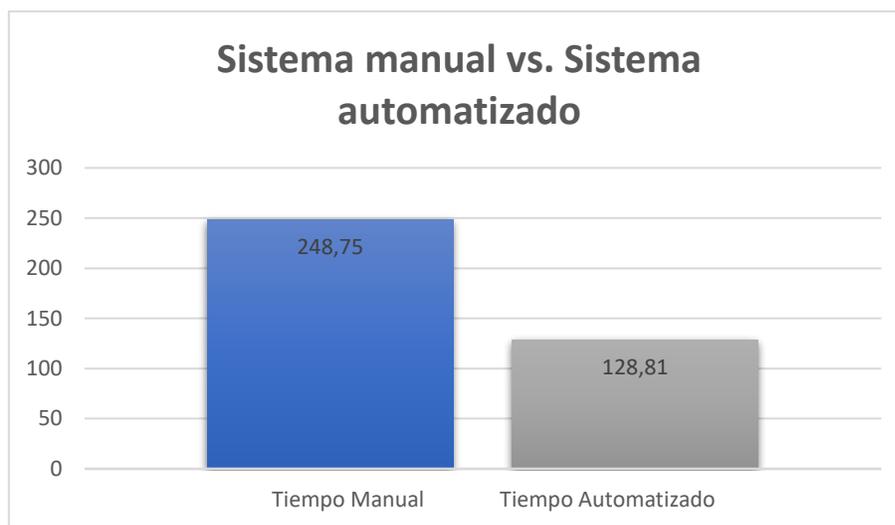


Gráfico 2-3: Sistema manual vs. Sistema automatizado
Realizado por: Yupangui María, 2019

Luego de realizar el análisis de las medias se obtuvo como resultado que el tiempo automatizado es menor en un 51.78% al sistema manual.

CONCLUSIONES

- Mediante las reuniones y entrevistas realizadas al gerente general quien es el encargado del manejo administrativo se determinó el proceso de facturación que realiza la Lavadora y Lubricadora “Los Ángeles”, el mismo que fue plasmado mediante un diagrama de procesos usando la herramienta Bizagi Modeler.
- Se recopiló los requerimientos funcionales, dando un total de 8 requerimientos del sistema los cuales se encuentran detallados en el apartado de Especificación de requerimientos. Además, se diseñó la base de datos del sistema dando un total de 14 tablas los cuales permitirán el correcto almacenamiento y funcionamiento del sistema. Mediante un estudio comparativo se determinó que la arquitectura Modelo - Vista – Controlador es la más adecuada ya que mantiene separada la lógica de negocio del diseño y facilita la mantenibilidad del sistema, permitiendo así que el sistema sea escalable, además cabe recalcar que es una característica del framework Laravel.
- En el desarrollo de los módulos se aplicó el framework Laravel reduciendo tiempo de desarrollo y costos para la empresa mediante el uso de sus características como: sistema de autenticación, rutas, ORM, modelos, migraciones, plantillas blade, controladores, líneas de comando Artisan los cuales se implementó en todos los módulos que componen el sistema. Para la ejecución del proyecto se realizó siguiendo la metodología ágil SCRUM en el cual se obtuvieron 53 historias de usuario y 12 historias técnicas cada una de ellas se agregaron a cada uno de los sprints dando un total de 18 sprints, además se usó de gráfico BurnDown Chart el cual indica el seguimiento del proyecto dando un aumento en el tiempo estimado de desarrollo del sprint 3, 10 y 14 por desconocimiento del framework. El proyecto tuvo una duración de 90 días con una jornada de trabajo de 8 horas de diarias por 5 días laborables, es decir con un total de 720 horas de desarrollo, cabe recalcar que no se necesitó realizar ninguna replanificación, cumpliendo así con la planificación establecida al inicio del proyecto.
- Para evaluar la eficiencia se apoyó en el estándar ISO/IEC 9126, el experimento se realizó midiendo los tiempos de respuesta del sistema manual y el sistema automatizado, obteniendo resultados significativos al comparar los tiempos de respuesta, por lo cual se concluye que el sistema automatizado es más eficiente en el proceso de facturación en un 51.78%.

RECOMENDACIONES

- La metodología ágil SCRUM es recomendable para el desarrollo de proyectos software, ya que permite la integración de todas las personas involucradas, además permite la integración de nuevas funcionalidades al sistema y las entregas continuas permiten verificar que las funcionalidades se estén cumpliendo de acuerdo a lo especificado por el Product Owner.
- La aplicación del framework Laravel permitió el desarrollo del sistema de una manera rápida y eficiente, ya que dispone de una gran cantidad de librerías, posee un sistema de ruteo fácil de manejar, sus consultas lo realizan mediante ORM y cuenta con un sistema de autenticación simple, además tiene una documentación amplia y una comunidad que apoya el proyecto y que cada día sigue creciendo.
- Para el funcionamiento correcto del sistema se recomienda utilizar navegadores compatibles con las vistas de la aplicación ya que fueron desarrollados con el framework Bootstrap y debido a esto existe interferencia en interpretar el framework con versiones de Internet Explorer; los navegadores recomendados son Google Chrome, Mozilla.
- Para futuras versiones del sistema se recomienda realizar un módulo de sistema de pagos en línea.

BIBLIOGRAFÍA

ACOSTA CONS, A.L., COTA SERVANTES, V., MARTÍNEZ MARTÍNEZ, J., *Características de MySQL* [en línea]. 02 Octubre, 2013. [Consulta: 03 mayo 2019]. Disponible en: <https://es.slideshare.net/Wario1025/my-sql-26811089>

ALEGSA, L., *Definición de AJAX* [en línea]. 05 diciembre, 2010. [Consulta: 03 abril, 2019]. Disponible en: <http://www.alegsa.com.ar/Dic/ajax.php>

ALMEYDA SIFUENTES, M., *Arquitectura de Software: ¿Qué es, y cómo funciona?* [blog]. 05 junio, 2019. [Consulta: 05 junio, 2019]. Disponible en: <http://blog.continental.edu.pe/sistemas-informatica/2013/01/10/arquitectura-de-software-que-es-y-como-funciona/>

ÁLVAREZ, M.A., *Qué es MVC* [en línea]. 02 enero, 2014. [Consulta: 01 mayo, 2019]. Disponible en: <https://desarrolloweb.com/articulos/que-es-mvc.html>

APIUMHUB., *Arquitectura de software y sus beneficios* [en línea]. 15 agosto, 2017. [Consulta: 03 mayo, 2019]. Disponible en: <https://apiumhub.com/es/tech-blog-barcelona/arquitectura-de-software/>

ARCE, A. *Programación PHP* [en línea]. [Consulta: 09 enero, 2019]. 2018. Disponible en: <https://buildmedia.readthedocs.org/media/pdf/programacion-php/latest/programacion-php.pdf>

AUBRY, C., *HTML5 y CSS3* [en línea]. Dept. Ciencia de la Computación, 2011, pp.2. [Consulta: 12 febrero, 2019]. Disponible en: <http://www.jtech.ua.es/dadm/2011-2012/restringido/web/sesion02-apuntes.pdf>

BARBARAPVN, L., *Entendiendo HTML5: guía para principiantes* [en línea]. 28 mayo, 2013. [Consulta: 03 mayo, 2019]. Disponible en: <https://hipertextual.com/archivo/2013/05/entendiendo-html5-guia-para-principiantes/>

BAQUERO GARCÍA, J.M., *¿Qué es Laravel?* [blog]. 11 diciembre, 2015. [Consulta: 03 mayo, 2019]. Disponible en: <https://www.arsys.es/blog/programacion/que-es-laravel/>

BEMBIBRE, V., *Definición de MySQL* [en línea]. Febrero, 2009. [Consulta: 13 enero, 2019]. Disponible en: <https://www.definicionabc.com/tecnologia/mysql.php>

BORBÓN ARDILLA., *Norma de evaluación ISO/IEC 9126* [en línea]. 12 marzo, 2013. [Consulta: 09 marzo, 2019]. Disponible en: <http://actividadreconocimiento-301569-8.blogspot.com/2013/03/norma-de-evaluacion-isoiec-9126.html>

CANO PITA, G., Las TICs en las empresas: Evolución de la tecnología y cambio estructural en las organizaciones. *Dominio de las Ciencias* [en línea], vol. 4, no. 1, pp. 499-510. [Consulta: 03 mayo, 2019]. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=6313252>.

CHAVARRIA NEIRA, B. y GUDIÑO DE LA A, E., *Implementación de un servidor web y un diseño de una página utilizando herramientas de software libre para el dispensario “Sagrada Familia” de la ciudad de Guayaquil.* [en línea] (Tesis) (pregrado). Universidad Politecnica Salesiana, Ecuador, Guayaquil. 2017. pp. 43 [Consulta: 14 mayo 2019]. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/14162/1/GT001840.pdf>.

CARITA, G. *Tutorial de Creación de Gráficos con Chart.js.* [en línea]. 12 marzo, 2018. [Consulta: 12 febrero, 2019]. Disponible en: <http://gidahatari.com/ih-es/tutorial-de-creacion-de-graficos-con-chartjs>

COBO, Á., GÓMEZ, P., PÉREZ, D. y ROCHA, R., *PHP y MySQL Tecnologías para el desarrollo de aplicaciones web* [en línea]. España: Ediciones Díaz de Santos, 2005. [Consulta: 03 mayo, 2019]. Disponible en: <http://www.editdiazdesantos.com/wwwdat/pdf/9788479787066.pdf>

COMAS, S., *Layouts en Blade* [en línea]. 04 febrero, 2014. [Consulta: 03 febrero, 2019]. Disponible en: <https://ayudalaravel.com/layouts-en-blade/>

CURSO, S., *CSS Y BOOTSTRAP 3* [en línea]. Sopra, 2016, pp. 48. [Consulta: 12 enero, 2019]. Disponible en: <http://www.islavisual.com/tutoriales/curso-de-maquetacion-css-y-bootstrap.pdf>

DÍAZ, J.M., SAMPEDRO, L. y VARGAS, F., *Instalación y configuración de Apache, un servidor Web gratis.* Ingeniería y Desarrollo, 2002, vol. 12, no. 12, pp. 10-23.

EGUÍLUZ, J., *Introducción a Css.* [en línea]. [Www.Librosweb.Es](http://www.Librosweb.Es), 2008, pp. 5. [Consulta: 20 enero, 2019]. Disponible en: http://www.jesusda.com/docs/ebooks/introduccion_css.pdf.

EJIE, S.A., *Manual de uso. Testlink* [en línea]. S.l.:2013. [Consulta: 23 febrero 2019]. Disponible en: www.ejie.es.

FERRER, J., *Conceptos generales de la arquitectura de aplicaciones web* [en línea]. Implantación

de aplicaciones web, 2012. pp. 11. [Consulta: 07 febrero, 2019]. Disponible en: <http://www.rama.es/descargas/descargar.php?fichero=Z3dkZXNjYXJnYXNwcm9mIzIzOSM5Nzg4NDk5NjQxNjA3X0NhG10dWxvIDEucGRm>

FLUJO, C. De., *Lenguajes de Programación* [en línea]. España: Creative Commons, 2018. no. 1563, pp. 1-57. [Consulta: 20 enero, 2019]. Disponible en: https://guimi.net/descargas/Monograficos/G-Lenguajes_de_programacion.pdf

HAMIDIAN FERNÁNDEZ, B.F. y OSPINO SUMOZA, G.R. *¿Por qué los sistemas de información son esenciales?* [en línea]. Anuario vol. 38, no. 2011, pp. 161-183. ISSN 1316-5852. Disponible en: <http://servicio.bc.uc.edu.ve/derecho/revista/idc38/art07.pdf>.

HOSTALIA., *Laravel, un framework de PHP* [en línea]. S.l.:HostaliaWhitepapers, 2016. pp. 3. [Consulta 12 abril, 2019]. Disponible en: https://pressroom.hostalia.com/wp-content/themes/hostalia_pressroom/images/framework-laravel-wp-hostalia.pdf

LECHON, S., *Administración y Gestión de un Servidor Web Apache* [en línea], pp. 24. [Consulta: 12 febrero, 2019]. Disponible en: <https://radiosyculturalibre.com.ar/biblioteca/SERVIDORES/Web/apache.pdf>.

LÓPEZ TORRALBA, M.A., *Estructura de laravel* [en línea]. 25 abril, 2018. [Consulta: 11 marzo, 2018]. Disponible en: <http://mialtoweb.es/estructura-de-laravel/>

MACEDO, D., *El impacto de PHP 7 y las principales razones para migrar* [blog]. 06 abril, 2018. [Consulta: 10 enero, 2019]. Disponible en: 2018 <https://www.workana.com/blog/freelanceando/php7-razones-migrar/>

MARTIN ALAIMO, M.S., *Proyectos Ágiles con Scrum: Flexibilidad, aprendizaje, innovación y colaboración en contextos complejos.* [en línea]. S.l.: s.n. ISBN 9874576340, 2015. [Consulta: 03 mayo, 2019]. Disponible en: https://www.amazon.es/gp/product/9874576340/ref=as_li_ss_tl?ie=UTF8&linkCode=sl1&tag=laofi cdeproyd-21&linkId=0cd380637f52c53cd41a4ac796d91208.

MCGLONE, J., *Instalacion y configuracion de un servidor web HTML & PHP* [en línea]. Asuntos a Tratar, 2008, pp: 3. [Consulta: 03 mayo, 2019]. Disponible en: <http://www2.elo.utfsm.cl/~iwg101/ClaseWeb.pdf>

MEDINA PRIETO, R., *Modelos y Estándares de Calidad Aplicados al Sistema de Información* [en línea]. 24 julio, 2017. Pp: [Consulta: 11 enero, 2019] Disponible en: <http://unidad4rociomp.blogspot.com/2017/07/46.html>

MENÉNDEZ, R., BARZANALLANA, A., *Desarrollo de Aplicaciones Web Xampp* [en línea]. 07 marzo, 2018. [Consulta: 03 mayo, 2019]. Disponible en: <https://www.um.es/docencia/barzana/DAWEB/Desarrollo-de-aplicaciones-web-Xampp.html>

PÉREZ GARCÍA, Alejandro Alfonso, *Desarrollo de herramientas web de gestión docente* [en línea] (Tesis) (pregrado). Universidad Politecnica de Cartagena, Cartagena, Colombia. 2007, pp. 13. [Consulta: 23 marzo, 2019]. Disponible en: <http://repositorio.upct.es/bitstream/handle/10317/179/pfc2475.pdf>

MORA, S.L., *Programación de aplicaciones web: historia, principios básicos y clientes web*. [en línea]. Alicante, 2002, pp. 3. [Consulta: 1 julio 2018]. Disponible en: <https://gplsi.dlsi.ua.es/~slujan/materiales/pi-cliente2-muestra.pdf>.

MOZAMMEL, H., *Finding laravel from a lost advanced beginner of java* [en línea]. 06 diciembre, 2013. [Consulta: 03 mayo, 2019]. Disponible en: <https://www.slideshare.net/mozammel8435/finding-laravel-from-a-lost-advanced-beginner-of-java>

NAVARRETE, T., *El lenguaje JavaScript* [en línea]. Fonaments Cartografia i SIG, 2007, pp. 11. [Consulta: 12 febrero, 2019]. Disponible en: <http://www.dtic.upf.edu/~tnavarrete/fcsig/javascript.pdf>.

ORTIZ, A.E., *Qué es un framework en informática o programación* [blog]. 05 abril, 2018. [Consulta: 12 marzo, 2019]. <https://blog.hostdime.com.co/que-es-un-framework-informatica-programacion/>

PEINADO, F. (2010). *LPS: Patrón Modelo-Vista-Controlador* [en línea]. S.l., 2010, pp. 4. [Consulta: 28 marzo, 2019]. Disponible en: <http://web.fdi.ucm.es/profesor/fpeinado/courses/oop/LPS-14ModeloVistaControlador.pdf>.

PEÑO SÁNCHEZ, M.J., *Escuela Técnica Superior de Ingeniería y Pruebas de Software. Fundamentos y técnicas*. [en línea]. S.l., 2015, pp. 109. [Consulta: 23 febrero 2019]. Disponible en: http://oa.upm.es/40012/1/PFC_JOSE_MANUEL_SANCHEZ_PENO_3.pdf.

RANGEL, J., *Que es y para que sirve CSS3?* [en línea]. 05 mayo, 2012. [Consulta: 13 enero, 2019].

Disponible en: <http://tc2-proyectodegradoingdesistemas.blogspot.com/2012/05/que-es-y-para-que-sirve-css3.html>

PERALTA, A. (2003). *Metodología scrum*. [en línea]. Universidad ORT Uruguay, 2003, pp. 3-9. [Consulta: 22 febrero, 2019]. Disponible en: <https://fi.ort.edu.uy/innovaportal/file/2021/1/scrum.pdf>.

PÉREZ GARCÍA, Alejandro Alfonso, *Desarrollo de herramientas web de gestión docente* [en línea] (Tesis) (pregrado). Universidad Politecnica de Cartagena, Cartagena, Colombia. 2007, pp. 13. [Consulta: 23 marzo, 2019]. Disponible en: <http://repositorio.upet.es/bitstream/handle/10317/179/pfc2475.pdf>

PEREZ, J.E., Introducción ajax. [en línea]. S,l: ,2008, pp. 5. ISSN 2231-4245, 2231-4245. [Consulta: 21 febrero, 2019]. Disponible en: <http://www.youblisher.com/p/81953-introduccion-a-ajax/>

PÉREZ VALDÉS, D., *¿Qué es Javascript?* [en línea]. 03 julio, 2007. [Consulta: 20 marzo, 2019]. Disponible en: <http://www.maestrosdelweb.com/que-es-javascript/>

RIVERA, L.A.. *Arquitectura del Software* [en línea]. S,l: ,2008, pp. 11-15. ISSN 1873-1244. DOI 10.1016/j.nut.2008.11.025. [Consulta: 13 enero, 2009]. Disponible en: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_1_a/capitulo2.pdf

RODRÍGUEZ, R.M., *Generación de informes de proyectos ágiles en TestLink* [en línea]. 05 junio, 2013. [Consulta: 25 febrero, 2019]. Diponible en: <https://www.adictosaltrabajo.com/2013/06/05/trabajando-con-test-link/>

RONDÓN, C., *Artisan, la interfaz de línea de comandos de Laravel* [en línea]. 12 febrero, 2016. [Consulta: 16 enero, 2019]. Diponible en: <https://styde.net/artisan-interfaz-linea-comandos-de-laravel/>

ROUSE, M., *Guía Esencial: Las bases de datos dan soporte a las tendencias de TI* [en línea]. Enero, 2015. [Consulta: 20 enero, 2019]. Disponible en: <https://searchdatacenter.techtarget.com/es/definicion/Base-de-datos>

SÁNCHEZ PEDRÓS, Jaime, *Desarrollo de un sistema web para la gestión de imágenes y álbumes* [en línea] (Tesis) (pregrado). Universidad Politécnica de Madrid, Madrid, España. 2017 pp. 6-7. [Consulta: 10 enero, 2019]. Disponible en: http://www.directorioidigital.com/empresas/agfoliares/PROYECTO_ZEOLITA_2006.pdf.

SÁNCHEZ SOTO, E. A., RODRÍGUEZ RIVERO, M.G., *Pregunta monvadora ¿ Qué es un navegador Web y para qué sirve* [en línea]. S.l.: , 2016, pp. 9. [Consulta: 22 marzo, 2019]. Disponible en:

https://repository.uaeh.edu.mx/bitstream/bitstream/handle/123456789/16662/PE_T2_U1_Navegadores.pdf?sequence=1&isAllowed=y

SÁNCHEZ VELAMAZÁN, J. *Diseño de un inventario web de productos químicos para la facultad de química* [en línea] (Tesis) (pregrado). Universidad de Barcelona, Barcelona, España. 2015, pp. 101. [Consulta: 10 enero, 2019]. Disponible en: <http://diposit.ub.edu/dspace/bitstream/2445/67906/2/memoria.pdf>

SARMIENTO, J., *Norma ISO / IEC 9126* [en línea]. S.l.: , 2010. pp 6-38. [Consulta: 03 mayo, 2019]. Disponible en: https://www.academia.edu/20851916/Norma_ISO_IEC_9126_2001_Inicio

SCHWABER, K. y SUTHERLAND, J., *La Guía definitiva de Scrum: Las Reglas de Juego.* [en línea]. Scrum.org, 2014, pp. 4-6. [Consulta: 10 febrero, 2019]. Disponible en: Scrum.org.

SEVILLA, (2004). *Introducción a las Aplicaciones Web* [en línea]. S.l.: , 2004, pp. 2. [Consulta: 11 septiembre 2018]. Disponible en: <http://www.lsi.us.es/docencia/get.php?id=854>.

SHOKEEN, M., *Comenzando Con Chart.js: Introducción* [en línea]. 18 abril, 2017. [Consulta: 12 febrero, 2019]. Disponible en: <https://code.tutsplus.com/es/tutorials/getting-started-with-chartjs-introduction--cms-28278>

SIERRA, K., *Qué es Laravel? Ventajas del desarrollo a medida para tus proyectos* [blog]. 31 enero, 2018. [Consulta: 13 abril, 2019]. Disponible en: <https://www.synergyweb.es/blog/laravel-desarrollo-medida/>

TERRERA, G., *TestLink – AutoAprendizaje – Parte I.* [en línea]. 10 abril, 2013. [Consulta: 12 febrero, 2019]. Diponible en: <https://testingbaires.com/2013/04/10/testlink-autoaprendizaje-parte-i/>

VALLE, J.G. y GUTIERREZ, J.G., *Definición arquitectura cliente servidor* [en línea]. Tecnología en informática, S.l.: , 2005, pp. 3. [Consulta: 25 febrero, 2019]. Disponible en: https://www.ecotec.edu.ec/documentacion/investigaciones/docentes_y_directivos/articulos/5743_TRECALDE_00212.pdf

VELÁSQUEZ VALLE, L.R. y ZELEDÓN BUSTILLO, C.A., *Sistemas De Inventario Y Facturacion De La Tienda De Accesorios De Computadoras Y Celulares “Decosys”*. [en línea] (Tesis) (Pregrado). Consejo Superior Univrsitario Centroamericano, S,l: , 2014, pp. 101. [Consulta: 11 febrero, 2019]. Disponible en: <http://repositorio.unan.edu.ni/1955/1/16393.pdf>.

VERGARA SANDOVAL, Marco Patricio, *Problemática en la implementación de facturación electrónica en el Ecuador* (tesis) (Maestria). Universidad Andina Simón Bolívar, Quito, Ecuador, 2017. pp. 11 [Consulta: 02 marzo, 2019]. Disponible en: <http://repositorio.uasb.edu.ec/bitstream/10644/6031/1/T2523-MT-Vergara-Problematica.pdf>.

VILLASANTE, L., *Desarrollo Web con Sublime Text* [en línea]. 21 febrero, 2014. [Consulta: 22 febrero, 2019]. Disponible en: <https://enboliviacom.wordpress.com/2014/02/21/desarrollo-web-con-sublime-text/>

YANQUI, F.E.T., “Metodología SCRUM para el desarrollo de software y gestión de proyectos en las pequeñas y medianas empresas de la ciudad de Juliaca.” *Investigación Andina* [en línea] (Tesis). (pregrado), vol. 13, no. 1, pp. 88. [Consulta: 02 febrero 2019]. ISSN 1994-8077. Disponible en: <http://repositorio.uancv.edu.pe/bitstream/handle/UANCV/2675/VOL13N1%282014%29%2011%20%2888-95%29.pdf?sequence=1&isAllowed=y>

ZEND TECHNOLOGIES., *PHP7: más velocidad de carga y acceso en Internet* [en línea]. 08 noviembre, 2018. [Consulta: 03 mayo, 2019]. Disponible en: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/caracteristicas-y-ventajas-de-php7/>