

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**DESARROLLO DE UN SISTEMA PROTOTIPO PARA REALIZAR
FACTURACIÓN, GESTIÓN DE MERCADERÍA Y COMPRA EN
LÍNEA PARA EMPRESAS DE DISTRIBUCIÓN DE TÓNERS Y
SUMINISTROS DE OFICINA**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
“INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN”**

GALARZA PAZMIÑO DIEGO ENRIQUE
diego.galarza@epn.edu.ec

DIRECTOR: MSc. XAVIER CALDERÓN HINOJOSA
xavier.calderon@epn.edu.ec

Quito, Julio 2019

AVAL

Certifico que el presente trabajo fue desarrollado por Galarza Pazmiño Diego Enrique, bajo mi supervisión.

MSC. XAVIER CALDERÓN
DIRECTOR DEL PROYECTO

DECLARACIÓN DE AUTORÍA

Yo Diego Enrique Galarza Pazmiño declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley. Reglamentos y Normas vigentes.

Galarza Diego

AGRADECIMIENTO

Al Magister Xavier Calderón que con entusiasmo y sabiduría, me guió como profesor y director, brindándome el apoyo necesario con su larga experiencia y conocimientos, para el desarrollo de este Trabajo de Titulación.

Diego Galarza

AGRADECIMIENTO

Me gustaría agradecer la ayuda que muchas personas y colegas me han prestado durante el proceso de desarrollo de este proyecto. Agradezco a mis padres y a mi hermana que me han ayudado y apoyado en todo el transcurso de mi vida.

A mis amigos y amigas, que mostraron de una manera sincera su colaboración y buena voluntad en todos estos años. Al Ingeniero Jefferson Benavides, por toda la ayuda prestada para la realización de este Trabajo de titulación.

A la Escuela Politécnica Nacional por ser la sede de todo el conocimiento y experiencias adquiridos en estos años.

Diego Galarza

DEDICATORIA

Dedico este proyecto a mis padres por su apoyo incondicional tanto económico como moral, en cada etapa de mi vida.

Diego Galarza

CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA	II
AGRADECIMIENTO	III
AGRADECIMIENTO	IV
DEDICATORIA	V
ÍNDICE DE CONTENIDO.....	VI
RESUMEN	XIII
ABSTRACT.....	XIV
1. INTRODUCCIÓN	1
1.1 OBJETIVOS.....	1
1.2 ALCANCE.....	2
1.3 MARCO TEÓRICO.....	4
1.3.1 OOHDM.....	4
1.3.2 WEBML.....	5
1.3.3 APLICACIONES WEB	8
1.3.4 COMPORTAMIENTO DEL USUARIO.....	8
1.3.5 JAVASCRIPT	9
1.3.6 AJAX.....	10
1.3.7 JSON	11
1.3.8 HTML5.....	12
1.3.9 CSS3.....	13
1.3.10 JQUERY	14
1.3.11 PHP.....	15
1.3.12 MySQL.....	16
1.3.13 BOOTSTRAP	17
1.3.14 MAILCHIMP	17
1.3.15 PAYPAL.....	18

1.3.16	ADMIN LTE	18
2.	METODOLOGÍA	19
2.1	INTRODUCCIÓN.....	19
2.2	ELECCIÓN DE LA METODOLOGÍA	19
2.3	CRITERIOS DE ELECCIÓN	19
2.3.1	PONDERACIÓN.....	19
2.3.2	DOCUMENTACIÓN	20
2.3.3	SIMPLICIDAD.....	20
2.3.4	COMPLETITUD.....	20
2.3.5	FLEXIBILIDAD.....	21
2.3.6	RESULTADOS	21
2.4	ANÁLISIS DE REQUERIMIENTOS	22
2.4.1	MÓDULO DE FACTURACIÓN	23
2.4.2	MÓDULO DE VENTA EN LÍNEA.....	25
2.4.3	MÓDULO DE MERCADERÍA	26
2.5	DISEÑO DE DATOS	27
2.5.1	MÓDULO DE FACTURACIÓN	27
2.5.2	MÓDULO DE VENTA EN LÍNEA.....	29
2.5.3	MÓDULO DE MERCADERÍA	32
2.6	DISEÑO HIPERTEXTO.....	33
2.6.1	MÓDULO DE FACTURACIÓN	34
2.6.2	MÓDULO DE VENTA EN LÍNEA.....	39
2.6.3	MÓDULO DE MERCADERÍA	46
2.7	IMPLEMENTACIÓN.....	48
2.7.1	BASE DE DATOS	48
2.7.2	LOGIN.....	49
2.7.3	MAILCHIMP.....	51
2.7.4	GOOGLE MAPS API.....	52

2.7.5	PAYPAL.....	52
2.7.6	EMISIÓN DE FACTURAS.....	55
2.7.7	DESPLEGAR EL SISTEMA WEB EN UN SERVIDOR LOCAL.....	56
2.8	INSTALACIÓN.....	59
2.8.1	DESPLEGAR EL SISTEMA WEB AL SERVICIO DE AWS.....	59
2.8.2	INSTALAR EL SISTEMA WEB EN LA MÁQUINA VIRTUAL	60
2.8.3	SISTEMA WEB IMPLEMENTADO EN AWS	61
3.	RESULTADOS Y DISCUSIÓN	62
3.1	PRUEBAS Y EVALUACIÓN	62
3.1.1	PRUEBAS DE CONTENIDO	62
3.1.2	PRUEBAS DE INTERFAZ DE USUARIO	64
3.1.3	PRUEBAS DE NAVEGACIÓN.....	65
3.1.4	PRUEBAS DE COMPATIBILIDAD	66
3.2	ENCUESTAS REALIZADAS.....	68
3.3	TABULACIÓN DE LOS RESULTADOS	68
3.4	ANÁLISIS DE LOS RESULTADOS	72
3.5	CORRECCIÓN DE ERRORES.....	72
4.	CONCLUSIONES Y RECOMENDACIONES.....	74
4.1	CONCLUSIONES.....	74
4.2	RECOMENDACIONES	75
5.	BIBLIOGRAFÍA	77
6.	ANEXOS.....	80

ÍNDICE DE FIGURAS

Figura 2.1. Diagrama de actividad del proceso de facturación.	24
Figura 2.2. Diagrama de actividad del proceso de venta en línea.	26
Figura 2.3. Diagrama de actividad del proceso de mercadería.....	27
Figura 2.4. Diagrama de casos de uso del módulo de facturación.	27
Figura 2.5. Diagrama relacional del módulo de facturación.....	28
Figura 2.6. Diagrama de secuencia.....	28
Figura 2.7. Diagrama de casos de uso del módulo de venta en línea.	29
Figura 2.8. Diagrama relacional del módulo de venta en línea.....	30
Figura 2.9. Diagrama de clases del módulo de venta en línea.	31
Figura 2.10. Diagrama de secuencia.....	31
Figura 2.11. Diagrama de casos de uso del módulo de mercadería.....	32
Figura 2.12. Diagrama relacional del módulo de mercadería.	32
Figura 2.13. Diagrama de secuencia.....	33
Figura 2.14. Diagrama hipertexto del módulo de facturación.....	34
Figura 2.15. Página login.php.	34
Figura 2.16. Página facturas-panel.php.....	35
Figura 2.17. Página editar-iva.php.	35
Figura 2.18. Página crear-factura.php.....	36
Figura 2.19. Página resumen-factura.php.	36
Figura 2.20. Página editar-cliente.php.....	37
Figura 2.21. Página editar-productos.php.	37
Figura 2.22. Página editar-fecha.php.	38
Figura 2.23. Página listar-facturas.php.....	38
Figura 2.24. Página detalle-factura.php.....	39
Figura 2.25. Diagrama hipertexto del módulo de venta en línea para el administrador....	40
Figura 2.26. Página lista-clientes.php.	40
Figura 2.27. Página lista-toners-HP.php.....	41
Figura 2.28. Página editar-toners-hp.php.	41
Figura 2.29. Página lista-toners-lex.php.....	42
Figura 2.30. Página editar-toners-lex.php.	42
Figura 2.31. Diagrama hipertexto del módulo de venta en línea para el cliente.....	43
Figura 2.32. Página index.php.	43
Figura 2.33. Página nosotros.php.	44
Figura 2.34. Página productos.php.	44

Figura 2.35. Página carrito_compra.php.	45
Figura 2.36. Página pago_finalizado.php.	45
Figura 2.37. Diagrama hipertexto del módulo de mercadería.	46
Figura 2.38. Página ingresar-merca.php.	46
Figura 2.39. Página salida-merca.php.....	47
Figura 2.40. Página listar-salida.php.	47
Figura 2.41. Página listar-merca.php.	48
Figura 2.42. Tabla administradores.....	49
Figura 2.43. Página editar-administrador.php.....	51
Figura 2.44. Creación del formulario de suscripción.....	51
Figura 2.45. Generación del formulario de Mailchimp.....	52
Figura 2.46. Venta de inicio del servidor WAMP.	56
Figura 2.47. Carpeta Tesis.....	57
Figura 2.48. Panel de contro de PhpMyAdmin.	57
Figura 2.49. Página menú-principal.php.....	58
Figura 2.50. Página index.php.	58
Figura 2.51. Página menú-principal.php.....	58
Figura 2.52. Descripción de la máquina virtual EC2.	59
Figura 2.53. Instalación correcta del servidor Apache.	59
Figura 2.54. Instalación correcta del servidor MySQL.	60
Figura 2.55. Implementación del sistema web en línea.	60
Figura 2.56. Implementación de la base de datos en línea.....	61
Figura 2.57. Página de inicio del sistema web.....	61
Figura 3.1. Error de compatibilidad en el campo Nombre_cliente.....	62
Figura 3.2. Resumen de facturas ingresadas.....	63
Figura 3.3. Resumen de factura ingresada..	63
Figura 3.4. Página index.php desplegada en el navegador Google Chrome.	64
Figura 3.5. Interacción con el menú de la página productos.php.	65
Figura 3.6. Página resumen-factura.php.	66
Figura 3.7. Sistema web en el navegador Google Chrome.	66
Figura 3.8. Sistema web en el navegador Mozilla Firefox.	67
Figura 3.9. Sistema web en el navegador Edge.	68
Figura 3.10.Tabulación de la pregunta 1.	69
Figura 3.11. Tabulación de la pregunta 2.	69
Figura 3.12. Tabulación de la pregunta 3.	69

Figura 3.13. Tabulación de la pregunta 4.....	69
Figura 3.14. Tabulación de la pregunta 5.....	70
Figura 3.15. Tabulación de la pregunta 6.....	70
Figura 3.16. Tabulación de la pregunta 7.....	70
Figura 3.17. Tabulación de la pregunta 8.....	70
Figura 3.18. Tabulación de la pregunta 9.....	71
Figura 3.19. Tabulación de la pregunta 10.....	71
Figura 3.20. Tabulación de la pregunta 11.....	71
Figura 3.21. Tabulación de la pregunta 12.....	71
Figura 3.22. Tabulación de la pregunta 13.....	72
Figura 3.23. Tabulación de la pregunta 14.....	72
Figura 3.24. Tabulación de la pregunta 15.....	72

LISTA DE TABLAS

Tabla 2. 1 Resultados de la mayor cantidad de documentación.	20
Tabla 2. 2. Resultados de la comparativa de las dos metodologías con el criterio de simplicidad.	20
Tabla 2. 3. Resultados de la comparativa de las dos metodologías con el criterio de completitud.	21
Tabla 2. 4. Resultados de la comparativa de las dos metodologías con el criterio de flexibilidad.	21
Tabla 2. 5. Resultado final de la comparativa.....	22
Tabla 2. 6. Detalle de los módulos del sistema web.	23

LISTA DE SEGMENTOS DE CÓDIGOS

Segmento de Código 1. 1. Ejemplo de código escrito en JavaScript.	10
Segmento de Código 1. 2. Ejemplo de AJAX.	11
Segmento de Código 1. 3. Ejemplo de JSON.	12
Segmento de Código 1. 4. Ejemplo de HTML5.	13
Segmento de Código 1. 5. Ejemplo de CSS3.....	14
Segmento de Código 1. 6. Ejemplo de JQuery.	15
Segmento de Código 1. 7. Ejemplo de PHP.....	16
Segmento de Código 1. 8. Ejemplo de MySQL.	17

RESUMEN

En la actualidad, en Ecuador existen empresas dedicadas a la distribución de repuestos de impresoras, arreglo de impresoras y equipos de oficina. En los últimos años se ha observado que el comercio y marketing en línea han tenido ventajas significativas para muchas empresas en todo el mundo [1]. Las empresas dedicadas a la distribución de tóners y suministros de oficinas se están popularizando y creciendo en Ecuador. Es por esto que el presente Trabajo de Titulación se enfoca en la creación de un sistema prototipo que permita gestionar, facturar y vender en línea.

Este sistema fue hecho con tecnologías de Front-End como: *AJAX*, *JSON*, *HTML5*, *JavaScript*, *JQuery* y *CSS3*. También con tecnologías de Back-End como: *PHP* y *MySQL* que juntos, permiten el correcto funcionamiento del sistema prototipo. Una vez finalizado el sistema prototipo localmente se alojó en Amazon Web Services.

Para la realización del sistema prototipo, se realizaron algunos procesos tales como: estudio teórico de las tecnologías, desarrollo del sistema prototipo, pruebas de funcionalidad local y en línea, y por último encuestas de funcionalidad.

Una vez finalizado el sistema prototipo, se pudo comprobar que dicho sistema cumple con los requerimientos de la empresa *Mr. Toner*.

Palabras Clave: *Front-End*, *Back-End*, *AJAX*, *JSON*, *HTML5*, *JavaScript*, *JQuery*, *CSS3*, *PHP*, *MySQL*.

ABSTRACT

Currently in Ecuador there are companies dedicated to the delivery of spare parts for printers, printers and office equipment. In recent years it has been observed that online commerce and marketing have had the advantages for companies around the world [1]. The companies are dedicated to the distribution of cartridges, printers and office equipment that are popularizing and growing in Ecuador. Therefore, the present title work has a focus on the creation of a prototype system that allows managing, billing and selling online.

This system was made with Front-End technologies such as: AJAX, JSON, HTML5, JavaScript, JQuery and CSS3 and Back-End technologies such as: PHP and MySQL that together, allow the correct functioning of the prototype system. Once the prototype system was finished locally, it was implemented in a web hosting environment on this Amazon Web Services.

For the realization of the prototype system some processes were carried out, these processes were: theoretical study of the technologies that were used in the development, prototype system development, tests of local and online functionality, and finally functionality surveys.

Once the prototype system was finished, it was possible to verify that said system meets the requirements of the company Mr. Toner.

Keywords: *Front-End, Back-End, AJAX, JSON, HTML5, JavaScript, JQuery, CSS3, PHP, MySQL.*

1. INTRODUCCIÓN

El presente Trabajo de Titulación describe la creación de un sistema web prototipo, que permite facturar, gestionar mercadería y vender en línea basado en la información proporcionada por la empresa Mr. Toner, dedicada a la distribución de tóners y suministros de oficina.

En el primer capítulo se estudian las tecnologías utilizadas en el desarrollo del sistema web, entre las cuales se encuentran: *AJAX*, *JSON*, *HTML5*, *JavaScript*, *JQuery*, *CSS3*, *PHP* y *MySQL*, además se estudian metodologías a utilizarse como OOHDM y WebML y finalmente herramientas como: MailChimp, Bootstrap y Admin Lte.

En el segundo capítulo se describe la etapa de esquematización y elaboración del sistema web. La etapa de esquematización comprende desde, la elección de la metodología a utilizarse, análisis de los requerimientos, diseño de datos (diagramas relacionales, diagramas de casos de uso, diagramas de flujo, etc.), también se hace el diseño hipertexto y finalmente la instalación del sistema web en Amazon Web Services.

En la etapa de elaboración se describe el desarrollo de la base de datos y el desarrollo del código del sistema web con tecnologías de Front-End y Back-End, además se pone en funcionamiento el sistema localmente.

En el tercer capítulo se analizan los resultados obtenidos de encuestas de funcionamiento del sistema, también se analizan las pruebas de contenido, pruebas de interfaz de usuario, pruebas de navegación y las pruebas de compatibilidad.

Finalmente, en el cuarto capítulo se realizan conclusiones y recomendaciones.

1.1 OBJETIVOS

El objetivo del presente Trabajo de Titulación es desarrollar un sistema web que permita realizar facturación, gestión de mercadería y venta en línea para empresas de distribución de tóner y suministros de oficina.

Los objetivos específicos son:

- Analizar los fundamentos teóricos.
- Diseñar los componentes del sistema web.
- Implementar los componentes del sistema web.
- Evaluar los resultados obtenidos.

1.2 ALCANCE

El prototipo estará conformado por los siguientes componentes: una base de datos, una aplicación web y finalmente los servicios web como: MailChimp, PayPal y Google Maps API. En la Figura 1.1 se expone como el cliente accede a los diferentes servidores y servicios web mediante el protocolo HTTP.



Figura 1.1. Esquema del sistema.

La estructura del sistema web contará con una arquitectura cliente-servidor [2], además contará con dos tipos de tecnologías: Front-End (la cual se encarga de la parte visual y el comportamiento del sistema web) ejecutándose en el cliente y Back-End (la cual se encarga de administrar, guardar y entregar los datos al cliente) ejecutándose en el servidor. Se configurará MailChimp (que es un distribuidor de servicios de mercadeo por correo electrónico [3]) para enviar un boletín semanal. Mediante Google Maps API los usuarios podrán encontrar el sitio en donde se encuentra la empresa y obtener una breve descripción del lugar, se adaptará Paypal (una plataforma de pagos en línea que soporta transferencias de dinero entre usuarios y es otra opción electrónica a los métodos de pago clásicos como cheques y giros postales [4]) al sistema para la venta de los productos. Para la entrega de los datos se utilizará el protocolo HTTP (Hypertext Transfer Protocol) el cual es un protocolo de comunicación que intercambia información en la World Wide Web

(WWW) [5]. El sistema web estará provisto de una base de datos que tendrá información sobre los productos a la venta, administradores y clientes. En la Figura 1.2 se representan las 3 capas que tiene el sistema web. La capa visual (Interfaz de usuario), la capa lógica (Procesador de la entrada del Front-End e Interacción con el usuario) y la capa de base de datos.

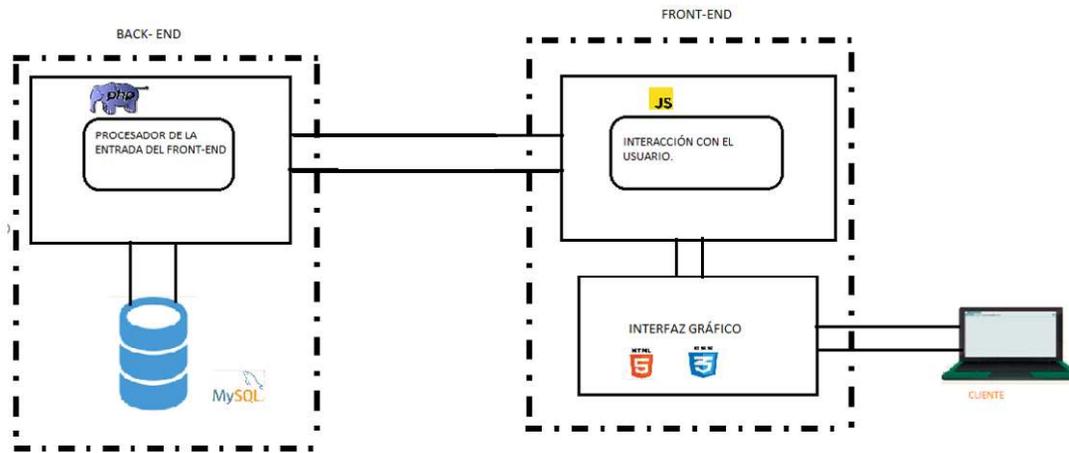


Figura 1.2. Esquema visual del sistema web.

El sistema web tendrá un nivel de protección en contra de ataques de SQL Injection para proteger los pagos y datos del usuario, se guardará en la base de datos aquellas personas que hicieron el pago mediante una cuenta en PayPal. Los perfiles de usuario estarán constituidos por el administrador y el cliente, el administrador contará con un área de administración personalizada (o Dashboard) creada con la herramienta Admin Lte para gestionar los datos guardados en la base de datos del sistema web.

Se programará el Front-End del sistema web para recoger los datos de entrada del usuario y se ajustarán a las especificaciones que demanda el Back-End para poder procesarlos, permitiendo que el Back-End devuelva una respuesta al Front-End y finalmente este reciba y exponga de una forma entendible al usuario.

En el Back-End se codificará el proceso de venta en línea, también se programará para recibir los pagos de Paypal y verificar aquellos usuarios que lo han realizado, también se codificará el proceso de facturación y también se codificará el manejo de mercadería. En el Front-End se codificará la recolección de datos (ingreso de nombres, número de productos, valor de venta del producto, etc.). Por último, se codificará el Dashboard para administrar el sistema web.

El cliente tendrá una sección de venta productos, otra de testimonios de personas que han comprado o recibido algún servicio por parte de la empresa, también una sección para saber cuáles son los integrantes de la empresa para comunicarse con ellos, otra parte para saber en dónde está ubicada la empresa.

Además, contará con un Dashboard elaborado mediante la herramienta Admin Lte para la administración del sistema web.

1.3 MARCO TEÓRICO

1.3.1 OOHDM

El modelo *Object Oriented Hypermedia Design Methodology* o OOHDM es una extensión de *Hypermedia Design Methodology* con orientación de objetos, que comprende cinco etapas para el desarrollo [6]:

- **Etapa de obtención de los requerimientos:** se adquiere todos los requisitos necesarios para la elaboración del aplicativo.
- **Etapa de modelo conceptual:** se desarrolla un esquema que se muestra por objetos de dominio o de clases y las relaciones entre estos objetos.
- **Etapa de diseño navegacional:** se definen las clases navegacionales como: nodo, enlaces y estructuras de acceso; en donde los nodos son las ventanas lógicas sobre las clases y los enlaces se inducen de las relaciones de la etapa anterior.
- **Etapa de diseño de interfaces abstractas:** se especifica la interfaz abstracta, aquí se presentan los contextos navegacionales.
- **Etapa de implementación:** es donde se pone en práctica y funcionamiento los objetos de interfaz con los objetos de implementación [7].

En la Figura 1.3 se muestran cada una de las etapas que conforman OOHDM.



Figura1.3. Etapas y secuencia de OOHDM.

1.3.2 WEBML

El modelo WebML o Web Modeling Language es un modelo para aplicaciones web complejas que usan datos intensivamente, establece especificaciones gráficas dentro de un proceso de diseño completo que puede ser desarrollado por herramientas de diseño visual.

WebML comprende cuatro modelos ortogonales:

- **El modelo estructural:** establece el contenido de datos del sitio, la recolección de datos y el diseño de datos.
- **El modelo hipertexto:** describe uno o más hipertextos que se pueden usar en el sitio, cada hipertexto estipula una vista del sitio.
- **El modelo de presentación:** establece el diseño y la apariencia de las páginas, sin importar el dispositivo de salida y el lenguaje de presentación.
- **El modelo personalización:** los usuarios y grupos de usuarios son explícitamente modelados en un esquema en forma de entidades [8].

En la Figura 1.4 se muestra el esquema de WebML en donde se explica de manera visual los modelos ortogonales.

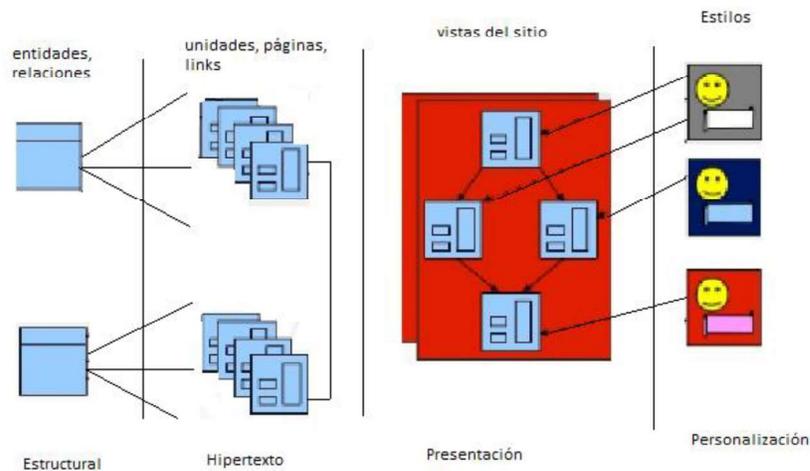


Figura1.4. Esquema de WebML.

WebML contiene siete fases las cuales son:

- **La fase de análisis de los requerimientos:** en donde se obtienen todos los requisitos y requerimientos que necesita el sistema web.
- **La fase de diseño de datos:** en donde los datos son expresados mediante diagramas para un mejor entendimiento de los requerimientos.
- **La fase de diseño hipertexto:** en donde se explica la navegación de las diferentes páginas del sistema web.
- **La fase de implementación:** en donde se codifica el sistema web.
- **La fase de pruebas y evaluación:** en donde se pone en marcha el sistema web y se busca posibles errores.
- **La fase de instalación:** en donde se instala el sistema web.
- **La fase de mantenimiento y evolución:** en donde se puede evaluar posibles cambios de ser necesario en el sistema web.

En la Figura 1.5 se muestran todas las fases y el orden en el que se debe seguir cada una de ellas, comenzando por los requerimientos del negocio y terminando por el mantenimiento y evolución [9].

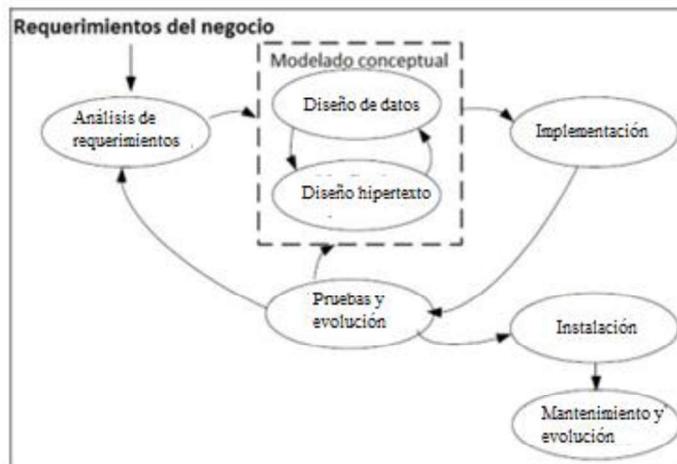


Figura 1.5 Fases de WebML.

WebML tiene los siguientes tipos de unidades para elaborar un diagrama de hipertexto como se muestra en la Figura 1.6 en donde se representan gráficamente cada uno de ellos.

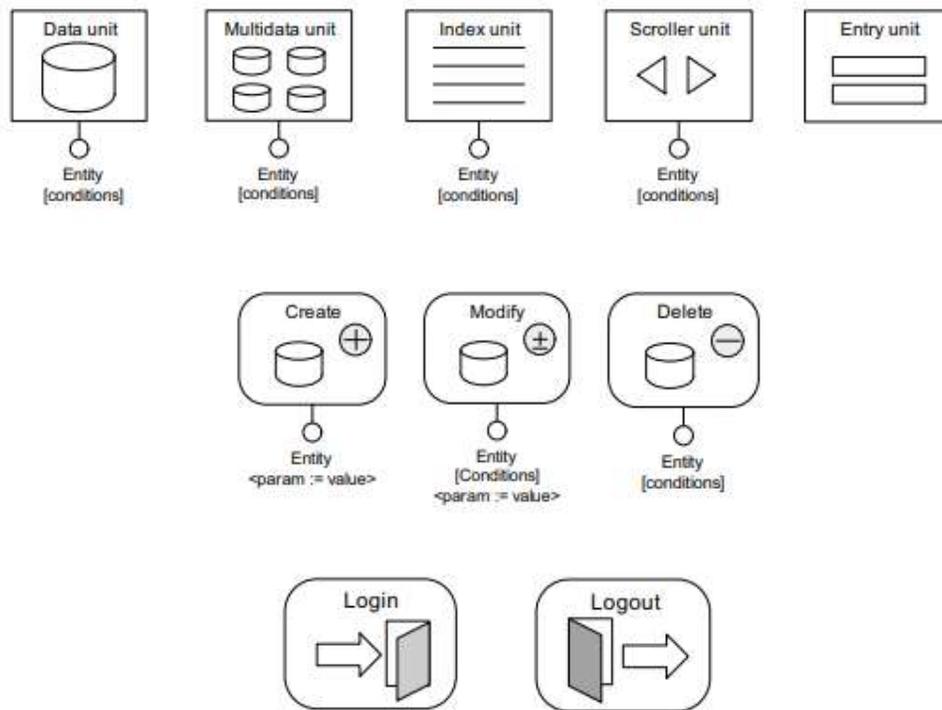


Figura 1.6. Unidades WebML.

- **Data Unit:** expone información de un único objeto.
- **Multidata Unit:** expone información sobre varios objetos.
- **Index Unit:** expone una lista de objetos.
- **Entry Unit:** se utiliza para exhibir un formulario.
- **Scroller Unit:** habilita el navegador de un conjunto ordenado de objetos pudiendo ser estos hiperenlaces.
- **Create:** permite crear una nueva entidad.
- **Delete:** permite borrar una entidad.
- **Modify:** permite modificar una entidad.
- **Login:** establece e identifica la identidad del usuario que está ingresando al sitio.
- **Logout:** Permite abandonar la sesión de un usuario previamente ingresado.

1.3.3 APLICACIONES WEB

Una aplicación web es el conjunto de herramientas que los usuarios pueden usar a través de un servidor web mediante Internet o de una intranet, es un programa que los navegadores web interpretan y lo ejecutan.

La estructura comúnmente de una aplicación web está dada por tres capas la cuales son [10]:

- El navegador web (interpretando el código).
- El servidor web (ofreciendo y ejecutando el código).
- La base de datos (guarda los datos necesarios para la aplicación web).

En la Figura 1.7 se muestra gráficamente cada capa de una aplicación web.

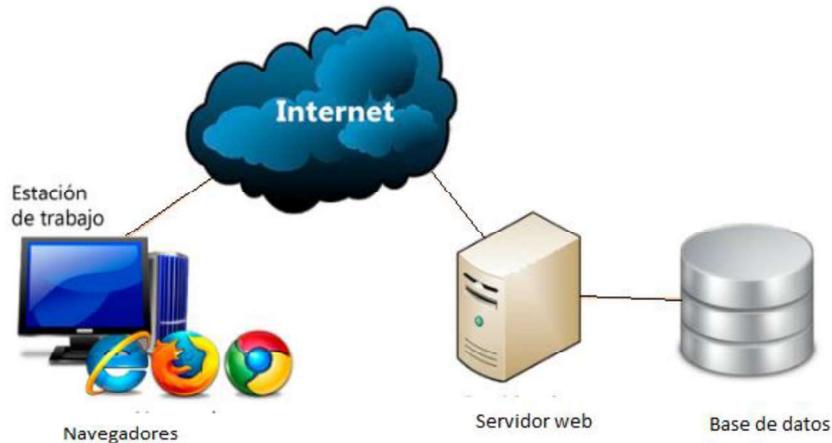


Figura 1.7. Esquema de una aplicación web.

1.3.4 COMPORTAMIENTO DEL USUARIO

El comportamiento del usuario hace referencia a la mejor estructura y contenido de un sitio web (tamaño de letra, colores, disposición de menús y botones, etc.) de tal manera que se asegure la detención y recepción de usuarios permitiéndoles obtener la mejor experiencia. Existen varias aplicaciones que permiten observar el comportamiento del usuario al usar navegadores, como por ejemplo: ClickHeat que es una aplicación web de código abierto que permite desarrollar mapas visuales de los clics que han realizado al navegar por un sitio web [11].

En la Figura 1.8 se muestra un ejemplo de un mapa obtenido con la herramienta ClickHeat, donde el color rojo muestra la mayor concentración de clics y el azul la menor concentración de clics en la página web.



Figura1.8. Mapa en las zonas que se han realizado clics en un sitio web.

1.3.5 JAVASCRIPT

JavaScript es un lenguaje de programación orientado a objetos, estipulado en el estándar ECMAScript. Se usa actualmente en el lado del cliente, en la arquitectura cliente-servidor, constituido como parte de un navegador web, facilitando la interfaz del usuario y proveyendo sitios web dinámicos.

La sintaxis de JavaScript es parecida al lenguaje de programación C, aunque adopta ciertos nombres y convenciones del lenguaje Java, sin embargo, JavaScript y Java son lenguajes para propósitos diferentes [12].

En el segmento de código 1.1 se muestra un ejemplo de JavaScript, en las líneas 3,4 y 5 se escucha el evento “input” mediante el método `addEventListener()`, luego de esto pasa a la función `validando_campos()`, en donde se valida el valor de las variables *nombre*, *apellido*, *email* y *direccion* estén vacíos, si están vacíos se inserta código HTML y estilos para alertar al usuario, de lo contrario se borra el código HTML y estilos insertados anteriormente.

```

1 //validar campos obligatorios
2 nombre.addEventListener('input',validando_campos);
3 apellido.addEventListener('input',validando_campos);
4 email.addEventListener('input',validando_campos);
5 direccion.addEventListener('input',validando_campos);
6 function validando_campos(){
7     if(this.value == ''){
8         error.style.display='block';
9         error.innerHTML='Este campo es obligatorio!';
10        this.style.border='1px solid red';
11        error.style.border='1px solid red';
12        btnregistro.disabled= true;
13    }
14    else{
15        error.style.display='none';
16        this.style.border='1px solid #cccccc';
17        btnregistro.disabled= true;
18    }
19 }

```

Segmento de Código 1. 1. Ejemplo de código escrito en JavaScript.

1.3.6 AJAX

Asynchronous JavaScript And XML (AJAX) es una tecnología de desarrollo web para realizar aplicaciones interactivas, ejecutándose en el cliente (navegador web) mientras mantiene una comunicación asincrónica con el servidor web en segundo plano. De esta manera se puede hacer modificaciones en sitios web sin necesidad de volver a recargarlos, optimizando la interactividad, velocidad y usabilidad de las aplicaciones web [13].

En el segmento de código 1.2 se muestra un ejemplo de AJAX en donde se define las opciones del método AJAX, *type* define la petición HTTP enviada por el formulario, *data* define los datos de entrada que en este caso es la variable *datos*, *url* es una cadena que contiene la URL (Uniform Resource Locator) a la que se envía la petición HTTP, *datatype* define el tipo de datos de entrada esperados del servidor web, *success* es una función de devolución de llamada que se ejecutará cuando la solicitud de AJAX tenga éxito, después se valida si la variable *resul.respuesta* que contiene la cadena de texto "éxito", de contener esta cadena se despliega una notificación de afirmación y se redirecciona a otra página web, de lo contrario se despliega una notificación de error.

```

1 $.ajax({
2   type: $(this).attr('method'),
3   data: datos,
4   url: $(this).attr('action'),
5   dataType: 'json',
6   success: function (data) {
7     resul= data;
8     if(resul.respuesta=='exito'){
9       swal(
10        'Correcto',
11        '¡Guardado correctamente!',
12        'success'
13       )
14       setTimeout(function(){
15         window.location.replace("http://localhost/tesis/administracion/resumen-factura.php")
16       },1000);
17     }
18
19     else{
20       swal(
21        'Incorrecto',
22        '¡Error en ingresar!',
23        'error'
24       )
25     }
26   }
27 }
28 })

```

Segmento de Código 1. 2. Ejemplo de AJAX.

1.3.7 JSON

JavaScript Object Notation (JSON) es un formato ágil de transferencia de datos. JSON es una fracción del formato de objetos de JavaScript aunque actualmente, debido a su acogida como otra opción a XML, se considera un formato de lenguaje independiente. El lenguaje es fácil de leer y de codificar para los programadores y además facilita el análisis del código para generar las respuestas desde los servidores. JSON usa citaciones de los lenguajes C, C ++, C #, Java, JavaScript, Perl, Python, etc. Estas propiedades hacen que JSON sea un lenguaje predilecto para el intercambio de datos [14].

JSON está constituido por dos estructuras:

- *Un conjunto de pares de nombre / valor.* Esto se desarrolla como un objeto, registro, estructura, diccionario, tabla hash, etc.
- *Una lista ordenada de valores.* Esto se realiza como: matriz, vector y lista.

En el segmento código 1.3 se muestra un ejemplo de JSON en donde se define la variable *empleados*, que es un arreglo que contiene 3 objetos cuyas llaves son Nombre y Apellido respectivamente.

```

1 {
2   var empleados;
3   empleados:[
4     {"Nombre":"John", "Apellido":"Doe"},
5     {"Nombre":"Anna", "Apellido":"Smith"},
6     {"Nombre":"Peter", "Apellido":"Jones"}
7   ]
8 }

```

Segmento de Código 1. 3. Ejemplo de JSON.

1.3.8 HTML5

HyperText Markup Language, versión 5 (HTML5) es la revisión más reciente del lenguaje básico de la World Wide Web, HTML.

Al no ser aceptado en versiones antiguas de navegadores web (por sus etiquetas nuevas), se recomienda al usuario tener la última versión del navegador, para poder explotar todo el potencial de HTML5.

La realización de este lenguaje de marcado es regulada por el World Wide Web Consortium (W3C) [15].

Algunas nuevas novedades que trae la versión cinco de HTML son:

- Tiene etiquetas (canvas 2D y 3D, vídeo, audio) para indicar los contenidos multimedia.
- Etiquetas para usar grandes cantidades de datos: Datagrid, Details, Menu y Command.
- Mejoras en los formularios. Nuevos tipos de datos (eMail, number, url, datetime)
- Drag & Drop. Nueva funcionalidad para enviar objetos como imágenes.

En el segmento de código 1.4 se muestra la etiqueta <section> la cual contiene etiquetas <h2> y de párrafo para mostrar diferentes cadenas de texto con diferentes tamaños y estilos.

```

1 <section class="sección contenedor">
2   <h2><p><i class="fa fa-credit-card aria-hidden="true"></i> ¿Cómo Ordenar?</p>
3   </h2>
4   <p>
5
6     Trabajamos de forma rápida y ágil, para su comodidad entregamos a domicilio sin costo alguno en la ciudad de
7     Quito, Para el resto del país se entrega Servientrega. </p>
8   </section>
9   <!-- sección -->
10  <section class="sección contenedor">
11   <h2><i class="fa fa-handshake-o" aria-hidden="true"></i> Garantías</p>
12   </h2>
13   <p>
14
15     Todos los productos comercializados por Mr. Toner cuentan con garantía de fábrica, para su tranquilidad
16     contamos con certificados de distribuidor autorizado. ¡Compre con garantía y confiabilidad!
17   </p>
18 </section>
19 <!-- sección -->

```

Segmento de Código 1. 4. Ejemplo de HTML5.

1.3.9 CSS3

Cascading Style Sheets, versión 3 (CSS3) es un lenguaje de diseño gráfico para establecer y desarrollar la presentación de una página estructurada escrita en un lenguaje de marcado. Es ampliamente utilizado para el diseño web de las páginas. Además, puede emplear estilos no visuales como hojas de estilo auditivas.

En el pasado, toda la información de la presentación de las páginas web era introducido en el código HTML, los tamaños, estilos y alineaciones de las fuentes, los bordes y tamaños. CSS3 otorga a los diseñadores la posibilidad de trasladar toda la información de la presentación a un diferente archivo (la hoja de estilos), dando como resultado un código HTML notoriamente más simple [16].

En el segmento de código 1.5 se muestra el selector *html* que contiene propiedades para cambiar el color de fondo, el tamaño de las letras y el interlineado de las cadenas de texto. También se muestran los selectores *::moz-selection* y *::selection* los cuales son pseudos selectores que sirven para aplicar reglas a una porción del documento, en este caso cambiar el color de fondo y la sombra de todo texto seleccionado por el usuario. El selector *hr* tiene propiedades para cambiar el largo de la etiqueta, el tamaño del borde, la forma y el color de la etiqueta, el tamaño del margen y para agregar espacio en el documento html.

```

1  html {
2    color: #222;
3    font-size: 1em;
4    line-height: 1.4;
5  }
6
7
8  ::-moz-selection {
9    background: #b3d4fc;
10   text-shadow: none;
11 }
12
13 ::selection {
14   background: #b3d4fc;
15   text-shadow: none;
16 }
17
18
19
20 hr {
21   display: block;
22   height: 1px;
23   border: 0;
24   border-top: 1px solid #ccc;
25   margin: 1em 0;
26   padding: 0;
27 }

```

Segmento de Código 1. 5. Ejemplo de CSS3.

1.3.10 JQUERY

jQuery es de código abierto, otorga varias funcionalidades fundamentadas en JavaScript, ahorra código en la sintaxis de JavaScript, por lo tanto con las funciones de esta biblioteca se tiene mejores soluciones en tiempo y espacio.

jQuery radica en un único fichero JavaScript que tiene las funcionalidades del Document Object Model o DOM. DOM es una interfaz que permite el acceso dinámico a través de la programación para acceder, añadir y cambiar dinámicamente contenido estructurado en documentos con lenguajes JavaScript [17].

La característica insignia de JQuery es que permite modificar el contenido del sitio web sin necesidad de recargarlo, mediante el manejo del árbol DOM y peticiones AJAX [18].

En el segmento de código 1.6 se muestra un ejemplo de JQuery, cualquier código que escriba dentro del método `$(document).ready()` se ejecutará una vez que la página DOM esté lista para ejecutar el código JavaScript. Además, se obtiene la clase sidebar-menu

para agregar a la función tree, también se obtiene el id registros para agregar a la función DataTable que contiene diferentes propiedades de la función.

```
1 $(document).ready(function () {
2     $('#sidebar-menu').tree()
3
4     $('#registros').DataTable({
5         'paging' : true,
6         'lengthChange': false,
7         'pageLength' : 8,
8         'searching' : true,
9         'ordering' : true,
10        'info' : true,
11        'autoWidth' : false,
12        'language' : {
13            paginate: {
14                next: 'Siguiete',
15                previous: 'Anterior',
16                last: 'Ultimo',
17                first: 'Primero'
18            },
19            info: 'Mostrando _START_ a _END_ de _TOTAL_ resultados',
20            emptytable: 'No hay registros',
21            infoEmpty: '0 Registros',
22            search: 'Buscar:'
23        }
24    });
```

Segmento de Código 1. 6. Ejemplo de JQuery.

1.3.11 PHP

Hypertext Preprocessor (PHP) es un lenguaje de programación de propósito general del lado del servidor.

Puede ser ejecutado en muchos de los servidores web y en la mayoría de los sistemas operativos [19]. PHP es utilizado en muchos sitios web [20]. Los sitios web hechos en PHP han decrecido paulatinamente en los últimos años, debido al surgimiento de tecnologías como: Node.JS, Java, ASP.NET, etc. Las características de PHP son:

- Orientado a la realización de aplicaciones web dinámicas.
- Es un lenguaje fácil de entender, ya que en su formación disminuyeron algunas especificaciones.
- El código fuente es oculto al navegador web y al cliente, porque el servidor es el que ejecuta el código y envía su resultado HTML al navegador.
- Capacidad de conexión con motores de base de datos que se usan en la actualidad, como MySQL y PostgreSQL.

En el segmento de código 1.7 se muestra cómo se realiza consultas a una base de datos mediante PHP se declaran variables que son recibidas en la petición POST de HTTP, después de eso se parametrizan los datos para realizar una consulta de actualizar los campos de la tabla clientes, al final se cierra la conexión y se detiene el código con la variable *\$respuesta*.

```
1 <?php
2 include_once "funciones/funciones.php";
3 $nombre = $_POST['nombre'];
4 $cedula = $_POST['cedula'];
5 $direccion = $_POST['direccion'];
6 $celular = $_POST['telefono'];
7 $email = $_POST['correo'];
8
9 $id_obtenido = $_POST['id_registro'];
10
11
12 if($_POST['registrar'] == 'actualizar'){
13
14
15     try {
16         $stmt = $conn->prepare("UPDATE cliente SET Nombre_cliente = ?, Cedula_ruc = ?, Direccion_cliente = ?, Telefono_cliente = ?,
17 Correo_electronico = ?, Fecha_edicion=NOW() WHERE ID_cliente = ? ");
18 $stmt->bind_param("sssssi", $nombre, $cedula, $direccion, $celular,$email, $id_obtenido);
19 $stmt->execute();
20 if($stmt->affected_rows) {
21     $respuesta = array(
22         'respuesta' => 'exito',
23         'id_actualizado' => $id_obtenido
24     );
25 } else {
26     $respuesta = array(
27         'respuesta' => 'error'
28     );
29 }
30 $stmt->close();
31 $conn->close();
32
33 } catch(Exception $e) {
34     $respuesta = array(
35         'respuesta' => $e->getMessage()
36     );
37 }
38
39 die(json_encode($respuesta));
40
```

Segmento de Código 1. 7. Ejemplo de PHP.

1.3.12 MySQL

MySQL es un sistema de gestión de bases de datos, es de código abierto y es considerado el sistema gestor más popular del mundo [21].

En aplicaciones web, hay poca concurrencia en la alteración de datos, pero el entorno es minucioso en lectura de datos, lo que hace que MySQL sea preciso para este tipo de aplicaciones web [22].

Al principio, MySQL no tenía elementos esenciales en las bases de datos relacionales como: integridad referencial y transacciones, sin embargo cautivó a los programadores de sitios web por su contenido dinámico y simplicidad.

Algunas características de MySQL son:

- Gran subconjunto del lenguaje SQL.
- Claves foráneas.
- Conectividad segura.
- Replicación.

En el código 1.8 se muestra la inserción de datos de la tabla *administrador* y creación de la tabla *ayuda_detalle*, que tiene un llave primaria y un índice para su respectiva relación, también se muestra la creación de la tabla *características* que contiene un una llave primaria y un campo de cadena de texto variable.

```
1 INSERT INTO `administradores` (`id_administrador`, `nickname`, `nombres`, `pass`, `Fecha_edicion`) VALUES
2 (2, 'root', 'Jefferson Benavides', '$2y$12$Z183qJZs0NXLEiAIC3hGl.brs7pkfD5uttWxtsbqKbTT4p0iuLLZO', '2019-01-23
3 00:00:00');
4
5 CREATE TABLE `ayuda_detalle` (
6   `ID_producto_detalle` int(11) NOT NULL,
7   `ID_factura_detalle` int(11) NOT NULL
8 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
9
10 CREATE TABLE `caracteristicas` (
11   `ID_caracteristica` int(11) NOT NULL,
12   `nombre_carac` varchar(50) NOT NULL
13 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Segmento de Código 1. 8. Ejemplo de MySQL.

1.3.13 BOOTSTRAP

Bootstrap es conjunto de herramientas de código abierto para el diseño de páginas y aplicaciones web. Tiene plantillas de diseño basadas en HTML y CSS, así como extensiones de JavaScript adicionales. Bootstrap solo se ocupa en el Front-End.

Bootstrap es modular y se basa en una serie de hojas de estilo que tienen una diversidad de elementos de la herramienta. Una hoja de estilo nombrada bootstrap.less contiene los elementos de las hojas de estilo. Los programadores pueden modificar el mismo archivo de Bootstrap para seleccionar los elementos que requieran en su proyecto [23].

1.3.14 MAILCHIMP

Mailchimp es una plataforma de automatización de marketing y provee un servicio de marketing por correo electrónico. Mailchimp comenzó como un servicio de pago, pero en

el año 2009 añadió una opción gratis de uso. En 2017, la empresa ganaba decenas de miles de nuevos usuarios por día [24].

1.3.15 PAYPAL

Es un sistema de pagos en línea que tolera transferencias de dinero entre usuarios, y tiene como propósito ser una opción electrónica a los métodos de pago convencionales como por ejemplo: giros postales y cheques. PayPal es una forma de pago por Internet muy acogida en el mundo [25]. El sistema trabaja como un procesador de pagos para tiendas en línea, sitios web de subastas entre otras [26].

PayPal propone una política de protección al comprador, de hasta dos mil quinientos dólares americanos, cubriendo problemas como: producto no recibido o producto muy distinto al descrito, cubriendo el valor del producto y los gastos de envío. Por cada una de las transacciones, PayPal cobra una comisión entre el dos y cuatro por ciento.

1.3.16 ADMIN LTE

Admin Lte es una plantilla para aplicaciones web de código abierto, que se emplea para paneles de administración o control. Es una plantilla HTML receptiva que utiliza CSS y Bootstrap, además se puede modificar elementos para crear un diseño que se pueda utilizar como interfaz de usuario para la administración aplicaciones web.

2. METODOLOGÍA

En este capítulo se explican las dos etapas (esquemmatización y elaboración) que comprende la realización del sistema web.

2.1 INTRODUCCIÓN

La etapa de esquematización contiene cuatro fases:

1. Elección de la metodología.
2. Análisis de los requerimientos.
3. Diseño de datos.
4. Diseño hipertexto.

La etapa de elaboración contiene dos fases:

1. Implementación.
2. Instalación.

2.2 ELECCIÓN DE LA METODOLOGÍA

Para la elección de la metodología se utilizó una matriz de decisión, en donde se analiza la tasa de la fuerza de relaciones entre las metodologías OOHDM y WebML.

2.3 CRITERIOS DE ELECCIÓN

Para poder elegir entre las dos metodologías, se consideró ciertas características de cada metodología, de tal modo que agilice el proceso de desarrollo. A continuación se muestran los criterios de elección:

- Documentación
- Simplicidad
- Completitud
- Flexibilidad

2.3.1 PONDERACIÓN

Se designa un valor de 20 puntos al cumplimiento total, el valor de 10 puntos al cumplimiento parcial, y 0 puntos al no cumplimiento.

2.3.2 DOCUMENTACIÓN

Para seleccionar la metodología que tiene mayor documentación, se realizó búsquedas de cada metodología usando buscadores como: Google, Bing y Dogpile teniendo como resultado lo indicado en la Tabla 2.1.

Tabla 2. 1 Resultados de la mayor cantidad de documentación.

Metodología	Buscadores						Total
	Google		Bing		Dogpile		
OOHDM	Cumple	20	Parcialmente	20	Parcialmente	10	50
WebML	Cumple	20	Cumple	20	Parcialmente	10	50

Se puede observar en base al análisis previo, que WebML y OOHDM tienen la misma ponderación.

2.3.3 SIMPLICIDAD

Comparando las dos metodologías con base en: fases, diagramas y navegación, WebML se acopla al requerimiento, ya que su implementación facilita el desarrollo y el mantenimiento del sistema web. Al llevar a cabo la comparación entre las dos metodologías descritas se ha obtenido la Tabla 2.2.

Tabla 2. 2. Resultados de la comparativa de las dos metodologías con el criterio de simplicidad.

Metodologías	Simplicidad	Puntaje
OOHDM	Parcialmente	10
WebML	Cumple	20

Se puede observar en base al análisis previo, que WebML tiene la mayor ponderación.

2.3.4 COMPLETITUD

Comparando las dos metodologías, se concluyó que OOHDM no tiene una fase de diseño tan detallada como WebML, ya que OOHDM integra en una sola fase la navegación y el diseño, también WebML tiene la fase de pruebas y evaluación que permite tener una perspectiva del comportamiento del sistema web antes de la instalación, además WebML

proporciona sus propios diagramas de hipertexto para la realización de un sistema web por lo que le hace más completo que OOHDM.

Al realizar la ponderación de cada una de las metodologías descritas se ha obtenido la Tabla 2.3.

Tabla 2. 3. Resultados de la comparativa de las dos metodologías con el criterio de completitud.

Metodologías	Complejidad	Puntaje
OOHDM	Parcialmente	10
WebML	Cumple	20

Se puede observar que con base en el análisis previo, WebML tiene mayor ponderación.

2.3.5 FLEXIBILIDAD

WebML se adapta a casi cualquier tipo de sistema web, a diferencia de OOHDM la cual no separa exclusivamente la fase de navegación, ya que tiende a confundir el funcionamiento y la navegación web de un sistema web complejo.

Al realizar la ponderación de cada una de las metodologías descritas se ha obtenido la Tabla 2.4.

Tabla 2. 4. Resultados de la comparativa de las dos metodologías con el criterio de flexibilidad.

Metodologías	flexibilidad	Puntaje
OOHDM	Parcialmente	10
WebML	Cumple	20

Se puede observar que con base en el análisis previo, WebML tiene mayor ponderación.

2.3.6 RESULTADOS

Los resultados obtenidos de la comparativa entre las dos metodologías WebML y OOHDM se resumen en la Tabla 2.5.

Tabla 2. 5. Resultado final de la comparativa.

	Metodologías	
Criterios de Selección	OOHDM	WebML
	Puntaje	Puntaje
Documentación	50	50
Simplicidad	10	20
Complejidad	10	20
Flexibilidad	10	20
Total	80	110

Con base en los resultados descritos en la Tabla 2.5 se puede concluir que WebML tiene la mayor ponderación, de acuerdo a los criterios descritos en la sección 2.3. Por lo tanto, WebML es la metodología seleccionada y será la base para el desarrollo del sistema web, teniendo en cuenta que la última fase no se realizará en este Trabajo de Titulación debido al alcance del mismo.

2.4 ANÁLISIS DE REQUERIMIENTOS

Para la obtención de los requerimientos se realizó una entrevista al Ingeniero Jefferson Benavides, dueño de la empresa Mr. Toner. Dicha entrevista está disponible en el ANEXO I, la cual tiene catorce preguntas enfocadas en tres aspectos de la empresa:

1. Proceso de venta.
2. Proceso de facturación.
3. Proceso de inventario.

De la entrevista se obtienen datos como: número de empleados que realizan las ventas, los roles de cada empleado, marca de los tóners que oferta la empresa, productos más vendidos, el proceso de cobro y emisión de facturas, el proceso de obtención, almacenamiento y salida de mercadería.

Se consideró que el sistema web debería realizar tres tipos de procesos:

- Venta en línea.
- Registro de la entrada y salida de la mercadería.
- Registro y emisión de facturas.

También se consideró que existen dos tipos de usuarios que van a utilizar el sistema web:

- Cliente.
- Administrador.

Se define a los clientes como aquellas personas que van a comprar en línea productos de la empresa Mr. Toner y los administradores como la(s) persona(s) que administra(n) la venta en línea, la emisión y registro de facturas, y realiza(n) proceso de inventario.

Además, se obtienen otros requerimientos funcionales como:

- Expiración de facturas emitidas con crédito.
- Anulación de facturas.
- Pago y venta en línea.
- Administración de las ventas en línea
- Salida y entrada de la mercadería.

Por lo tanto, de aquí en adelante se definen a los diferentes procesos de la empresa como *módulos* del sistema web descritos en la Tabla 2.6.

Tabla 2. 6. Detalle de los módulos del sistema web.

Módulo	Descripción
Facturación	Permite al usuario registrar y emitir facturas.
Venta en Línea	Permite al usuario comprar y administrar las ventas realizadas en línea.
Mercadería	Permite al usuario tener un registro actual de inventario.

2.4.1 MÓDULO DE FACTURACIÓN

Este módulo realiza el proceso manual de emisión y registro de facturas como se explica en la Figura 2.1. Para la emisión, se ha decidido separar los datos necesarios para una factura de la siguiente manera: los datos del cliente, de la factura y productos.

Los datos que son necesarios del cliente son:

- Razón social.
- Cédula de identidad (CI) o Registro Único de Contribuyentes (RUC.).
- Fecha.
- Teléfono.
- Dirección.

Los datos necesarios para la factura son:

- Fecha de emisión.
- Número de factura.
- Subtotal.
- Total del IVA.
- Valor del IVA.
- Total de la factura.
- Crédito.

Para el registro de facturas se deben tener en cuenta los siguientes datos:

- Fecha de emisión.
- Fecha de expiración de la factura (en caso de que exista crédito).
- Días restantes para la expiración (en caso de que exista crédito).
- Días después de la expiración (en caso de que exista crédito).

Los datos necesarios para los productos son:

- Descripción del producto.
- Valor unitario.
- Cantidad.
- Valor Total.

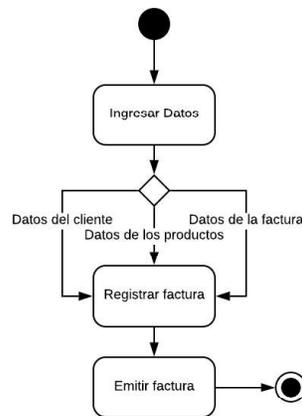


Figura 2.1. Diagrama de actividad del proceso de facturación.

2.4.2 MÓDULO DE VENTA EN LÍNEA

Este módulo realiza el proceso de venta en línea como se muestra en la Figura 2.2, permite además presentar los productos (tóners) disponibles y además gestionar su venta. La venta en línea se realiza mediante la plataforma PayPal, para permitir la facilidad de pago del cliente y crear fidelidad en la entrega de los productos comprados [27]. Para el desarrollo del módulo se deben obtener los datos del cliente y de los tóners, los cuales se describen a continuación.

Los datos del cliente son:

- Nombre.
- Apellido.
- CI o RUC.
- Dirección.
- Teléfono.
- Correo electrónico.

Para los datos de los tóners se debe dividir en dos subgrupos: los de marca HP y marca Lexmark, ya que estos son los dos únicos tipos de marca con los que trabaja la empresa.

Los datos de la marca HP son:

- Descripción.
- Código.
- Precio.

Los datos de la marca Lexmark son:

- Descripción.
- Pin de identificación.
- Precio.

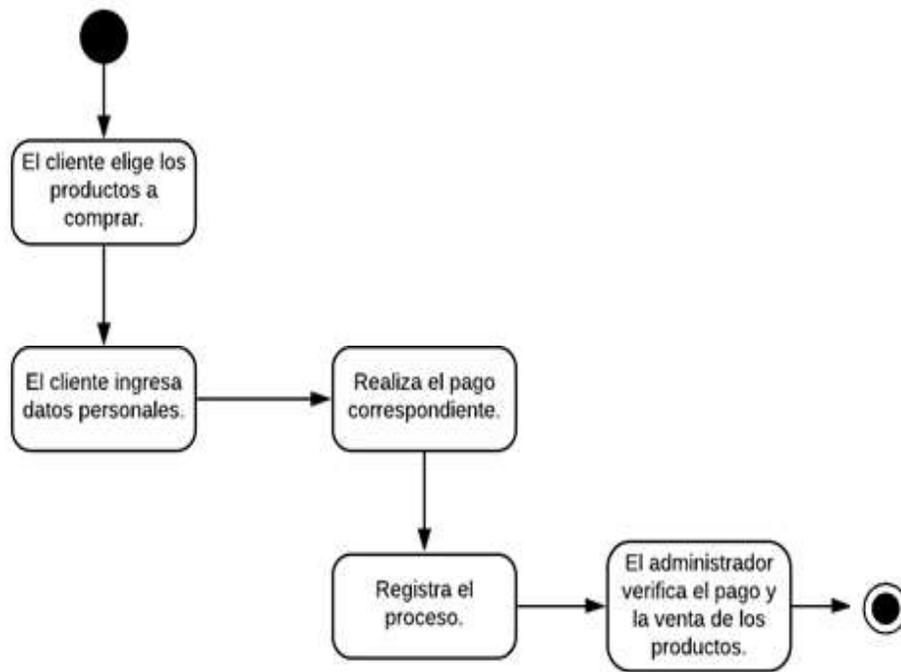


Figura 2.2. Diagrama de actividad del proceso de venta en línea.

2.4.3 MÓDULO DE MERCADERÍA

Este módulo realiza el proceso de inventario como se explica en la Figura 2.3, en donde se ingresa mercadería, se obtiene información de la cantidad de productos disponibles y su salida.

Los datos necesarios para el inventario son:

- Descripción de la mercadería ingresada.
- Cantidad de la mercadería ingresada.
- El precio unitario al cual se compró esa mercadería.
- Fecha de entrada.
- Característica (tóner genérico, tóner original, tóner usado, otro.)
- Cantidad actual en inventario.
- Fecha de salida de la mercadería.
- Cantidad de salida de mercadería.

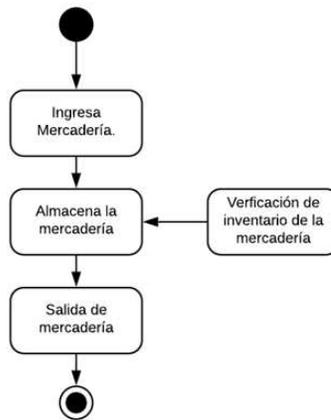


Figura 2.3. Diagrama de actividad del proceso de mercadería.

2.5 DISEÑO DE DATOS

En el diseño de datos se organiza la información obtenida en la sección 2.4 y se transforma en un modelo conceptual compuesto por: diagramas de caso de uso, diagramas relacionales, diagramas de clase, etc. Los cuales facilitan la comprensión y el desarrollo del sistema web.

2.5.1 MÓDULO DE FACTURACIÓN

2.5.1.1 Diagrama de casos de uso

En la Figura 2.4 se puede observar que el módulo facturación tiene un solo usuario, siendo este el administrador. Por el momento la empresa necesita un solo administrador, por lo que solo se da acceso a un solo usuario a este módulo.

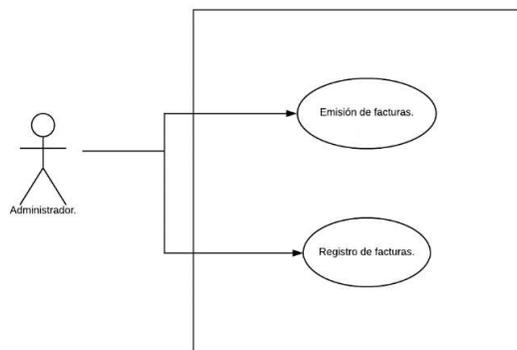


Figura 2.4. Diagrama de casos de uso del módulo de facturación.

2.5.1.2 Diseño de la base de datos

El módulo de facturación tiene seis tablas como se indica en el diagrama relacional de la Figura 2.5. Cada una relacionada con una clave primaria y una clave foránea.

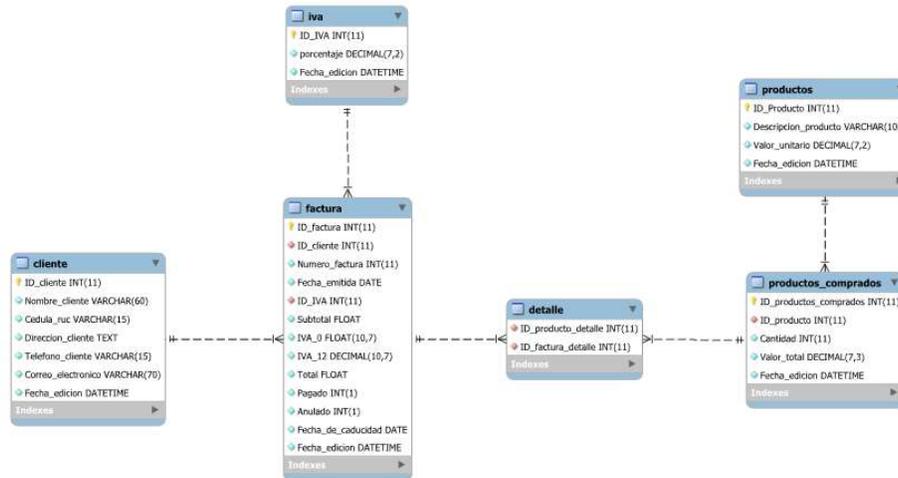


Figura 2.5. Diagrama relacional del módulo de facturación.

La tabla *cliente* almacena los datos del *cliente* de la factura emitida, la tabla *factura* almacena los datos de la factura emitida, la tabla *productos* almacena los datos de los productos, la tabla *productos_comprados* almacena los datos de los productos vendidos, la tabla *iva* almacena el porcentaje del IVA y finalmente la tabla *detalle* relacionar ayuda a relacionar la tabla *factura* con la tabla *productos_comprados* debido a que cada factura tiene varios productos y varios productos pueden estar en varias facturas.

2.5.1.3 Diagrama de secuencia

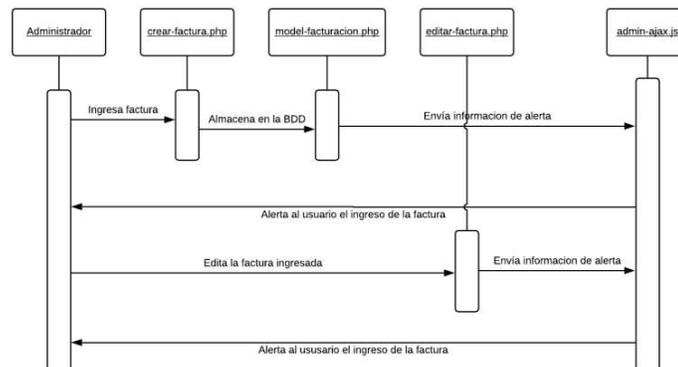


Figura 2.6 Diagrama de secuencia del módulo de facturación.

En la Figura 2.6 se observan los archivos utilizados en el módulo de facturación, el administrador ingresa los datos mediante el archivo *crear-factura.php*, después este archivo envía la información ingresada al archivo *model-facturacion.php* en donde se guardan los datos en la base de datos, después se envía una información de alerta al archivo *admin-ajax.js* y por último este responde al usuario con una notificación de ingreso de la factura. El usuario también puede editar la factura mediante el archivo *editar-factura.php*.

2.5.2 MÓDULO DE VENTA EN LÍNEA

2.5.2.1 Diagrama de casos de uso

En la Figura 2.7 se puede observar que el módulo de venta en línea tiene dos tipos de usuarios, siendo estos el administrador y el cliente. El administrador puede realizar la visualización de la venta (tóners comprados, el pago efectivo de la venta, la cantidad de cada uno de los tóners vendidos, el valor total pagado en la venta y si el pago se realizó correctamente), además el administrador puede editar el precio de los tóners y las diferentes características de estos. Por otro lado, el cliente solo puede visualizar los tóners ofertados y comprar los tóners ofertados.

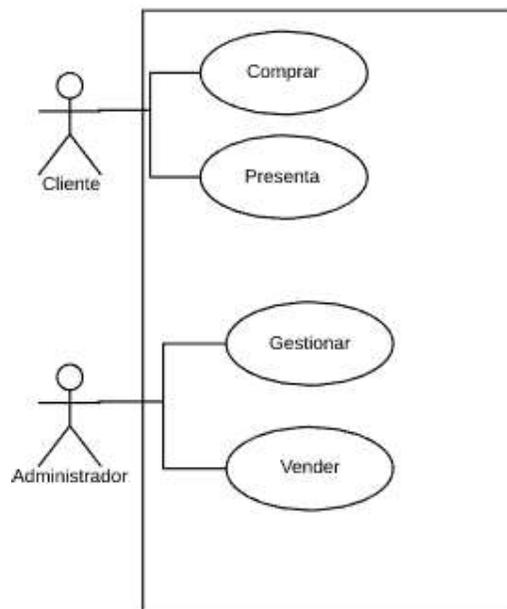


Figura 2.7. Diagrama de casos de uso del módulo de venta en línea.

2.5.2.2 Diseño de la base de datos

El módulo de facturación tiene tres tablas como se indica en la Figura 2.8. Estas tablas no se correlacionan entre si. Se ha decidido relacionar mediante arreglos en la parte lógica de la programación, para permitir la facilidad de la presentación final de los datos al administrador.

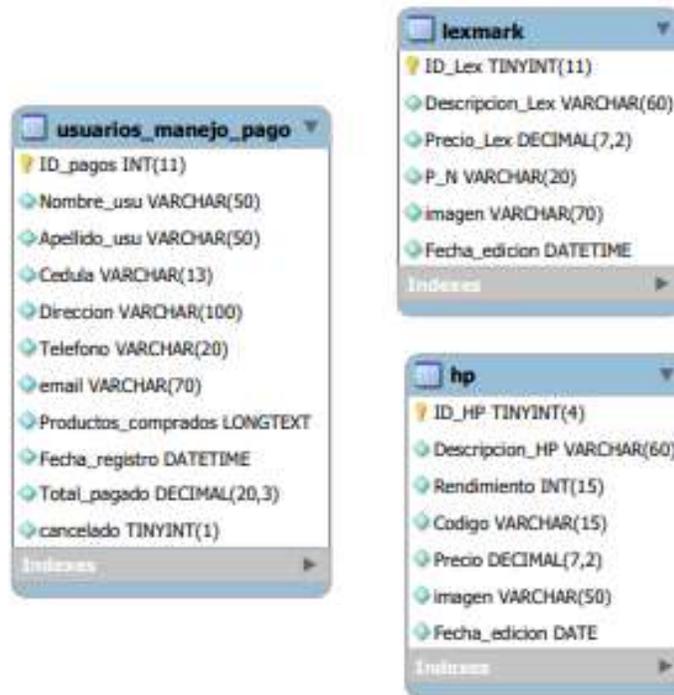


Figura 2.8. Diagrama relacional del módulo de venta en línea.

La tabla *HP* almacena los datos de los tóners de marca HP, la tabla *Lexmark* almacena los datos de los tóners de marca Lexmark, la tabla *usuarios_manejo_pago* almacena los datos de los clientes que realizaron la compra en línea, los productos obtenidos y el valor total de venta.

2.5.2.3 Diagrama de clases

Se utilizan ciertas clases del kit de desarrollo de software (SDK) de PayPal para la venta en línea, de tal manera que se adapte al sistema web, en la Figura 2.9 se muestra las clases y funciones respectivamente utilizadas en el sistema web.

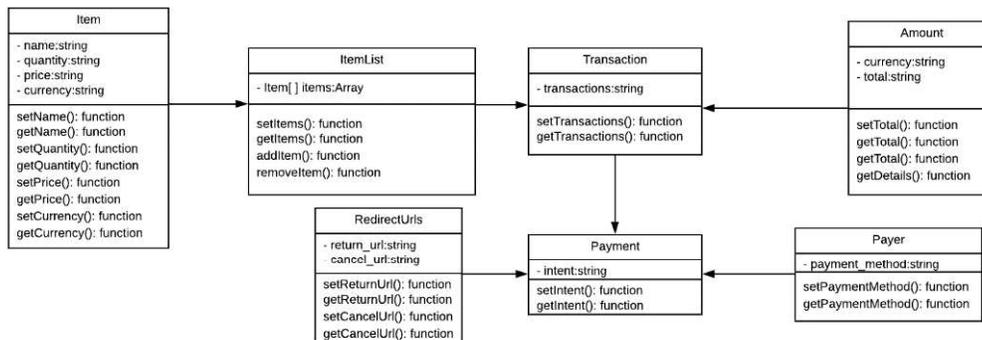


Figura 2.9. Diagrama de clases del módulo de venta en línea.

La clase *Item* permite agregar el nombre del tóner seleccionado, la moneda, la cantidad de tóners escogidos y su precio, la clase *Item_list* permite agregar los tóners elegidos, la clase *Amount* permite agregar la moneda (dólares, pesos, euros, etc.) a la transacción y la cantidad de dinero a pagar de la venta, que en este caso sería la suma del valor total de los ítems vendidos, la clase *Transactions* permite agregar las clases: *Amount*, *Item* e *Item_list*, para preparar el envío total a la plataforma de Paypal, la clase *Payer* permite agregar el método de pago, la clase *RedirectURL* permite redireccionar a la página de inicio después de realizar la venta y obtener el *PayerID* (identificador del pago realizado). la clase *Payment* permite enviar la información de la venta a la plataforma de Paypal.

2.5.2.4 Diagrama de secuencia

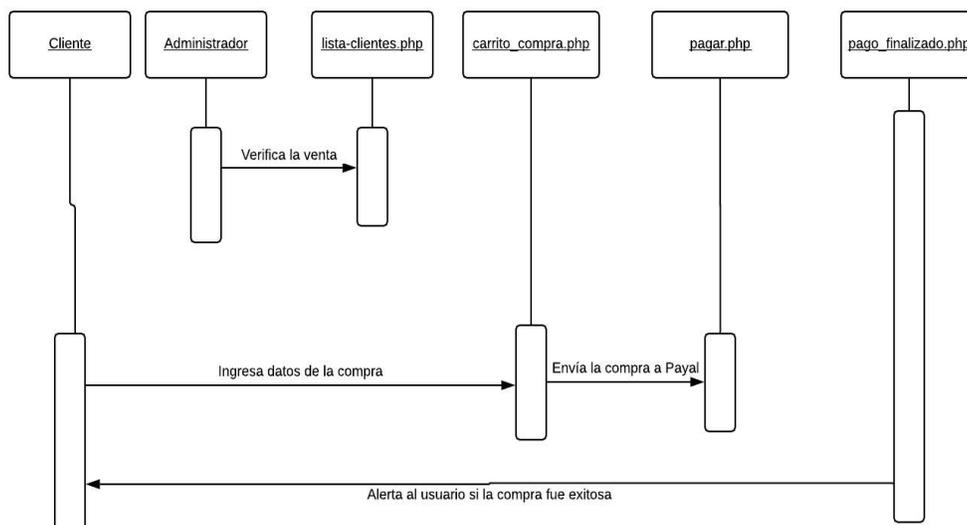


Figura 2.10. Diagrama de secuencia del módulo de venta en línea.

En la Figura 2.10 se observan los archivos utilizados en el módulo de venta en línea, el administrador verifica la venta realizada por el cliente mediante el archivo *lista-clientes.php*, por otro lado el cliente accede a la selección de tóners mediante el archivo *lista-caritto.php*, después se envía la información de la compra al archivo *pagar.php* en donde se realiza la compra y se envía los datos del pago a la plataforma de PayPal, y por último el archivo *pago_finalizado.php* notifica al cliente si la compra fue exitosa.

2.5.3 MÓDULO DE MERCADERÍA

2.5.3.1 Diagrama de casos de uso

En la Figura 2.11 se puede observar que el módulo facturación tiene un solo tipo de usuario, siendo este el administrador. Por el momento la empresa necesita un solo administrador, por lo que solo se da acceso a un solo usuario a este módulo.

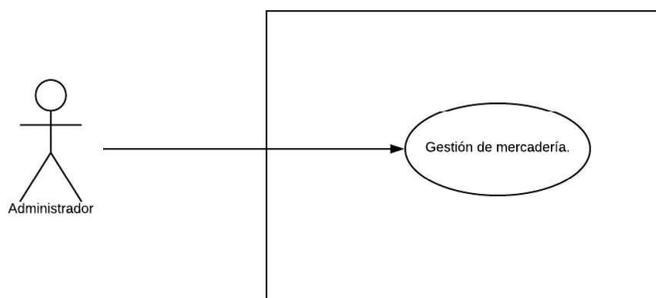


Figura 2.11. Diagrama de casos de uso del módulo de mercadería.

2.5.3.2 Diseño de la base de datos

El módulo de facturación tiene tres tablas como se indica en la Figura 2.12. Cada una relacionada con una clave primaria y una clave foránea.



Figura 2.12. Diagrama relacional del módulo de mercadería.

La tabla *caracteristicas* almacena los datos de las características de los tóners, la tabla *mercaderia* almacena los datos de la mercadería ingresada, la tabla *salida_mercaderia* almacena los datos de la mercadería obtenidos en la sección.

2.5.3.3 Diagrama de secuencia

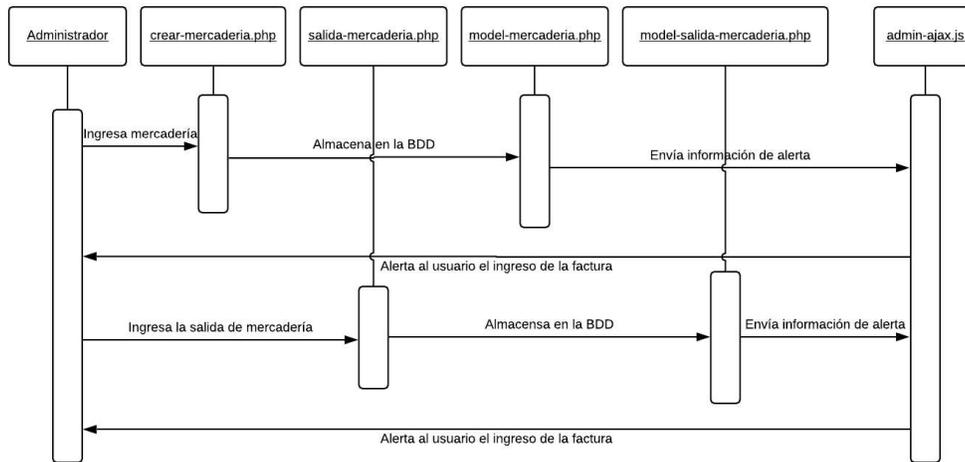


Figura 2.13. Diagrama de secuencia del módulo de mercadería.

En la Figura 2.13 se observan los archivos utilizados en el módulo de mercadería, el administrador ingresa la mercadería mediante el archivo *crear-mercaderia.php*, después este archivo envía la información ingresada al archivo *model-mercaderia.php* en donde se guardan los datos en la base de datos, después se envía una información de alerta al archivo *admin-ajax.js* y por último este responde al administrador con una notificación de ingreso de la mercadería. El administrador también puede registrar la salida de mercadería mediante el archivo *salida-mercaderia.php*, después este archivo envía la información ingresada al archivo *model-salida-mercaderia.php* en donde se guardan los datos en la base de datos, después se envía una información de alerta al archivo *admin-ajax.js* y por último este responde al administrador con una notificación de ingreso de la mercadería.

2.6 DISEÑO HIPERTEXTO

El diseño hipertexto transforma los requerimientos obtenidos en las secciones 2.4 y 2.5 en un modelo visual en donde se explica el funcionamiento de cada hipertexto o página dentro del sistema web.

2.6.1 MÓDULO DE FACTURACIÓN

En la Figura 2.14 se observa el diagrama hipertexto del módulo de facturación, Para el cual es necesario empezar por la Página de Login e iniciar sesión, después navegar por las diferentes páginas del sistema web, pudiendo crear, modificar y borrar facturas.

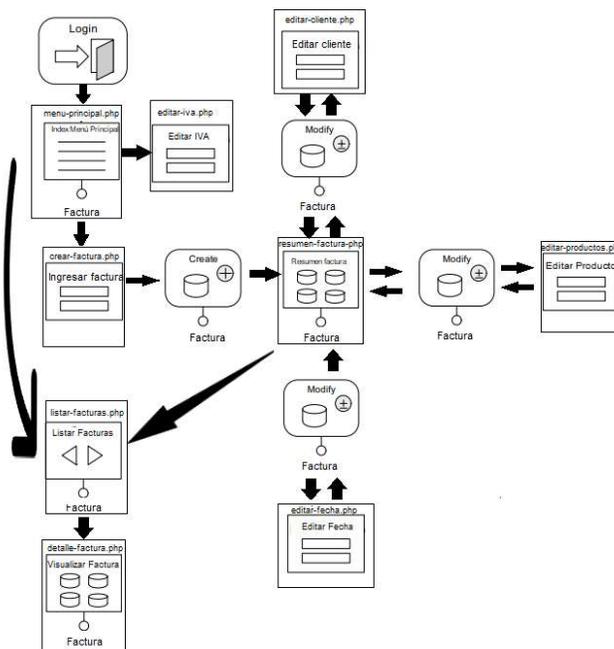


Figura 2.14. Diagrama hipertexto del módulo de facturación.

2.6.1.1 login.php

Esta página permite ingresar al módulo de facturación e iniciar sesión. En la Figura 2.15 se muestra la página login.php.

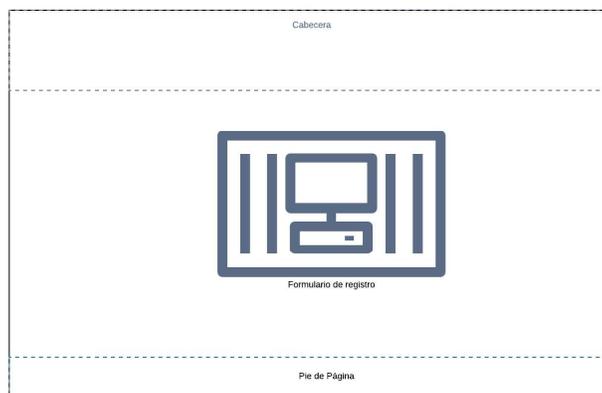


Figura 2.15. Página login.php.

2.6.1.2 facturas-panel.php

Esta página muestra consultas de la base de datos en forma de gráficas y widgets obtenidos de la tabla Factura. En la Figura 2.16 se muestra la página facturas-panel.php.

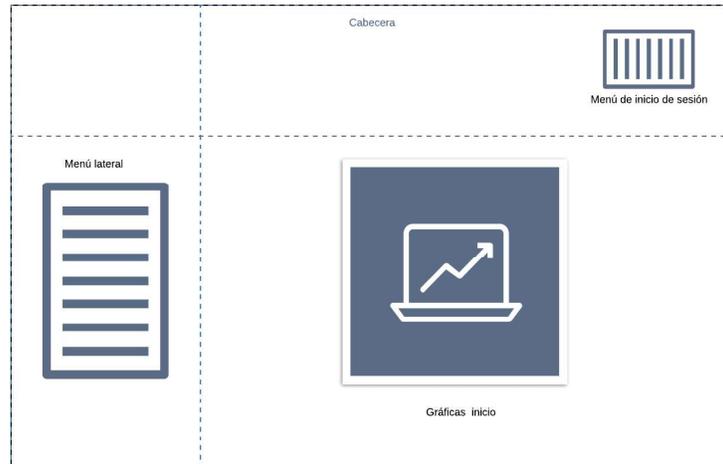


Figura 2.16. Página facturas-panel.php.

2.6.1.3 editar-iva.php

Esta página permite modificar el porcentaje del IVA. En la Figura 2.17 se muestra la página editar-iva.php.

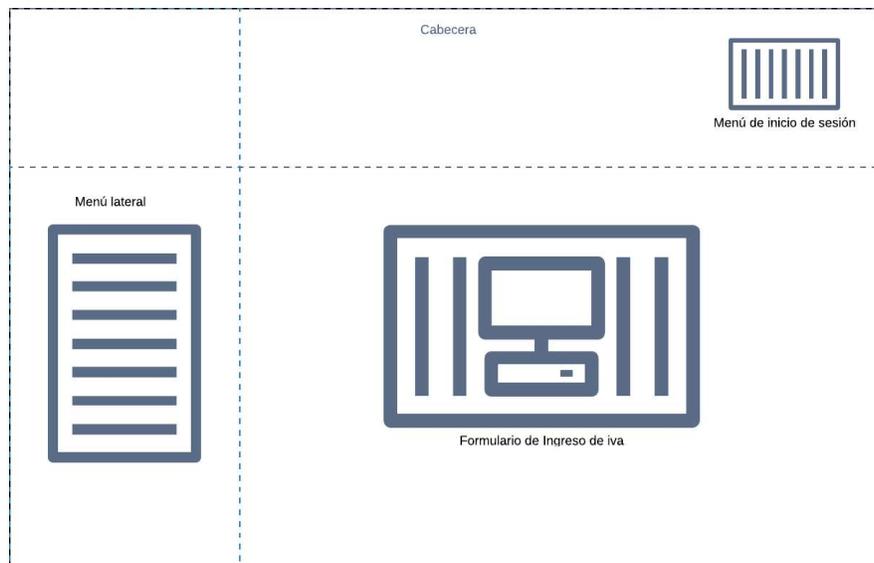


Figura 2.17. Página editar-iva.php.

2.6.1.4 crear-factura.php

Esta página permite ingresar los datos correspondientes a factura. En la Figura 2.18 se muestra la página crear-factura.php.

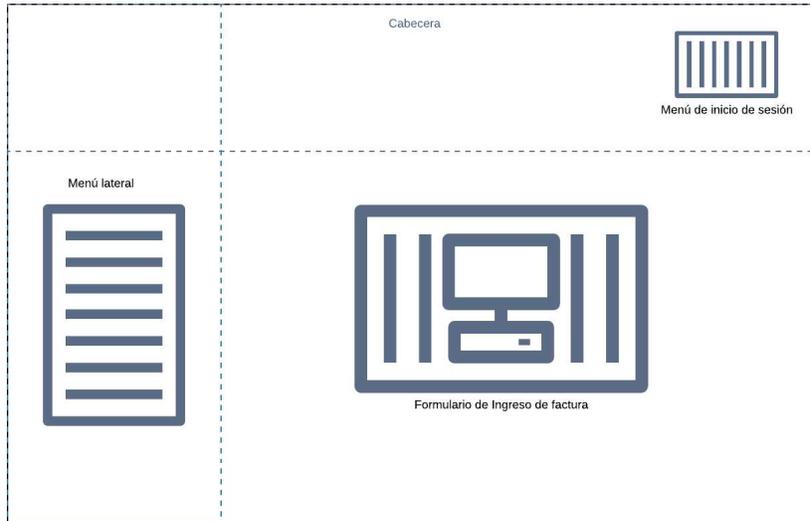


Figura 2.18. Página crear-factura.php.

2.6.1.5 resumen-factura.php

Esta página permite visualizar los datos de la factura recién ingresada. En la Figura 2.19 se muestra la página resumen-factura.php.

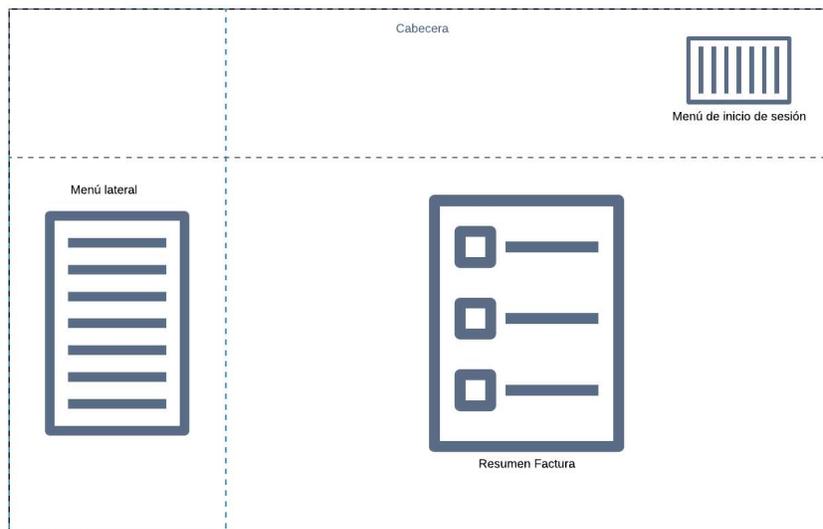


Figura 2.19. Página resumen-factura.php.

2.6.1.6 editar-cliente.php

Esta página permite modificar los datos de la factura ingresada. En la Figura 2.20 se muestra la página editar-cliente.php.

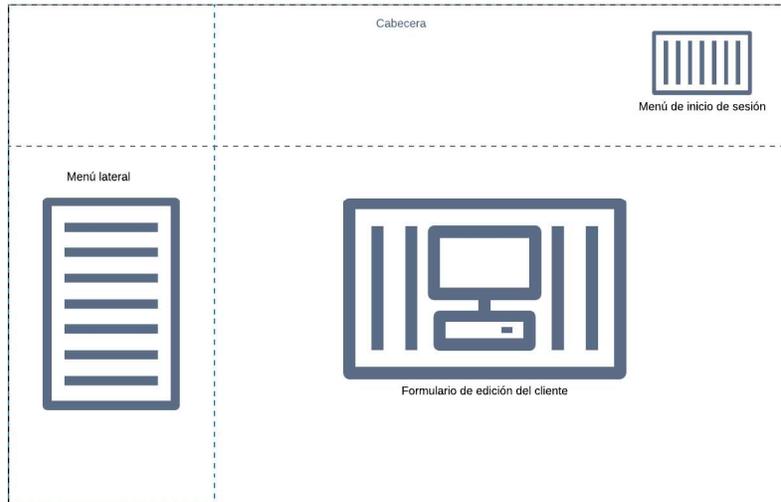


Figura 2.20. Página editar-cliente.php.

2.6.1.7 editar-productos.php

Esta página permite modificar los datos de los productos correspondientes a la factura ingresada. En la Figura 2.21 se muestra la página editar-productos.php.

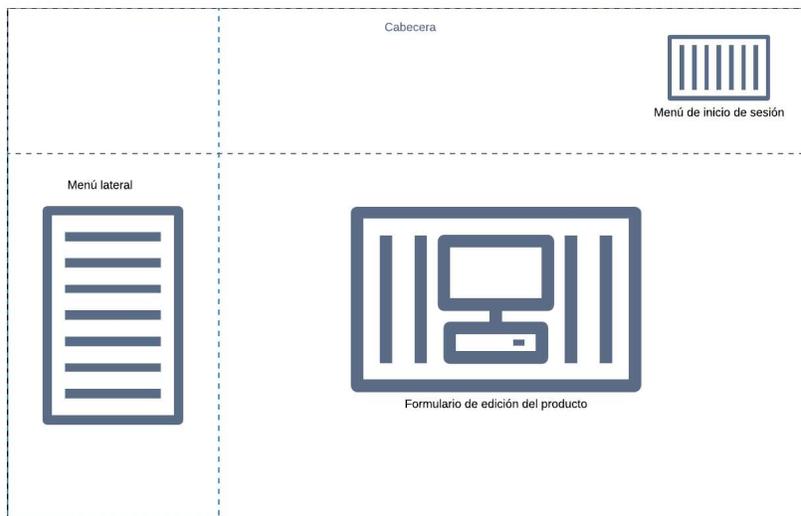


Figura 2.21. Página editar-productos.php.

2.6.1.8 editar-fecha.php

Esta página permite modificar la fecha de la factura ingresada. En la Figura 2.22 se muestra la página editar-fecha.php.

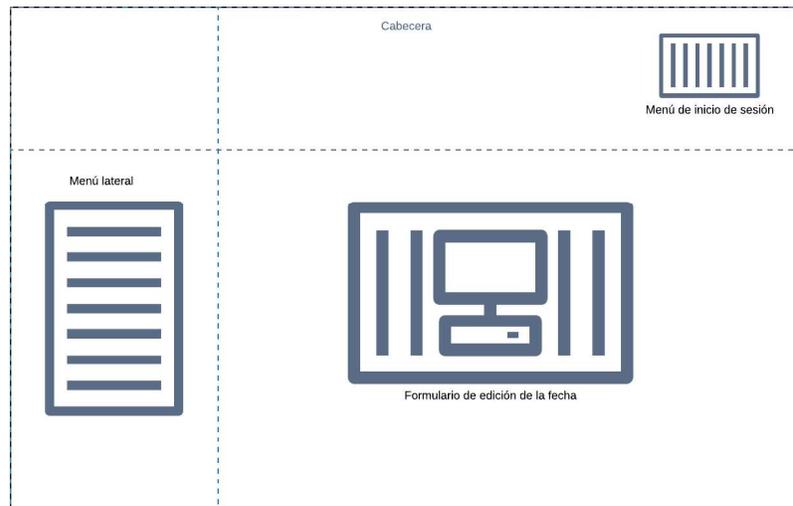


Figura 2.22. Página editar-fecha.php.

2.6.1.9 listar-facturas.php

Esta página permite listar todas las facturas ingresadas. En la Figura 2.23 se muestra la página listar-facturas.php.

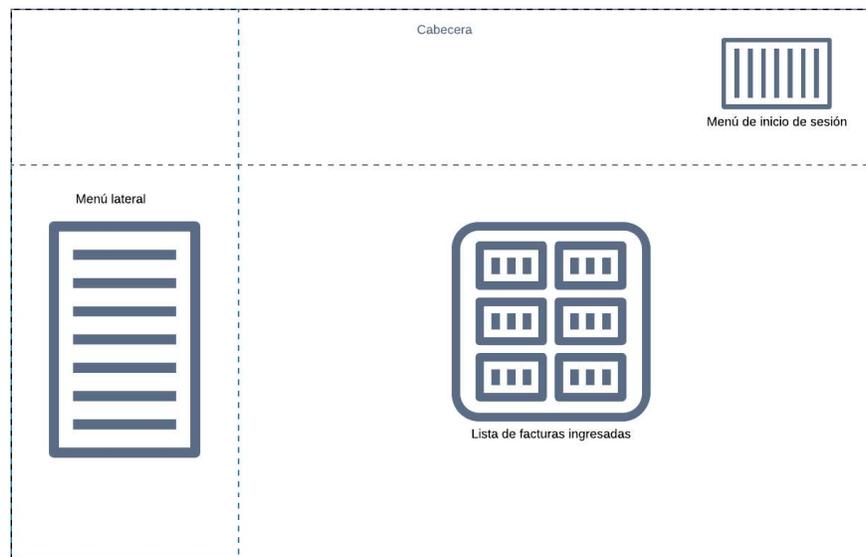


Figura 2.23. Página listar-facturas.php.

2.6.1.10 detalle-factura.php

Esta página permite visualizar a detalle la factura ingresada. En la Figura 2.24 se muestra la página detalle-factura.php.

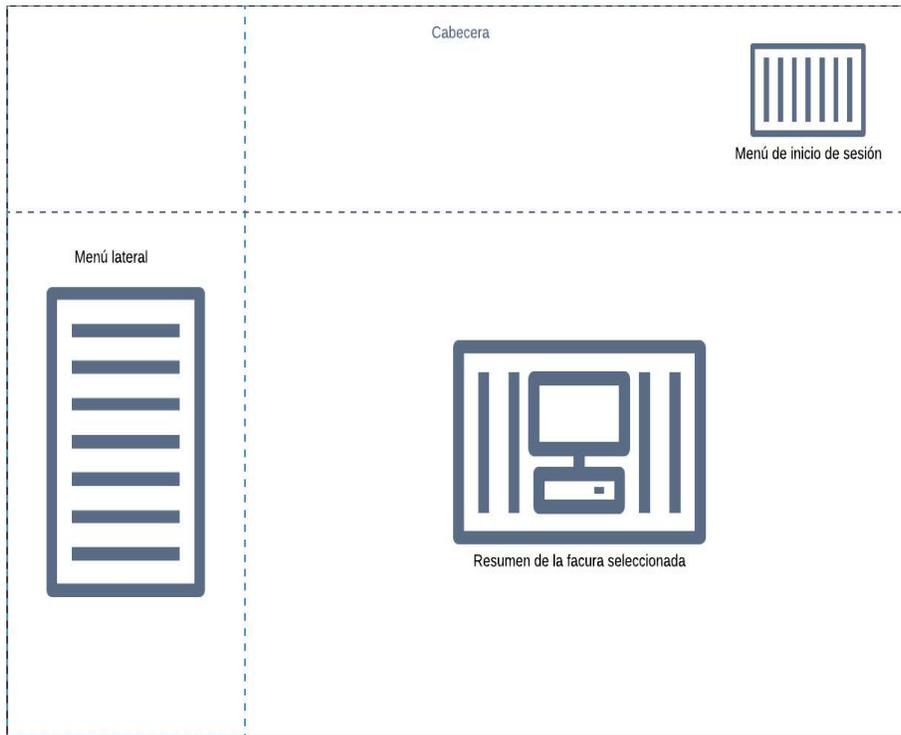


Figura 2.24. Página detalle-factura.php.

El diseño visual web de todas las páginas del área de administrador se realizó con la herramienta Admin Lte que está construida con Bootstrap 3. El diseño web utiliza sus propias hojas de estilo de CSS3, widgets y aplicaciones en JavaScript, solo se adapta dicho diseño a las necesidades y exigencias del sistema web.

2.6.2 MÓDULO DE VENTA EN LÍNEA

2.6.2.1 Usuario Administrador

En la Figura 2.25 se observa el diagrama hipertexto del módulo de venta en línea para el usuario administrador, en donde se debe empezar por la unidad de Login para iniciar sesión, luego navegar dentro del módulo pudiendo observar y modificar las ventas realizadas.

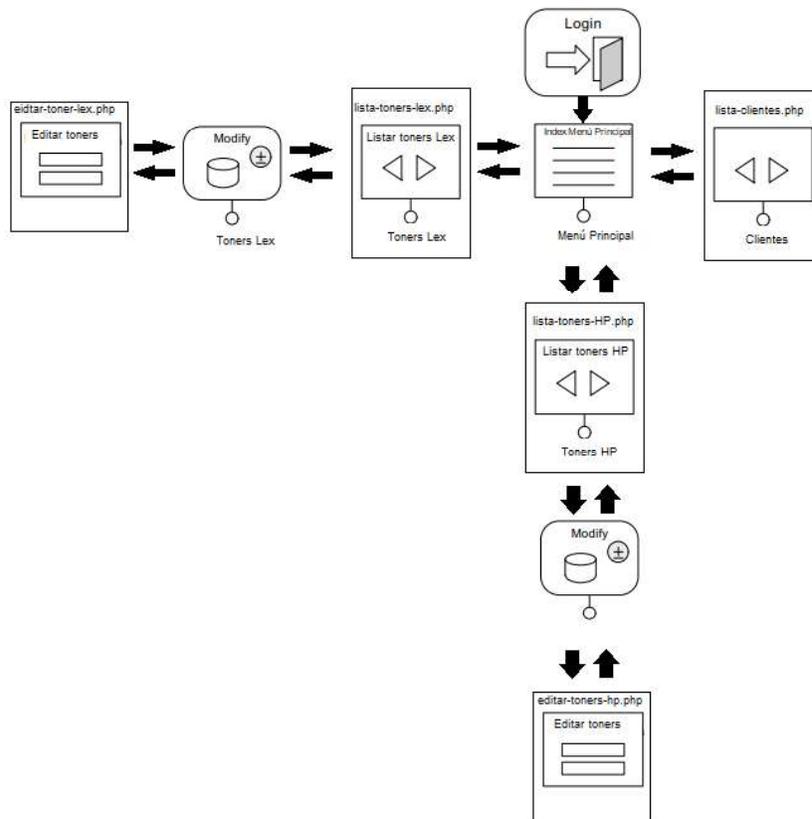


Figura 2.25. Diagrama hipertexto del módulo de venta en línea para el administrador.

2.6.2.1.1 lista-clientes.php

Esta página nos permite visualizar a detalle las ventas realizadas, el administrador puede visualizar los datos personales del cliente, la cantidad de productos comprados y el total pagado por la venta. En la Figura 2.26 se muestra la página lista-clientes.php.

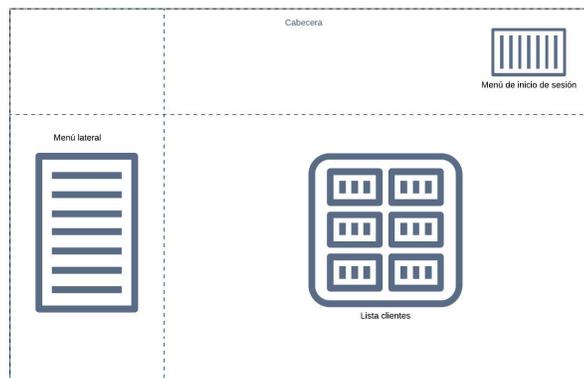


Figura 2.26. Página lista-clientes.php.

2.6.2.1.2 lista-toners-HP.php

Esta página permite visualizar todos los tóners ofertados de la marca HP. En la Figura 2.27 se muestra la página lista-toners-HP.php.

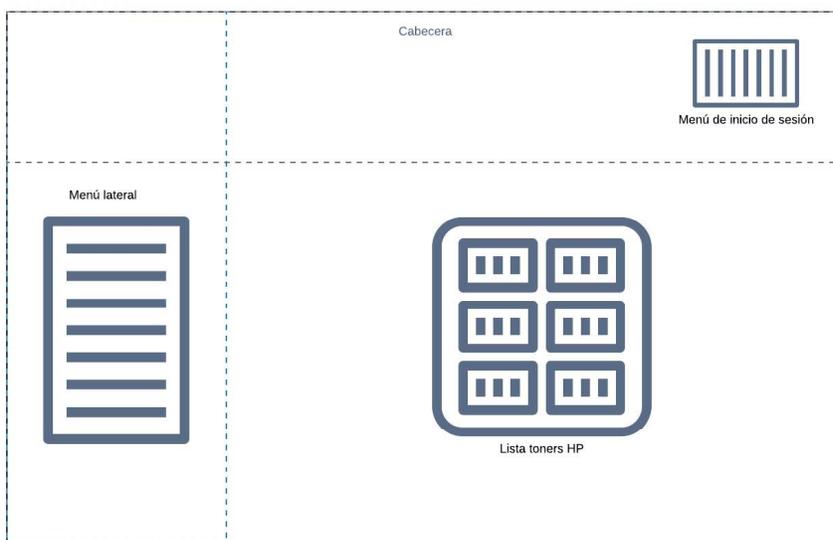


Figura 2.27. Página lista-toners-HP.php.

2.6.2.1.3 editar-toners-hp.php

Esta página permite editar los campos de los tóners de la marca HP. En la Figura 2.28 se muestra la página editar-toners-hp.php.

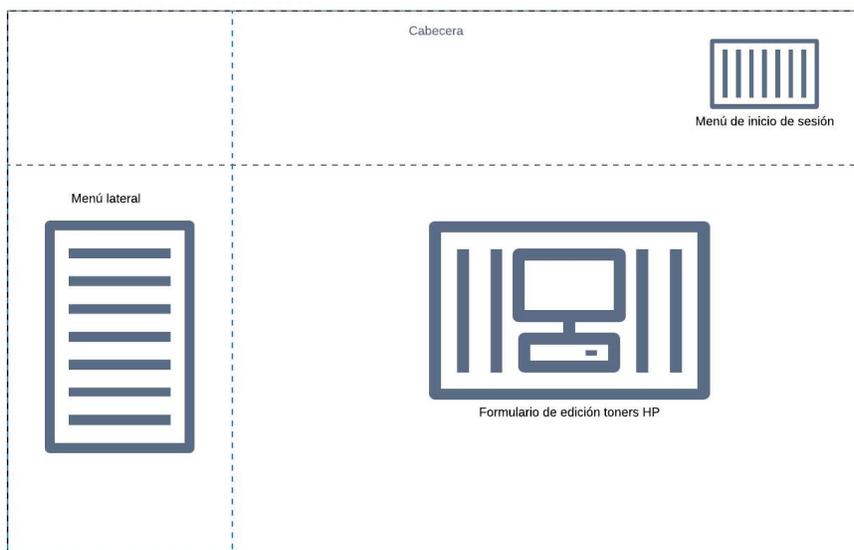


Figura 2.28. Página editar-toners-hp.php.

2.6.2.1.4 lista-toners-lex.php

Esta página nos permite visualizar todos los tóners ofertados de la marca Lexmark dentro la tabla lexmark. En la Figura 2.29 se muestra la página lista-toners-lex.php.

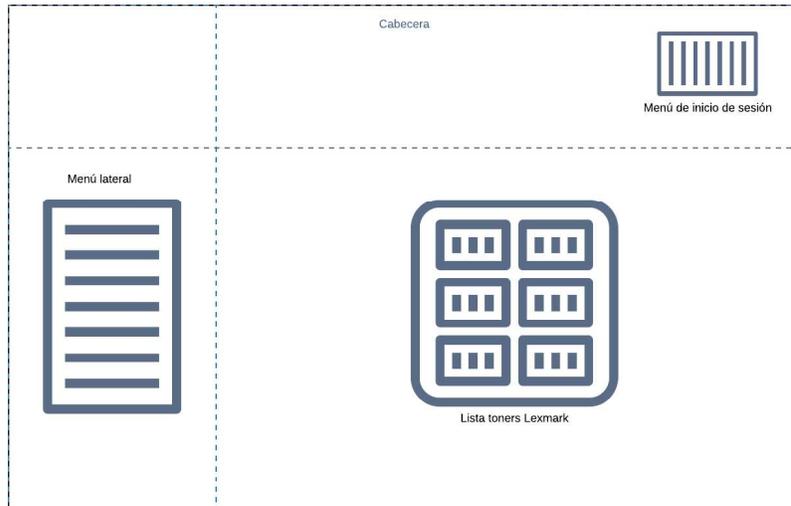


Figura 2.29. Página lista-toners-lex.php.

2.6.2.1.5 editar-toners-lex.php

Esta página permite editar los campos de los tóners de la marca Lexmark. En la Figura 2.30 se muestra la página editar-toners-lex.php.

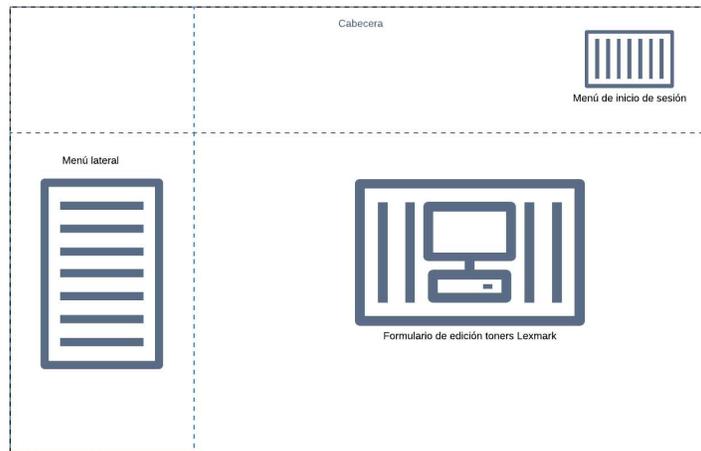


Figura 2.30. Página editar-toners-lex.php.

2.6.2.2 Usuario Cliente

En la Figura 2.31 se observa el diagrama hipertexto del módulo de venta en línea para el usuario cliente, en este caso no es necesario iniciar sesión y el usuario puede navegar por las diferentes páginas. En la página “carrito-compra.php” es donde se realiza la compra de los productos ofertados.

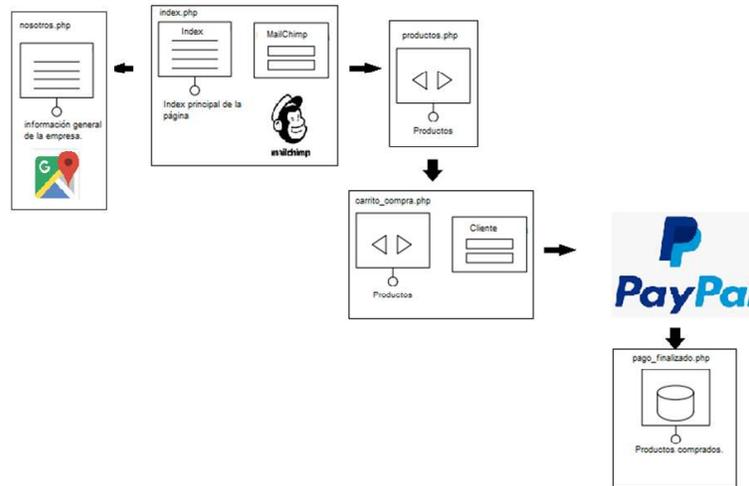


Figura 2.31. Diagrama hipertexto del módulo de venta en línea para el cliente.

2.6.2.2.1 index.php

Esta página es la principal, en ella se encuentra la información de la empresa y se puede suscribir al boletín de la empresa mediante el servicio de Mailchimp. En la Figura 2.32 se muestra la página index.php.

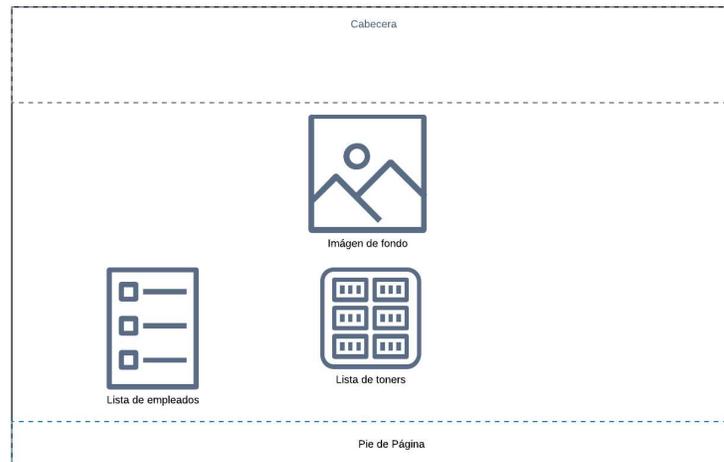


Figura 2.32. Página index.php.

2.6.2.2.2 nosotros.php

Esta página tiene información adicional que requiera el cliente como por ejemplo: ubicación y teléfono de contacto. En la Figura 2.33 se muestra la página nosotros.php.

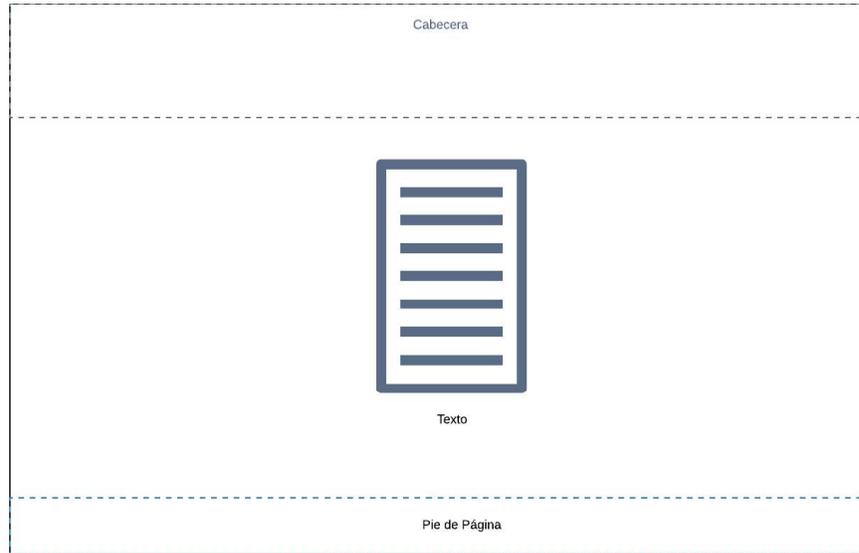


Figura 2.33. Página nosotros.php.

2.6.2.2.3 productos.php

Esta página permite visualizar los tóners ofertados que tiene la empresa. En la Figura 2.34 se muestra la página productos.php.

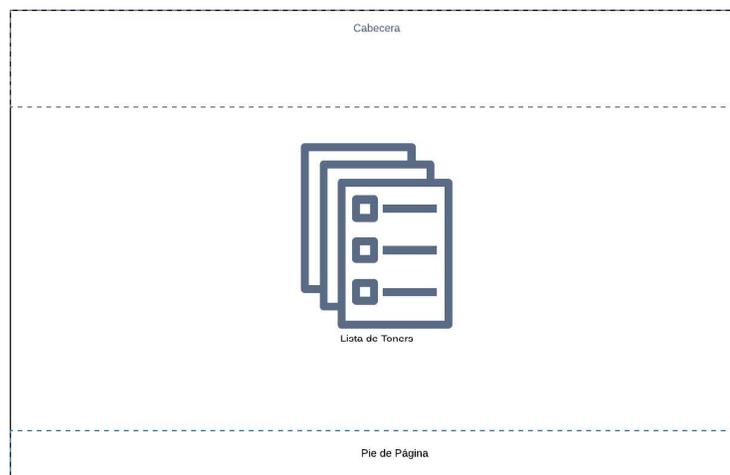


Figura 2.34. Página productos.php.

2.6.2.2.4 carrito_compra.php

Esta página permite elegir los tóners disponibles para realizar la compra. Una vez realizada la compra se redirecciona a la plataforma Paypal con el monto elegido, pudiendo aquí pagar el monto total de los tóners seleccionados. En la Figura 2.35 se muestra la página carrito_compra.php.

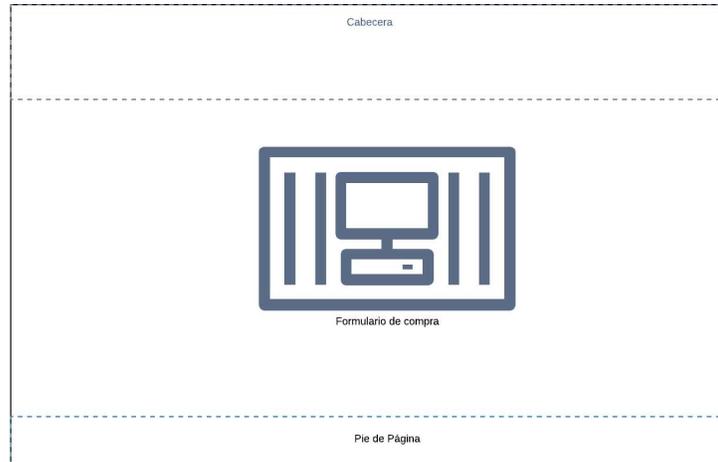


Figura 2.35. Página carrito_compra.php.

2.6.2.2.5 pago_finalizado.php

Una vez realizado el debido pago desde la plataforma Paypal, se redirecciona a la página pago_finalizado.php para que el usuario pueda obtener su ID de compra. En la Figura 2.36 se muestra la página pago_finalizado.php.

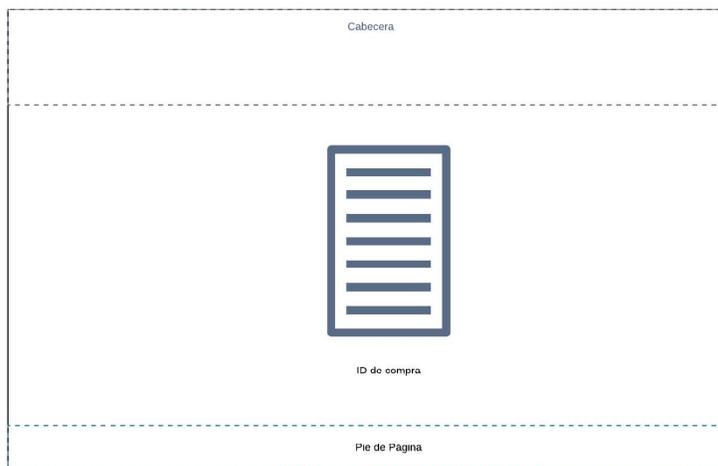


Figura 2.36. Página pago_finalizado.php.

2.6.3 MÓDULO DE MERCADERÍA

En la Figura 2.37 se observa el diagrama hipertexto del módulo de facturación, en el cual primero es necesario iniciar sesión mediante la unidad Login, después se puede navegar por las diferentes páginas del módulo pudiendo añadir, quitar o ingresar inventario.

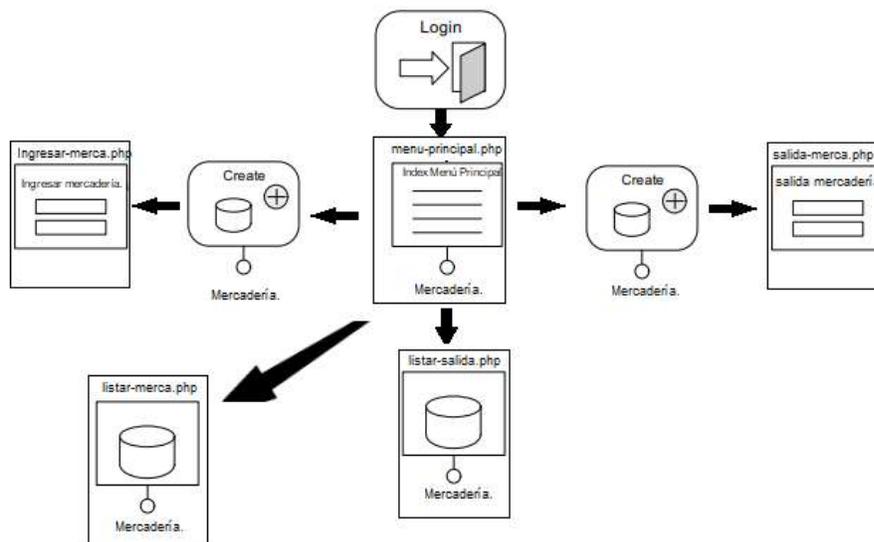


Figura 2.37. Diagrama hipertexto del módulo de mercadería.

2.6.3.1 ingresar-merca.php

Esta página permite ingresar datos de la mercadería. En la Figura 2.38 se muestra la página ingresar-merca.php.

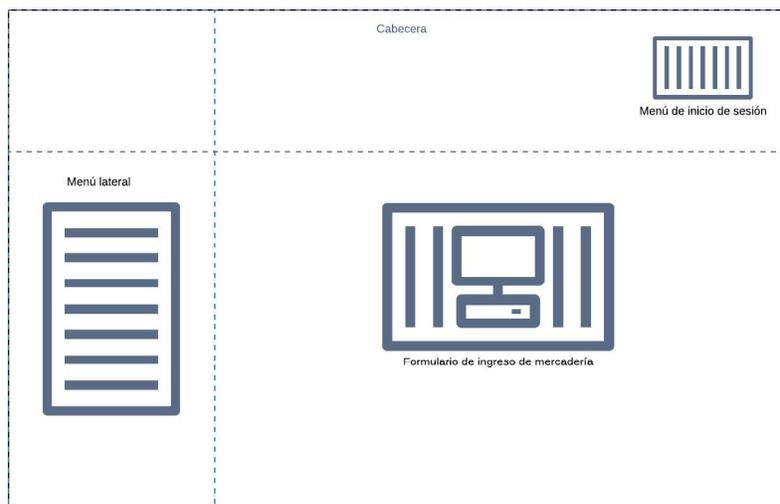


Figura 2.38. Página ingresar-merca.php.

2.6.3.2 salida-merca.php

Esta página permite sacar un producto determinado de inventario en una fecha indicada y la cantidad. En la Figura 2.39 se muestra la página salida -merca.php.

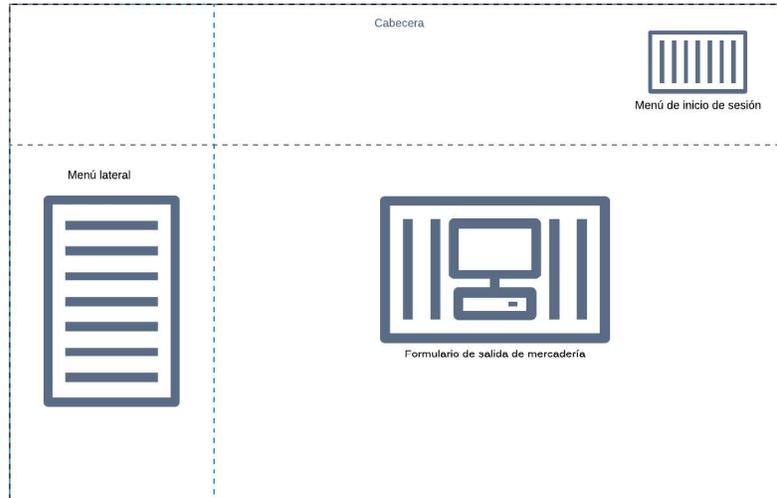


Figura 2.39. Página salida-merca.php.

2.6.3.3 listar-salida.php

Esta página permite visualizar los datos de la salida de mercadería (tipo de producto, fecha, cantidad). En la Figura 2.40 se muestra la página listar-salida.php.

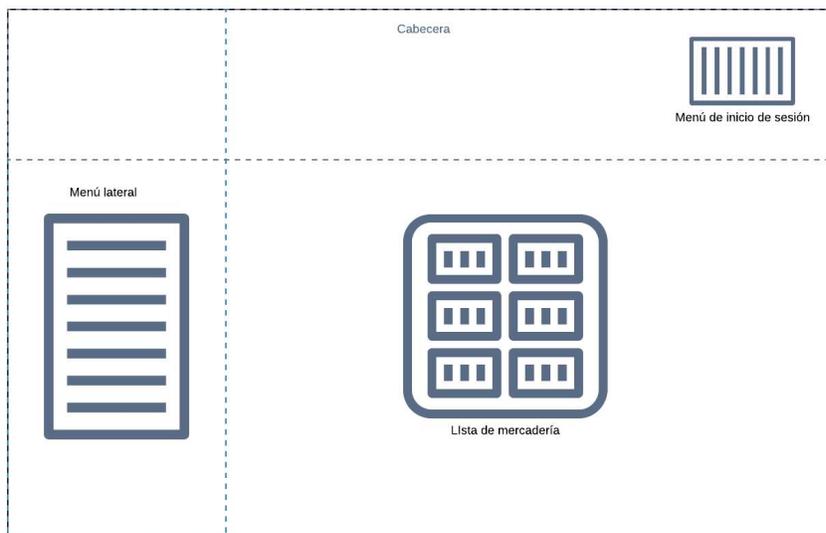


Figura 2.40. Página listar-salida.php.

2.6.3.4 listar-merca.php

Esta página permite visualizar los datos de mercadería en la base de datos. En la Figura 2.41 se muestra la página listar-merca.php.

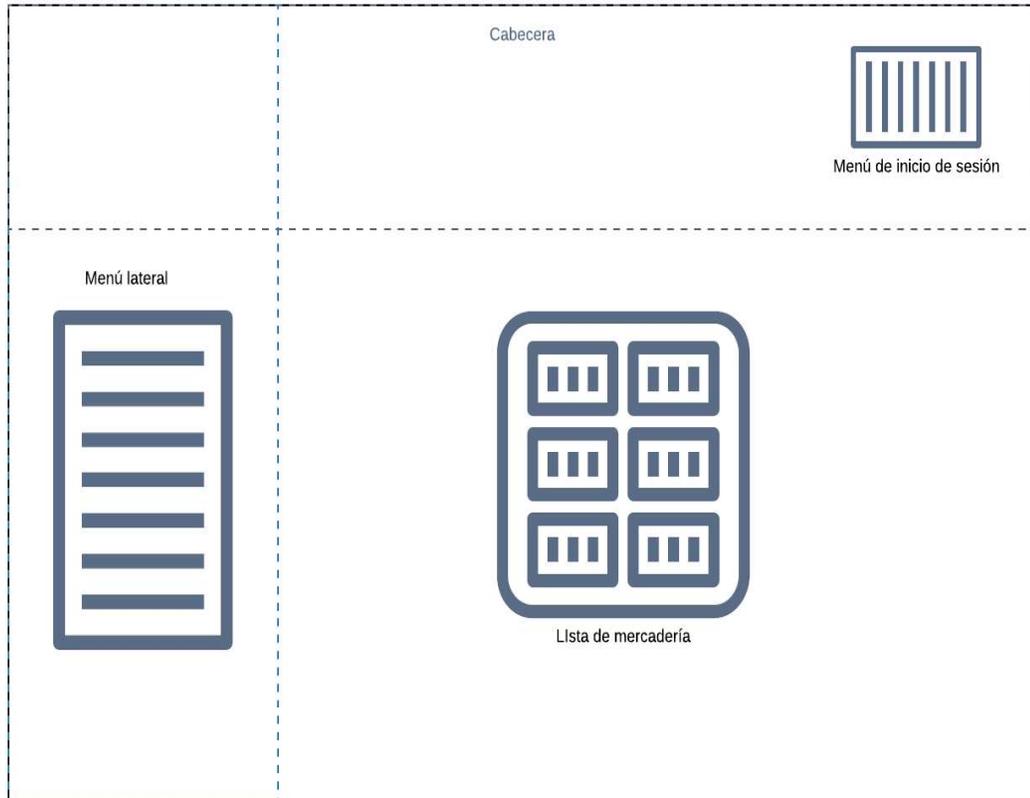


Figura 2.41. Página listar-merca.php.

2.7 IMPLEMENTACIÓN

2.7.1 BASE DE DATOS

Para la codificación de la base de datos se utilizó el servidor de base de datos MySQL y el diagrama relacional creado en la sección 2.4.

En el segmento de código 2.1 se muestra un ejemplo de la creación de la tabla factura, con su respectiva clave primaria y con dos claves foráneas.

```

1 CREATE TABLE `factura` (
2   `ID_factura` int(11) NOT NULL,
3   `ID_cliente` int(11) NOT NULL,
4   `Numero_factura` int(11) NOT NULL,
5   `Fecha_emitida` date NOT NULL,
6   `ID_IVA` int(11) NOT NULL,
7   `Subtotal` float NOT NULL,
8   `IVA_0` float(10,7) NOT NULL,
9   `IVA_12` decimal(10,7) NOT NULL,
10  `Total` float NOT NULL,
11  `Pagado` int(1) NOT NULL,
12  `Anulado` int(1) NOT NULL,
13  `Fecha_de_caducidad` date NOT NULL,
14  `Fecha_edicion` datetime NOT NULL
15 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
16
17 ALTER TABLE `factura`
18 ADD PRIMARY KEY (`ID_factura`),
19 ADD KEY `ID_cliente` (`ID_cliente`),
20 ADD KEY `ID_IVA` (`ID_IVA`);
21 ALTER TABLE `factura`
22 MODIFY `ID_factura` int(11) NOT NULL AUTO_INCREMENT;
23
24 ALTER TABLE `factura`
25 ADD CONSTRAINT `factura_ibfk_1` FOREIGN KEY (`ID_cliente`) REFERENCES `cliente` (`ID_cliente`),
26 ADD CONSTRAINT `factura_ibfk_2` FOREIGN KEY (`ID_IVA`) REFERENCES `iva` (`ID_IVA`);

```

Segmento de Código 2. 1. Creación tabla factura.

2.7.2 LOGIN

Para la autenticación del administrador se implementó una tabla llamada administradores como se muestra en la Figura 2.42. En donde se guardan los requisitos necesarios para la autenticación del usuario administrador.



Figura 2.42. Tabla administradores.

El valor de nickname es único en la tabla y no se puede repetir, la contraseña está cifrada y es una cadena de texto para proveer mayor seguridad al sistema web.

Para iniciar sesión con el usuario administrador es necesario implementar la función `session_start ()` y la variable `$_SESSION`, primero se extrae el usuario y le contraseña que el usuario ingresó en el formulario de la página `Login.php` y se las guarda en las variables `usu` y `pass`, después se realiza una consulta a la base de datos para verificar la existencia de ese usuario y extraer la contraseña del usuario existente, luego mediante la función

`password_verify()` se cifra la cadena ingresada por el usuario y se obtiene un hash, después se compara dichos hashes y en caso de ser iguales se guarda en la variable y finalmente se da una notificación al usuario de inicio de sesión correcto, de no existe coincidencia entre los hashes el usuario obtiene una notificación de error y no le permite ingresar al sistema web como se muestra en el segmento de segmento de código 2.2.

```
1 <?php
2 if(isset($_POST['login-administrador'])){
3     $usu = $_POST['usuario'];
4     $pass = $_POST['pass'];
5     try {
6         include_once "funciones/funciones.php";
7         $stmt = $conn->prepare("SELECT * FROM administradores WHERE nickname=?");
8         $stmt->bind_param("s", $usu);
9         $stmt->execute();
10        $stmt->bind_result($id, $nick_usu,$nombre_usu, $pass_usu,$fecha_editado);
11        if($stmt->affected_rows){
12            $probar= $stmt-> fetch();
13            if($probar){
14                if(password_verify($pass,$pass_usu)){
15                    session_start();
16                    $_SESSION['usu']=$nick_usu;
17                    $resp = array(
18                        'respuesta'=>'existe',
19                        'usuario'=>$nombre_usu
20                    );
21                }
22            }
23            else{
24                $resp = array(
25                    'respuesta'=>'pass_incorrecto'
26                );
27            }
28        }
29        else{
30            $resp = array(
31                'respuesta'=>'no_registros'
32            );
33        }
34        $stmt->close();
35        $conn->close();
36    }
37    catch(Exception $e) {
38        Echo "Error!" . $e->getMessage();
39    }
40    die(json_encode($resp));
41 }
```

Segmento de Código 2. 2 Iniciar sesión.

Si el usuario desea cambiar cualquier dato de la tabla administradores, se creó la página `editar-administrador.php` en donde se puede modificar uno o varios campos de la tabla, como se observa la Figura 2.43.

Editar Administrador

Nickname
root

Nombre
Jefferson Benavides

Password
Ingresa una contraseña

Repetir Password
Ingresa una contraseña

Guardar

Figura 2.43. Página editar-administrador.php.

2.7.3 MAILCHIMP

Para utilizar la API de MailChimp es necesario registrarse en MailChimp, después de llevar acabo el registro, se procede a personalizar los formularios necesarios para la suscripción que posteriormente serán agregados al sistema web y finalmente Mailchimp genere el script del formulario automáticamente. En la Figura 2.44 se muestra la creación del formulario que inicia al momento de ingresar a la página index.php y en la Figura 2.45 se muestra la generación del código del formulario de Mailchimp.

Desktop Mobile

Design Fields Content Settings

Email Address

Sus

Text Color

Button Styling

Button text
Sus

Text Color

Background Color

Hover Color

Alignment
Left

Full width

Modal Styling

Generare Code

Figura 2.44. Creación del formulario de suscripción.

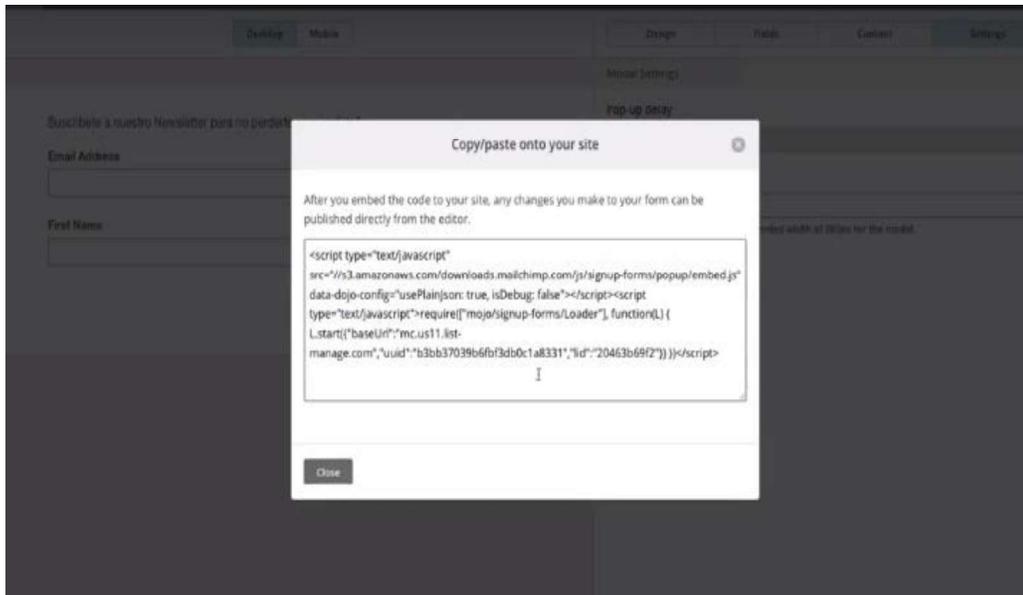


Tabla 2.45. Generación del formulario de Mailchimp.

2.7.4 GOOGLE MAPS API

Para la implementación de Google Maps API es necesario crear un archivo de JavaScript llamado: main.js, en el cual se configura la latitud y longitud y un mensaje de la empresa en la localización elegida. En el segmento de código 2.3 se muestra la API de Google Maps configurada y personalizada.

```

1 var api="AIzaSyCDj0pSmu694d989tb1Si_ezR3rQ1zPUDk";
2 function initMap(){var n={lat:-0.289689,lng:-78.47381},
3 a=new google.maps.Map(document.getElementById("mapa"),{center:n,zoom:15,mapTypeId:google.maps.MapTypeId.ROADMAP}),
4 e=new google.maps.InfoWindow({content:"<h2>Mr. toner</h2><p>Visítanos!</p>"}),
5 i=new google.maps.Marker({position:n,map:a,title:"Mr. Toner"});i.addListener("click",function(){e.open(a,i)})}
6

```

Segmento de Código 2. 3. Google Maps API.

2.7.5 PAYPAL

El SDK de Paypal es un conjunto de archivos que funcionan como una REST API para poder implementar en cualquier sistema web. Es necesario crear dos archivos llamados pagar.php y cotizando.js. El archivo cotizando.js obtiene mediante DOM (Document Object Model) el resumen de compra de todos los tóners elegidos por el cliente, es decir: la cantidad de tóners, el precio de cada tóner y el monto final a pagar como se muestra en el segmento de código 2.4 en donde primero se obtienen los elementos del formulario (nombre, toner seleccionado, precio) mediante la función getElementById(), después se

obtiene la suma total de todos los tóners elegidos por el usuario y finalmente mediante la función `addEventListener()` se calcula el monto total a pagar.

```
1 function() {
2   document.addEventListener('DOMContentLoaded', function(){
3     var nombre= document.getElementById('nombre');
4
5     var toner= document.getElementById('toner');
6     var toner6= document.getElementById('toner6');
7     var toner7= document.getElementById('toner7');
8     var toner8= document.getElementById('toner8');
9
10    var precio1= document.getElementById('precio1');
11
12
13    btnregistro.disabled= true;
14    if(document.getElementById('suma-total')){
15
16      toner.addEventListener('input',calcularPrecios);
17
18      function calcularPrecios(event){
19        event.preventDefault();
20        var Impresora= parseInt(toner.value, 10)|| 0
21        var preciotoner1=parseFloat(precio1.value, 10)|| 0;
22
23        var Total = (Impresora*preciotoner1) +(Impresora*preciotoner2) +(Impresora*preciotoner3);
24
25        var lista_productos =[];
26        if(Impresora>=1){
27          lista_productos.push(Impresora+' '+nombre1.value+' Valor Unitario: '+'$'+preciotoner1);
28        }
29      }
30    }
31  }
32 }
```

Segmento de Código 2. 4. Código archivo cotizando.js.

Después se envían estos datos mediante la petición POST al archivo `pagar.php` en donde se agregan los datos obtenidos por el archivo `cotizando.js` en las clases descritas en la sección 2.5.2.3. En el segmento de código 2.5 se puede observar el código del archivo `pagar.php` en donde se agregan los valores a las diferentes clases, se empieza creando el método de pago que en este caso es: `PayPal`, después mediante el arreglo `arreglo_toners` se guardan todos los tóners seleccionados en la clase `ItemList`, después se agrega la variable `total_pedido` la cual tiene el valor final a pagar en la clase `Amount`, después se agrega la variable `transacción` la cual contiene todas las variables anteriormente mencionadas y se las agrega a las clase `Transaccion`, después se agrega la variable `redireccionar` a la clase `RedirectUrls` para redireccionar al usuario al sistema web después que haya realizado el pago en la plataforma de `PayPal`, y finalmente la variable `pago` ingresa los datos de las variables anteriormente mencionadas en la clase `Payment`.

```

1 <?php
2 $compra = new Payer();
3 $compra->setPaymentMethod("paypal");
4 $i = 0;
5 $arreglo_toners = array();
6 foreach($numero_toners as $key => $value) {
7     if( (int) $value['cantidad'] > 0 ) {
8
9         [{"articulo$i"} = new Item();
10        $arreglo_toners[] = [{"articulo$i"}];
11        [{"articulo$i"}->setName('Toner: ' . $key)
12        [{"articulo$i"}->setCurrency('USD')
13        [{"articulo$i"}->setQuantity( (int) $value['cantidad'] )
14        [{"articulo$i"}->setPrice( (float) $value['precio'] );
15
16        $i++;
17    }
18 }
19 $lista_carrito = new ItemList();
20 $lista_carrito->setItems($arreglo_toners);
21 $cantidad = new Amount();
22 $cantidad->setCurrency('USD')
23 [{"cantidad"}->setTotal($total_pedido);
24
25 $transaccion = new Transaction();
26 $transaccion->setAmount($cantidad)
27 [{"transaccion"}->setItemList($lista_carrito)
28 [{"transaccion"}->setDescription('Pago ')
29 [{"transaccion"}->setInvoiceNumber($ID_regis);
30
31 $redireccionar = new RedirectUrl();
32 $redireccionar->setReturnUrl(URL_SITIO . "/pago_finalizado.php?&id_pago={$ID_regis}")
33 [{"redireccionar"}->setCancelUrl(URL_SITIO . "/pago_finalizado.php?&id_pago={$ID_regis}");
34
35 $pago = new Payment();
36 $pago->setIntent("sale")
37 [{"pago"}->setPayer($compra)
38 [{"pago"}->setRedirectUrls($redireccionar)
39 [{"pago"}->setTransactions(array($transaccion));
40
41 try {
42     $pago->create($apiContext);
43 } catch (PayPal\Exception\PayPalConnectionException $pce) {
44     // Don't spit out errors or use "exit" like this in production code
45     echo '<pre>';print_r(json_decode($pce->getData()));exit;
46 }
47 $aprobado = $pago->getApprovalLink();
48
49 header("Location: {$aprobado}");

```

Segmento de Código 2. 5. Código del archivo pagar.php.

Para la inserción de datos de las tablas correspondientes al módulo de venta en línea desde el cliente, se utilizó la función `bind_param()`, para parametrizar los datos y que no se pueda inyectar código SQL como se muestra en el segmento de código 2.6 en dónde se ingresan datos de la tabla `usuarios_manejo_pago`, con la función `bind_param()` y después de la inserción se cierra la conexión con el servidor de base de datos.

```

1 <?php
2 try {
3     require_once('includes/funciones/cadena_conexion.php');
4     $stmt = $conn->prepare("INSERT INTO usuarios_manejo_pago (Nombre_usu, Apellido_usu, Cedula, Direccion, Telefono, email, Productos_comprados, Fecha_registro, Total_pagado, cancelado) VALUES
5     (?,?,?,?,?,?,?,?,?,?)");
6     $stmt->bind_param("ssssssssdi", $nombre, $apellido, $cedula, $direccion, $celular, $email, $productos_comprados, $fecha,$total_pedido,$cancelado);
7     $stmt->execute();
8     $ID_regis= $stmt ->insert_id;
9     $stmt->close();
10    $conn->close();
11    //header("Location: validar_datos.php?success=1");
12 } catch (Exception $e) {
13     $error = $e->getMessage();
14 }

```

Segmento de Código 2. 6. Código del archivo pagar.php.

Para la verificación del pago efectuado se creó el archivo `pago_finalizado.php` en donde se verifica el estado de pago, como se muestra en el segmento de código 2.7 en donde se obtiene el identificador de la compra realizada mediante la petición GET y después mediante la clase *Payment* se puede obtener los datos necesarios para la verificación del pago, en el arreglo *respuesta* se obtiene el campo *state*, en donde se encuentra la información necesaria para verificar el estado del pago, si este campo tiene el valor de “completed” se alerta al usuario que el pago se realizó correctamente de lo contrario se alerta al usuario que existió algún error en el pago.

```

1 <?php
2     $paymentId = $_GET['paymentId'];
3     $id_pago = $_GET['id_pago'];
4
5     $payment = Payment::get($paymentId, $apiContext);
6     $execution = new PaymentExecution();
7     $execution->setPayerId( $_GET['PayerID'] );
8
9     $result = $payment->execute($execution, $apiContext);
10    echo "<hr>";
11    $respuesta = $result->transactions[0]->related_resources[0]->sale->state;
12
13    if($respuesta == "completed") {
14        echo "<div class='resultado correcto'>";
15        echo "El pago se realizó correctamente <br/>";
16        echo "el ID es {$paymentId}";
17        echo "</div>";
18
19        require_once('includes/funciones/cadena_conexion.php');
20        $stmt = $conn->prepare("UPDATE usuarios_manejo_pago SET cancelado = ? WHERE ID_pagos = ? ");
21        $pagado = 1;
22        $stmt->bind_param("ii", $pagado, $id_pago);
23        $stmt->execute();
24        $stmt->close();
25        $conn->close();
26    }
27    else {
28        echo "<div class='resultado error'>";
29        echo "El pago no se realizo";
30        echo "</div>";
31    }

```

Segmento de Código 2. 7. Código del archivo `pago_finalizado.php`.

2.7.6 EMISIÓN DE FACTURAS

Para la emisión de facturas se creó el archivo `imprimir.php` en donde se decidió utilizar las librerías: *Printer*, *EscposImage*, *FilePrintConnector* y *WindowsPrintConnector*, para imprimir las facturas ingresadas como se explica en [28]. En el código 2.7 se muestra un ejemplo de cómo imprimir con la tecnología PHP.

```

1 <?php
2
3 require __DIR__ . '/../autoload.php';
4 use Mike42\Escpos\Printer;
5 use Mike42\Escpos\PrintConnectors\FilePrintConnector;
6 use Mike42\Escpos\EscposImage;
7 $connector = new FilePrintConnector("http://localhost");
8 $printer = new Printer($connector);
9 /* Initialize */
10 $printer -> initialize();
11 /* Text */
12 $printer -> text("Factura\n");
13 $printer -> cut();
14 /* Line feeds */
15 $printer -> text("$factura");
16 $printer -> feed(?);
17 $printer -> cut();
18 /* Font modes */
19 $modes = array(
20     Printer::MODE_FONT_B,
21     Printer::MODE_EMPHASIZED,
22     Printer::MODE_DOUBLE_HEIGHT,
23     Printer::MODE_DOUBLE_WIDTH,
24     Printer::MODE_UNDERLINE);
25 for ($i = 0; $i < pow(2, count($modes)); $i++) {
26     $bits = str_pad(decbin($i), count($modes), "0", STR_PAD_LEFT);
27     $mode = 0;
28     for ($j = 0; $j < strlen($bits); $j++) {
29         if (substr($bits, $j, 1) == "1") {
30             $mode |= $modes[$j];
31         }
32     }
33     $printer -> selectPrintMode($mode);
34     $printer -> text("ABCDEFGHIIjabcdeFGHIjk\n");
35 }
36 $printer -> cut();
37 /* Pulse */
38 $printer -> pulse();

```

Código 2.7. Código del archivo imprimir.php.

2.7.7 DESPLEGAR EL SISTEMA WEB EN UN SERVIDOR LOCAL

Para el despliegue del sistema web localmente se utilizó el servidor WAMP, el cual es un conjunto de servidores (Apache y MySQL) para el sistema operativo Windows como se muestra en la Figura 2.46.



Figura 2.46. Vista de inicio del servidor WAMP.

Para la ejecución del sistema es necesario mover todos los archivos del sistema web a la carpeta htdocs como se muestra en la Figura 2.47.

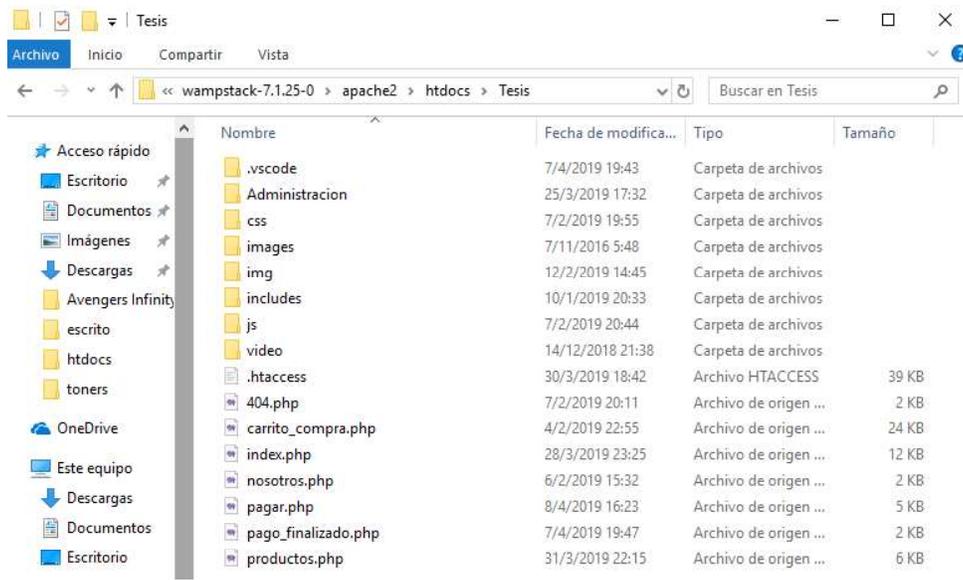


Figura 2.47. Carpeta Tesis.

Para crear la base de datos se usó la herramienta PhpMyAdmin que es un software libre escrito en PHP, para el manejo del servidor MySQL y MariaDB. En la Figura 2.48 se observa el panel de control de PhpMyAdmin.

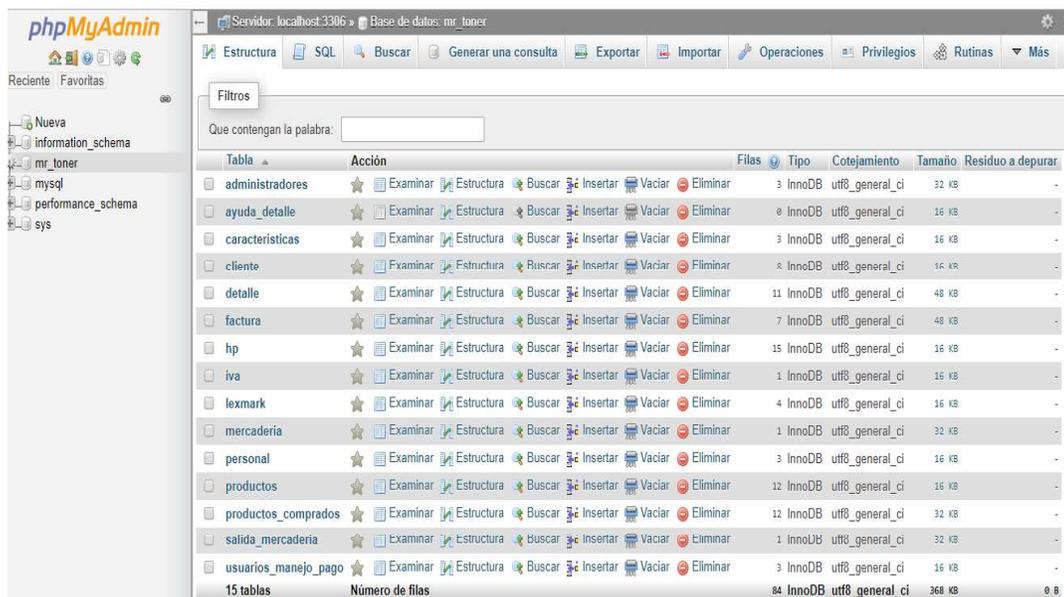


Figura 2.48. Panel de control de PhpMyAdmin.

A continuación, se ingresó al sistema web desde el navegador de Google Chrome. Como se observa en la Figura 2.49 y la Figura 2.50.

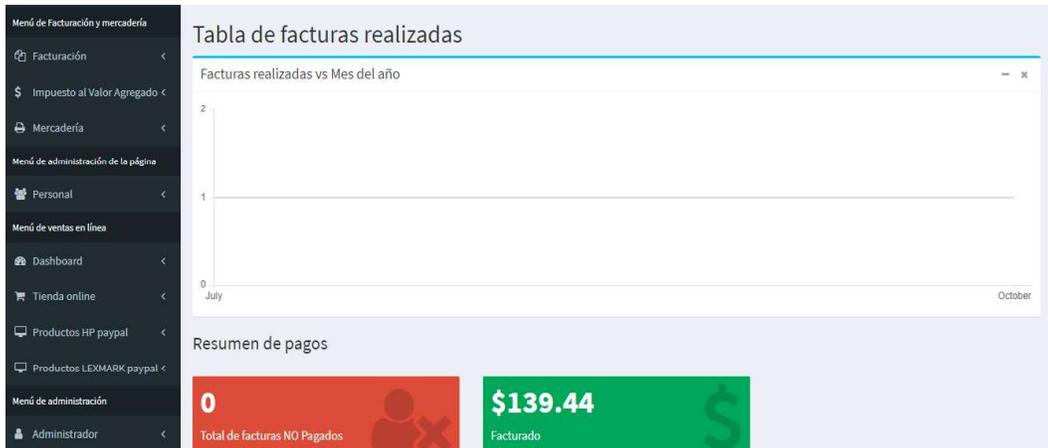


Figura 2.49. Página menú-principal.php.



Figura 2.50. Página index.php.

En la consola del navegador Google Chrome se observa que no existe ningún tipo de error en la página Panel de control, como se muestra en la Figura 2.51.

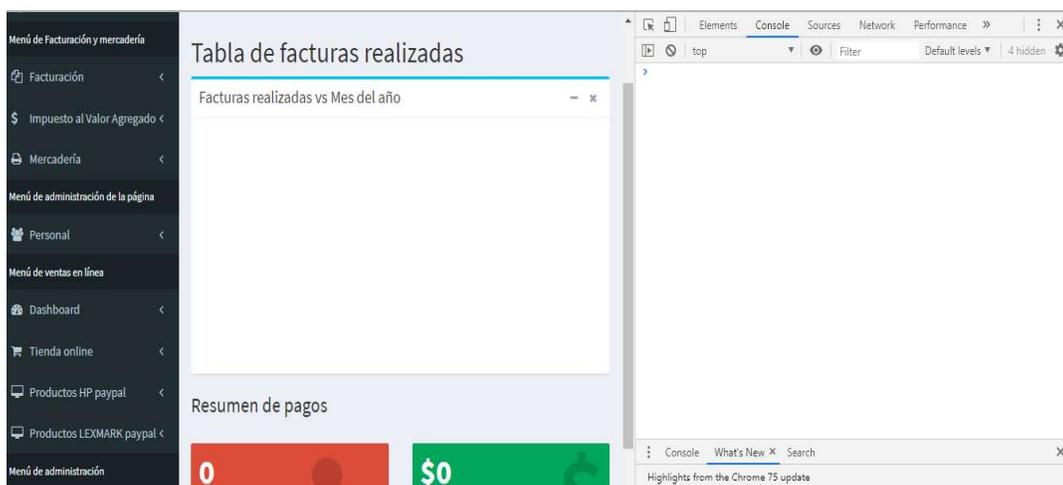


Figura 2.51. Página menú-principal.php.

Se concluye que el código es compatible con el servidor local y no existe ningún tipo de inconveniente.

2.8 INSTALACIÓN

2.8.1 DESPLEGAR EL SISTEMA WEB AL SERVICIO DE AWS

2.8.1.1 Instancia RDS

Mediante el uso del servicio de EC2 de Amazon Web Services, se realizó la implementación de la máquina virtual con el sistema operativo Amazon Linux [29] que albergará el sistema web prototipo como se muestra en la Figura 2.52.

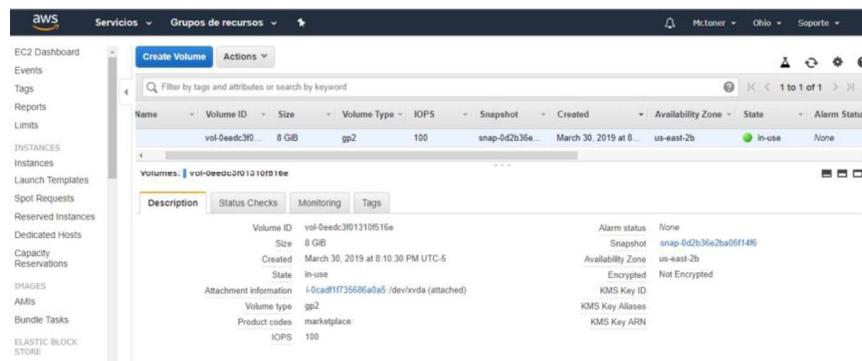


Figura 2.52. Descripción de la máquina virtual EC2.

2.8.1.2 Instalación del servidor Apache

Se instaló el servidor web Apache en la máquina virtual con los pasos que se sugieren en [30]. En la Figura 2.53 se muestra la correcta instalación.

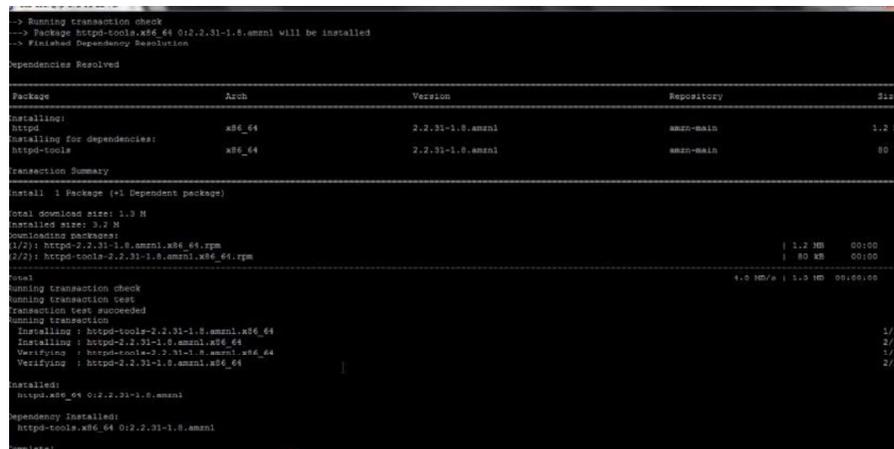


Figura 2.53. Instalación correcta del servidor Apache.

2.8.1.3 Instalación del servidor de base de datos

Se instaló el servidor MySQL en la máquina virtual con los pasos que se sugieren en [30]. En la Figura 2.54 se muestra la correcta instalación.

```
XAMPP will be installed to /opt/lampp
Press [Enter] to continue:

-----
Setup is now ready to begin installing XAMPP on your computer.

Do you want to continue? [Y/n]: yes

-----
Please wait while Setup installs XAMPP on your computer.

Installing
0% _____ 50% _____ 100%
#####

-----
Setup has finished installing XAMPP on your computer.

root@ip-172-31-45-155:/home/ubuntu# sudo /opt/lampp/lampp start
Starting XAMPP for Linux 1.8.3-5...
XAMPP: Starting Apache...already running.
XAMPP: Starting MySQL...ok.
XAMPP: Starting ProFTPD...ok.
```

Figura 2.54. Instalación correcta del servidor MySQL.

2.8.2 INSTALAR EL SISTEMA WEB EN LA MÁQUINA VIRTUAL

Mediante la herramienta WinSCP se subió el sistema web a la máquina virtual recién creada. Todos los archivos fueron almacenados en la carpeta var/www/html/Tesis/ como se muestra en la Figura 2.55 y se creó una base de datos como se muestra en la Figura 2.56.

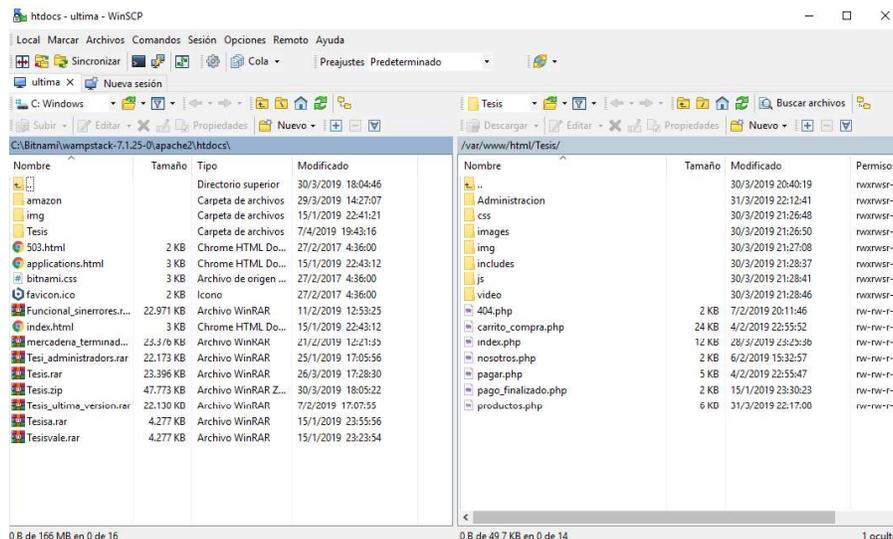


Figura 2.55. Implementación del sistema web en línea.

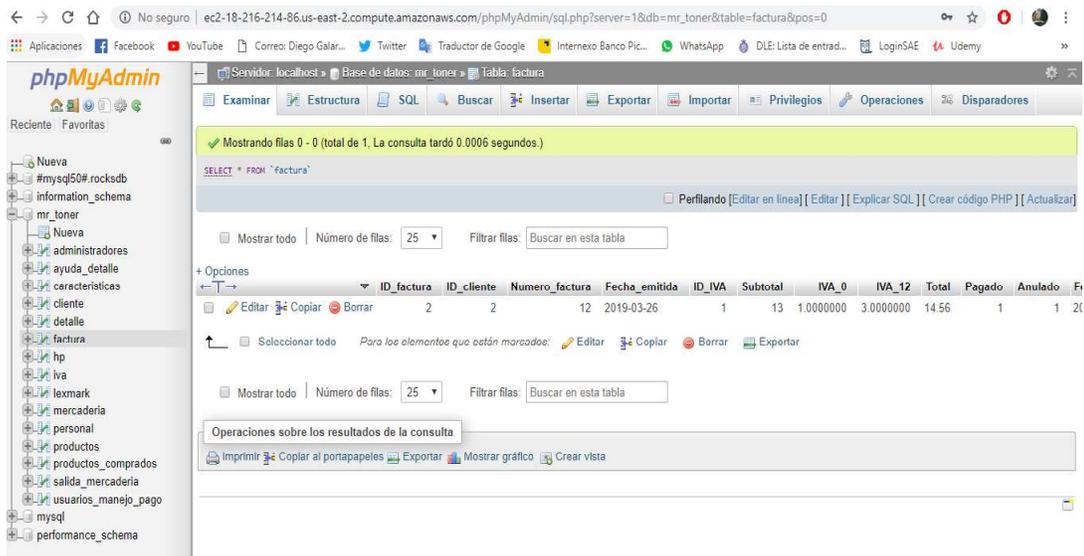


Figura 2.56. Implementación de la base de datos en línea.

2.8.3 SISTEMA WEB IMPLEMENTADO EN AWS

Una vez finalizada la creación de la instancia, se procedió a iniciar todos los servicios (web y base de datos) para la instalación del sistema web como se muestra en la Figura 2.57 en donde se puede ver la página principal (index.php) funcionando.

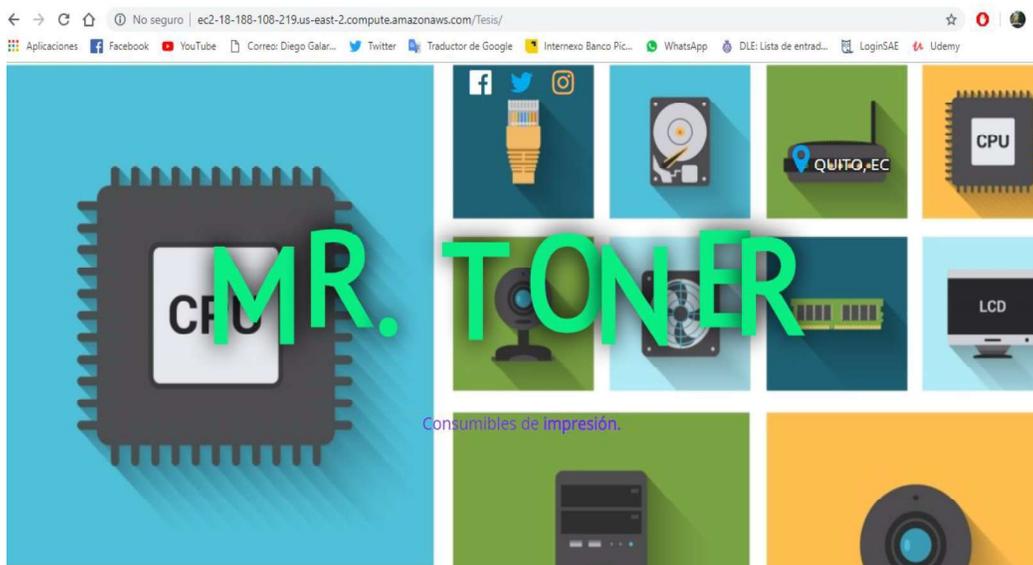


Figura 2.57. Página de inicio del sistema web.

3. RESULTADOS Y DISCUSIÓN

3.1 PRUEBAS Y EVALUACIÓN

Una vez llevado a cabo la fase de implementación del sistema web localmente, se procedió a realizar las respectivas pruebas que permitieron concretar el correcto funcionamiento del sistema web. Teniendo en cuenta los siguientes aspectos [31]:

- Contenido.
- Interfaz de usuario.
- Navegación.
- Compatibilidad.

3.1.1 PRUEBAS DE CONTENIDO

Las pruebas de contenido están destinadas a corregir errores de tipo semántico, ortográfico entre otros, debido a que puede existir incompatibilidad con los idiomas del texto en los servidores. Los servidores están instalados con el lenguaje inglés por defecto y no permiten el uso de los diferentes rasgos que se colocan en las letras de las palabras en español (tildes, diéresis, etc.). Para esto se realizó la codificación de la función *html_entity_decode()* en todas las consultas hacia la base de datos, permitiendo traducir las entidades HTML a sus caracteres correspondientes [32]. En la Figura 3.1 se muestra la incompatibilidad mencionada

Opciones	ID_cliente	Nombre_cliente	Cedula_ruc	Direccion_cliente	Telefono_cliente	Correo_electronico	Fecha_edicion
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	Diego Enrique Galarza PazmiÁ±o	1715898647	AscÁ±zubi, 29 de mayo E4-127	984358674	raptor100500@hotmail.com	2001-01-23
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	2	Diego Enrique Galarza PazmiÁ±o	1715898647	AscÁ±zubi, 29 de mayo E4-127	984358674	raptor100500@hotmail.com	2001-01-23
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	3	Diego Enrique Galarza PazmiÁ±o	1715898647	AscÁ±zubi, 29 de mayo E4-127	984358674	raptor100500@hotmail.com	2001-01-23
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	4	Diego Enrique Galarza PazmiÁ±o	1715898647	AscÁ±zubi, 29 de mayo E4-127	984358674	raptor100500@hotmail.com	2001-01-23

Figura 3. 1 Error de compatibilidad en los campos Nombre_cliente y Direccion_cliente.

Prueba: Visualización en la página listar-facturas.php

Procedimiento:

1. Ingresar una factura.
2. Visualizar los datos ingresados en la página listar-facturas.php (Ver figura 3.2)

Tabla para la administración de Facturas

Buscar:

Razón social	Número de factura	Total	Fecha de compra	Fecha límite de pago	Días restantes	Detalle Factura
Diego Enrique Galarza Pazmiño Cancelado	12345	45.48	2019-07-08	2019-07-08	Cancelado	
Diego Enrique Galarza Pazmiño No ha pagado	1235	46.92	2019-07-10	2019-07-25	+11 días No caducado	
Razón social	Número de factura	Total de la compra	Fecha de compra	Fecha límite de pago	Días restantes	Detalle Factura

Mostrando 1 a 2 de 2 resultados

Anterior **1** Siguiente

Figura 3. 2. Resumen de facturas ingresadas.

Resultado: Se corrige el error de incompatibilidad en el campo razón social.

Prueba: Visualización en la página resumen-factura.php

Procedimiento:

1. Ingresar una factura.
2. Visualizar los datos ingresados en la página resumen-factura.php (Ver figura 3.3)

Factura ingresada

Nombre cliente	Cédula o RUC	Dirección	Teléfono	Correo	Acción
Diego Enrique Galarza Pazmiño	1715891235	Ascázubi, 29 de mayo E4-127	984358674	raptor100500@hotmail.com	

Descripción producto	Valor unitario	Cantidad	Valor Total	Acción
Toner 1	13.00	3	39.000	

Fecha emitida	Fecha de caducidad	Acción
2019-07-09	2019-07-09	

39 Sub total	1 IVA 0	4.56 IVA 12	43.56 Total
------------------------	-------------------	-----------------------	-----------------------

Figura 3. 3. Resumen de factura ingresada.

Resultado: Se corrige el error de incompatibilidad en el campo razón social y Dirección.

3.1.2 PRUEBAS DE INTERFAZ DE USUARIO

Las pruebas de interfaz de usuario mejoraron la navegación e interacción del usuario con el sistema web, permitiendo adecuar:

- El mejor tamaño y tipo de texto de tal manera que el usuario no tenga problemas al leer.
- Imágenes de fondo.
- Colores de fondo y texto.

En las pruebas de interfaz de usuario se determina si el despliegue de las páginas es el correcto.

Prueba: Navegación de algunas páginas del sistema web.

Procedimiento:

1. Ingreso a la página index.php (ver Figura 3.4).
2. Navegar a otras páginas del sistema web mediante la cabecera e hiperenlaces dentro de la página index.php.
3. Verificar el despliegue de toda la información de las páginas sea el correcto.

Resultado: Correcto navegación del sistema web.

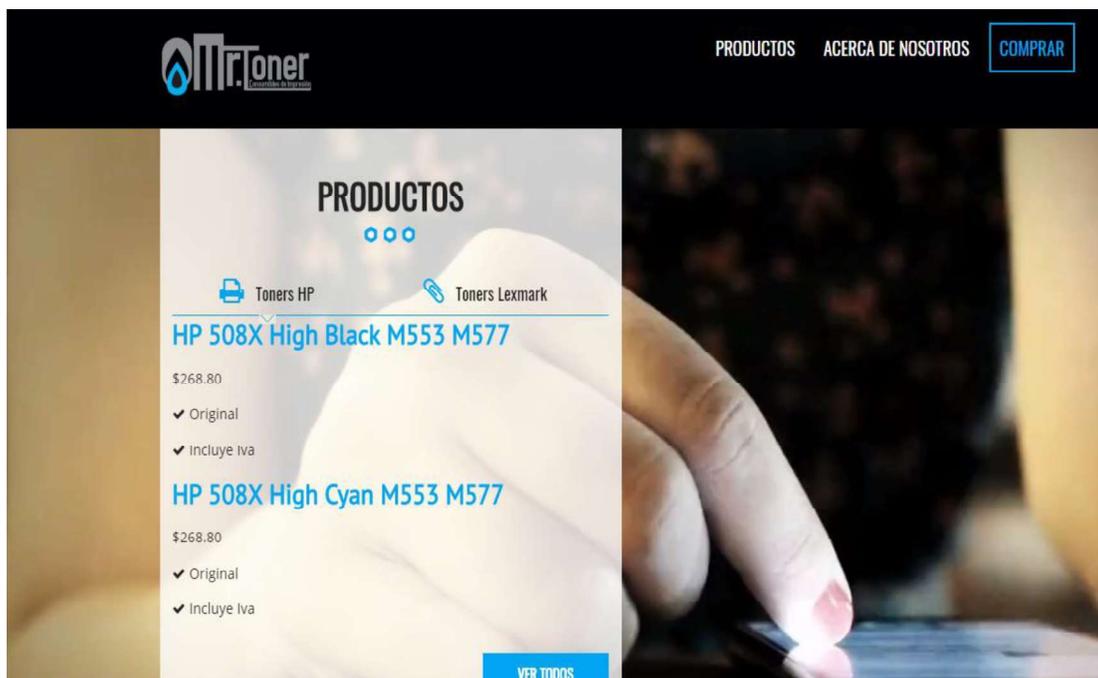


Figura 3. 4. Página index.php desplegada en el navegador Google Chrome.

Prueba: Interacción de algunas páginas del sistema web.

Procedimiento:

1. Ingreso a la página index.php (ver Figura 3.5).
2. Interactuar con elementos dentro de las páginas del sistema web.
3. Verificar el despliegue de toda la información de las páginas sea el correcto.

Resultado: Correcto navegación del sistema web.

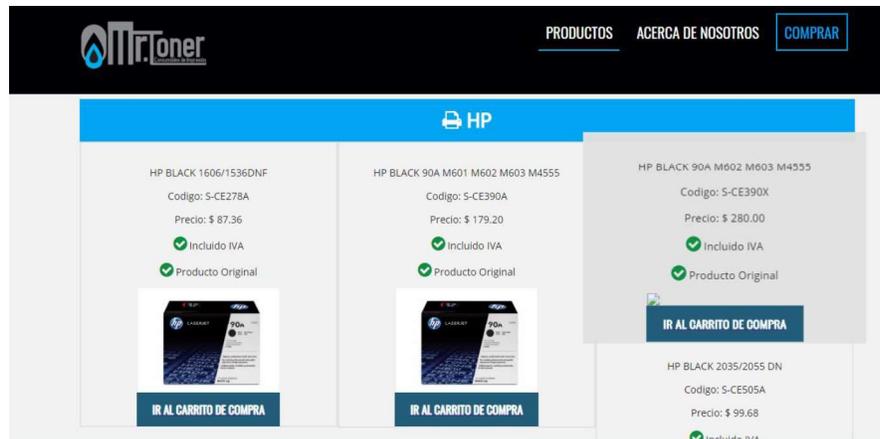


Figura 3. 5. Interacción con el menú de la página productos.php

3.1.3 PRUEBAS DE NAVEGACIÓN

El sistema web es muy intuitivo y fácil de usar, tanto para el cliente como para el administrador. Todo el menú del panel de administración del sistema web está disponible y solamente se necesita que el administrador provea la información necesaria para ingresar los campos de las tablas de la base de datos.

Por otro lado, el cliente puede comprar tóners sin mayor dificultad. Las pruebas de navegación consisten en que el usuario navegue de tal forma que pueda acceder a los datos de una manera intuitiva y fácil de usar.

Prueba: Despliegue de las páginas del sistema web.

Procedimiento:

1. Ingresar al panel de control del sistema web (ver figura 3.6).
2. Visualizar el menú lateral del panel de control.
3. Acceder a todas las páginas web mediante el menú lateral.

Nombre cliente	Cédula o RUC	Dirección	Teléfono	Correo	Acción
Diego Enrique Galarrza Pazmillo	1715989647	Asiububi, 29 de mayo E4-127	984329674	raptor10550@gmail.com	

Descripcion producto	Valor unitario	Cantidad	Valor Total	Acción
aa	13,00	1	13,000	
aa	13,00	1	13,000	

Fecha emitida	Fecha de caducidad	Acción
2019-04-30	2019-04-30	

26 Sub total	2 IVA 0	2.88 IVA 12	28.88 Total
------------------------	-------------------	-----------------------	-----------------------

Figura 3. 6 Página resumen-factura.php.

Resultado: El menú lateral ayuda a mejorar la experiencia del usuario al ingresar al panel de control debido a que puede acceder a todas las paginas rápidamente.

3.1.4 PRUEBAS DE COMPATIBILIDAD

Las pruebas de compatibilidad se las realizaron en tres diferentes tipos de navegadores, siendo estos: Google Chrome, Mozilla Firefox y Edge. Las pruebas de compatibilidad permiten al usuario comprobar si el sistema web funciona de manera correcta y no tiene diferentes comportamientos o errores en los diferentes navegadores web.

Prueba: Compatibilidad con el navegador Google Chrome.

Procedimiento:

1. Navegar a las diferentes páginas del sistema web.
2. Corroborar que las páginas del sistema web no tiene comportamientos inusuales en el navegador.

Resultado: En la Figura 3.7 se muestra que la página carrito_web.php es compatible en un 100%, no existen fallos o errores de consola que puedan perjudicar al sistema web.

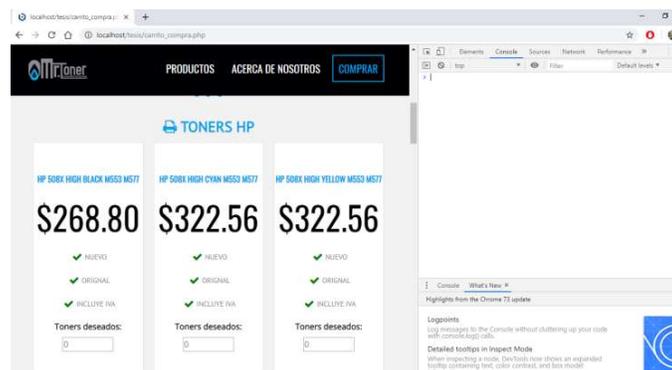


Figura 3. 7 Sistema web en el navegador Google Chrome.

Prueba: Compatibilidad con el navegador Firefox.

Procedimiento:

1. Navegar a las diferentes páginas del sistema web.
2. Corroborar que las páginas del sistema web no tiene comportamientos inusuales en el navegador.

Resultado: En la Figura 3.8 se muestra que la página carrito_web.php es compatible en un 100%, no existen fallos o errores de consola que puedan perjudicar al sistema web, con la diferencia que en este navegador las flechas de los casilleros (aumentar el número de tóners deseados) aparece por defecto y no cuando se le da clic al tóner.

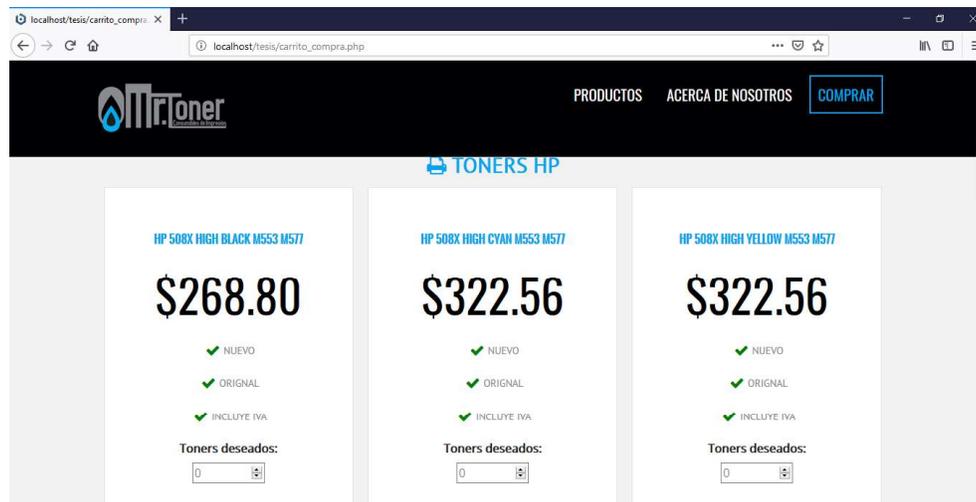


Figura 3. 8 Sistema web en el navegador Mozilla Firefox.

Prueba: Compatibilidad con el navegador Edge.

Procedimiento:

1. Navegar a las diferentes páginas del sistema web.
2. Corroborar que las páginas del sistema web no tiene comportamientos inusuales en el navegador.

Resultado: En la Figura 3.8 se muestra que la página carrito_web.php es compatible en un 100%, no existen fallos o errores de consola que puedan perjudicar al sistema web, con la diferencia que en este navegador las flechas de los casilleros (aumentar el número de tóners deseados) aparece por defecto y no cuando se le da clic al tóner.

En la Figura 3.9 se muestra que el sistema web es compatible en un 100%, no existen fallos o errores de consola que puedan perjudicar al sistema web, con la diferencia que en

este navegador las flechas de los casilleros (aumentar el número de tóners deseados) no aparecen y al cliente le toca ingresar manualmente el número de tóners a comprar.

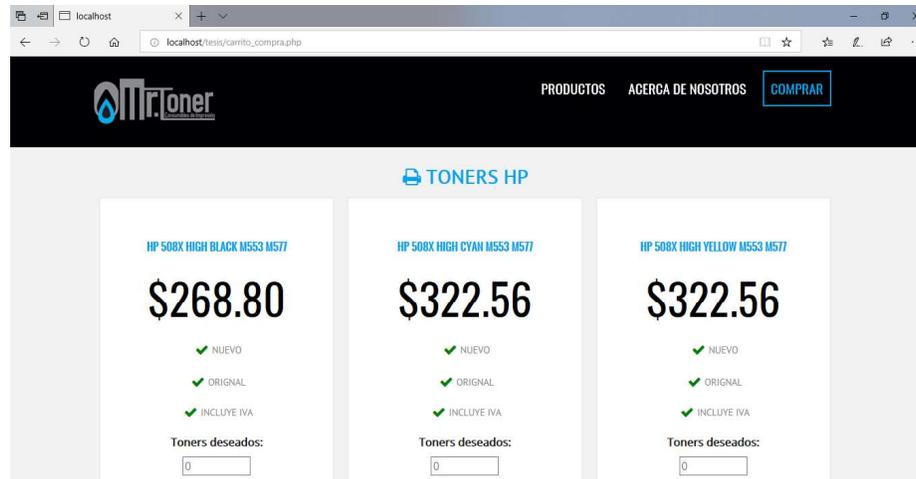


Figura 3. 9 Sistema web en el navegador Edge.

3.2 ENCUESTAS REALIZADAS

Las encuestas realizadas se encuentran disponibles en el ANEXO II y se las realizó a siete diferentes empresas que ofrecen los mismos tipos de servicios que la empresa Mr. Toner. Dichas encuestas tienen 15 preguntas enfocadas en cuatro aspectos del sistema web:

- Funcionalidad.
- Aceptación.
- Utilidad.
- Satisfacción.

3.3 TABULACIÓN DE LOS RESULTADOS

El resumen de los resultados de la encuesta de validación ha sido tabulado a continuación:

1. **¿Cree usted necesario tener un sistema web que le permita automatizar el proceso de facturación?**

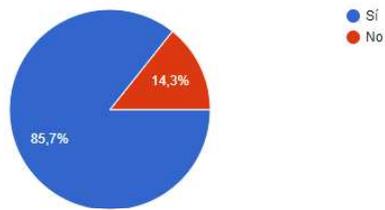


Figura 3. 10. Tabulación de la pregunta 1.

2. ¿Cree usted necesario tener un sistema web que le permita automatizar el proceso de inventario?

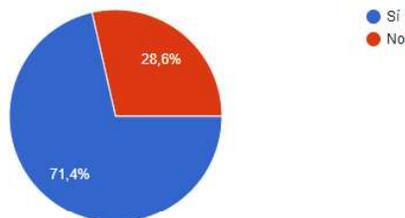


Figura 3. 11. Tabulación de la pregunta 2.

3. ¿Cree usted necesario tener un sistema web que le permita vender en línea?

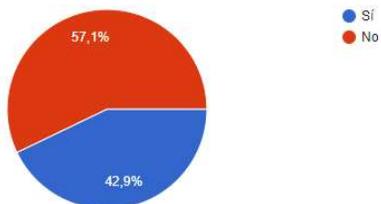


Figura 3. 12. Tabulación de la pregunta 3.

4. ¿Le pareció útil el sistema web presentado?

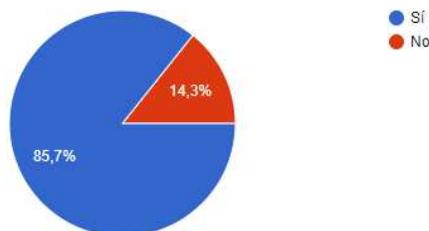


Figura 3. 13. Tabulación de la pregunta 4.

5. ¿Desearía un sistema web parecido para su empresa?

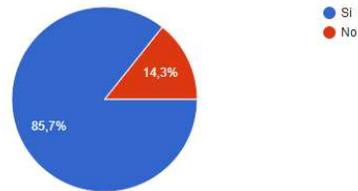


Figura 3. 14. Tabulación de la pregunta 5.

6. Desarrollar un sistema web que le permita vender en línea, facturar y mantener inventario es:

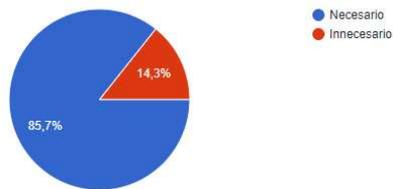


Figura 3. 15. Tabulación de la pregunta 6.

7. ¿Qué tan importante considera el uso del sistema web?

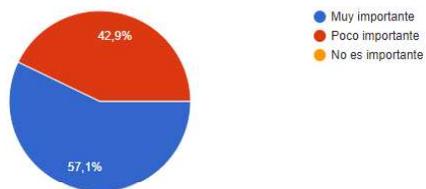


Figura 3. 16. Tabulación de la pregunta 7.

8. ¿Funcionó correctamente el sistema web?



Figura 3. 17. Tabulación de la pregunta 8.

9. Accesibilidad del sistema web



Figura 3. 18. Tabulación de la pregunta 9.

10. Información de datos recibida por el sistema web

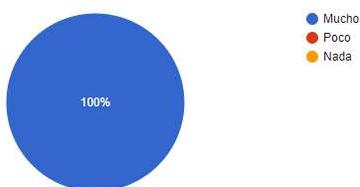


Figura 3. 19. Tabulación de la pregunta 10.

11. Diseño adecuado y comodidad de uso.



Figura 3. 20. Tabulación de la pregunta 11.

12. Rapidez con la que se resolvió la incidencia.

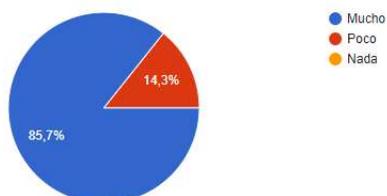


Figura 3. 21. Tabulación de la pregunta 12.

13. Su valoración general sobre el sistema web es:

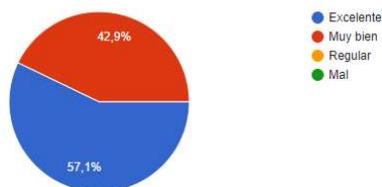


Figura 3. 22. Tabulación de la pregunta 13.

14. ¿Los datos aparecieron de forma intuitiva?



Figura 3. 23. Tabulación de la pregunta 14.

15. ¿La navegación del sistema web fue la óptima?



Figura 3. 24. Tabulación de la pregunta 15.

3.4 ANÁLISIS DE LOS RESULTADOS

Analizando los datos obtenidos en la sección 3.3 se puede concluir que el sistema web tiene aceptación, ya que la mayoría de las preguntas tuvo el resultado esperado.

3.5 CORRECCIÓN DE ERRORES

En los archivos *model-facturacion.php*, *model-mercaderia.php* y *model-salida.php* ocurrieron errores de compatibilidad del servidor web con las consultas a las bases de datos. Para la corrección de este tipo de errores es necesario utilizar la función `multi_query`

() como se muestra en el segmento de código 3.1, en donde se insertan y se actualizan varios campos de diferentes tablas, que permite realizar varias consultas de las bases de datos en el mismo archivo PHP.

```

    $sql .= "UPDATE ayuda_detalle SET ID_factura_detalle=LAST_INSERT_ID() WHERE ID_factura_detalle=1;";
    $sql .= "INSERT INTO detalle SELECT * FROM ayuda_detalle;";
    $sql .= "TRUNCATE TABLE ayuda_detalle;";
    $sql .= "UPDATE factura SET Subtotal=( SELECT SUM(Valor_total) AS suma_total FROM productos_comprados
        INNER JOIN detalle
        ON productos_comprados.ID_productos_comprados=detalle.ID_producto_detalle
        WHERE ID_factura_detalle = LAST_INSERT_ID()
        where ID_factura=LAST_INSERT_ID());";
    $sql .= "UPDATE factura SET IVA_12=(SELECT porcentaje/100 FROM iva)*(Subtotal-IVA_0) WHERE
    ID_factura=LAST_INSERT_ID();";
    $sql .= "UPDATE factura SET Total=Subtotal+IVA_12 WHERE ID_factura=LAST_INSERT_ID();";

    if ($conn->multi_query($sql)) {
        $resp = array(
            'respuesta'=>'exito',
            'id_administrador'=>12
        );
    }
    else {
        $resp = array(
            'respuesta'=>'error'
        );
    }

    mysqli_close($conn);
    die(json_encode($resp));

```

Segmento de Código 3.1. Utilización de la función multi_query ().

4. CONCLUSIONES Y RECOMENDACIONES

En este capítulo se detallan las conclusiones y recomendaciones después de haber finalizado el presente trabajo.

4.1 CONCLUSIONES

- Es importante automatizar el proceso de facturación, ya que se puede tener un registro de todas las cuentas por cobrar y un registro de la entrada de dinero de la empresa. Además, se puede realizar diferentes consultas a la base de datos para el uso adecuado de la información proveída por el usuario.
- De igual manera es importante automatizar el proceso de inventario, ya que este proceso cuantifica la cantidad de salida y entrada de la mercadería, teniendo una cantidad acertada cada vez que se quiera consultar. Además, se puede obtener el precio de entrada de la mercadería.
- La plataforma de Paypal facilita la forma de pago y da confiabilidad tanto al cliente como al vendedor. Paypal otorga al cliente una devolución en caso de ser necesario, dando así un respaldo al momento de realizar la compra en línea.
- MailChimp es una herramienta muy utilizada, permite enviar boletines a muchos usuarios de forma gratuita convirtiéndose así en una herramienta favorita para los programadores. Permitted que el sistema web obtenga una base de datos de correos electrónicos para de esta manera enviar información sobre promociones, noticias, etc. sobre la empresa.
- Google Maps API permite al usuario identificar el lugar exacto de la empresa, proveyendo al cliente información necesaria para realizar una compra, la API fue fácil de implementar y permite que la empresa mantenga su dirección al día y además provee a los clientes ir a la empresa Mr. Tóner sin mayor dificultad.
- Admin Lte permite diseñar diferentes aplicaciones web, adaptándose a cada uno de ellos. Tiene muchas herramientas visuales que le permite a Admin Lte ser uno de los diseños gratuitos más completos. Permitted que el sistema web tenga un

diseño web intuitivo y fácil de usar, mejorando la experiencia del usuario al momento de usarlo.

- La metodología WebML permite construir un sistema web de manera eficiente, debido a que sus fases son fáciles de entender y permiten la flexibilidad para adaptar la mayoría de los sistemas web. La metodología fue aplicada con éxito en el sistema web y proveyó mayor entendimiento de los requerimientos de la empresa Mr. Tóner ayudando así a la elaboración del sistema web.
- Obtener los datos para cumplir todas las necesidades que exige un usuario es la parte esencial del sistema web, debido a que de esta fase se elaboran las demás fases de la metodología.

4.2 RECOMENDACIONES

- Utilizar siempre la función `mysqli_multi_query ()` para la realización de varias consultas a la base de datos en un mismo archivo PHP, debido a que algunos servidores web no permiten el uso de varias consultas sin esta función.
- Utilizar debidamente las clases y los atributos del SDK de Paypal. Así mismo se puede implementar el costo de envío, impuestos, etc.
- El diseño web es importante, y se debe implementar con el correcto uso de las diferentes tecnologías como CSS3.
- Realizar el correcto diseño de base de datos, para que las consultas se las realicen de forma ágil y sencilla.
- Elegir una buena metodología que se adapte correctamente al sistema web.
- Un buen editor de texto facilita enormemente la codificación del sistema web.
- Realizar preguntas coherentes y concisas para la correcta obtención de los requerimientos.

- Utilizar las mismas versiones de los servidores (locales y en línea) tanto web como de base de datos, debido a que pueden ocurrir errores de incompatibilidad de código.
- Tener en cuenta que la seguridad de la información es importante ya que se maneja dinero en las transacciones. Usar de forma adecuada las clases del SDK de PayPal es una forma de dar seguridad al sistema web.

5. BIBLIOGRAFÍA

- [1] A. G. Khan, «Electronic Commerce: A Study on Benefits and Challenges in,» *Global Journal of Management and Business Research: B*, vol. 16, nº 1, 2016.
- [2] K. R. A. P. Frank Pilhofer, *Distributed Application Architecture*, Morgan Kaufmann, 2011.
- [3] Mailchimp, «Mailchimp.com,» 2001-2018. [En línea]. Available: <https://mailchimp.com/>.
- [4] I. PayPal Holdings, «UNITED STATES SECURITIES AND EXCHANGE COMMISSION,» 2015. [En línea]. Available: <https://www.sec.gov/Archives/edgar/data/1633917/000163391718000029/pyp1201710-k.htm>.
- [5] T. I. Society, *RFC2616*, UC Irvine: Compaq/W3C, 1999.
- [6] D. S. P. V. Natacha Güell, *Modeling Interactions and Navigation in Web Applications*, Rio de Janeiro: Springer Verlag New York, 2007.
- [7] G. R. Daniel Schwabe, *An Object Oriented Approach to Web-Based Application Design*, Rio de Janeiro: Prentice Hall Inc, 2011.
- [8] S. Ceri, P. Fraternali, A. Bongio, M. Brambilla y S. C. & M. Matera, *Designing Data-Intensive Web Applications*, USA: Morgan Kaufmann, 2002.
- [9] J. C. Salva, «Metodología WebML,» de *Aplicación práctica de la Metodología WebML de Desarrollo de Software*, Salta, 2016, pp. 12-14.
- [10] S. Luján Mora, *Programación en Internet: Clientes Web*, Alicante: Club Universitario, 2001.
- [11] dugwood, «clickheat,» 08 06 2011. [En línea]. Available: <https://www.dugwood.com/clickheat/index.html>. [Último acceso: 09 11 2018].
- [12] D. Flanagan y P. Ferguson, *avaScript: The Definitive Guide*, New York: O'Reilly Media, 2002.
- [13] J. J. Garrett, «Ajax: A New Approach to Web Applications,» *Adaptive Path*, 18 02 2005. [En línea]. Available: <http://adaptivepath.org/ideas/ajax-new-approach-web-applications/>. [Último acceso: 09 11 2018].
- [14] json, «json.org/about,» [En línea]. Available: <https://www.json.org/>. [Último acceso: 09 11 2018].

- [15] w3c, «HTML 5.2 W3C Recommendation,» 14 12 2017. [En línea]. Available: <https://www.w3.org/TR/2017/REC-html52-20171214/introduction.html#introduction>. [Último acceso: 10 11 2018].
- [16] w3c, «Introduction to CSS3,» 23 04 2001. [En línea]. Available: <https://www.w3.org/TR/2001/WD-css3-roadmap-20010523/>. [Último acceso: 10 11 2018].
- [17] W3C, «www.w3.org/DOM/,» 06 01 2009. [En línea]. Available: <https://www.w3.org/DOM/>. [Último acceso: 07 01 2019].
- [18] The jQuery Foundation, «What is jQuery?,» 20 05 2018. [En línea]. Available: <https://jquery.com/>. [Último acceso: 10 11 2018].
- [19] PHP Group, «Documentation,» PHP Group, 08 05 2018. [En línea]. Available: <https://secure.php.net/manual/es/intro-what-is.php>. [Último acceso: 12 11 2018].
- [20] Q-Success, «programming_language,» W3Tech, 07 09 2018. [En línea]. Available: https://w3techs.com/technologies/history_overview/programming_language. [Último acceso: 12 11 2018].
- [21] solid IT gmbh, «DB-Engines Ranking,» 2018. [En línea]. Available: <https://db-engines.com/en/ranking>. [Último acceso: 12 11 2018].
- [22] L. WELLING y L. THOMSON, PROGRAMACION DESARROLLO WEB CON PHP Y MYSQL, New York: ANAYA MULTIMEDIA, 2009.
- [23] D. Cochran, Twitter Bootstrap Web Development, Packt Publishing, 2012.
- [24] inc, «mailchimp-company-of-the-year-2017,» 2017, [En línea]. Available: <https://www.inc.com/magazine/201802/mailchimp-company-of-the-year-2017.html>. [Último acceso: 12 11 2018].
- [25] forbes, «PayPal Extends Payment Services To Nigeria, 9 Other Countries,» 17 06 2014. [En línea]. Available: <https://www.forbes.com/sites/mfonobongnsehe/2014/06/17/paypal-extends-payment-services-to-nigeria-9-other-countries/#27be6241649b>. [Último acceso: 12 11 2018].
- [26] muycomputerpro, «noticias,» MCpro, 7 11 2016. [En línea]. Available: <https://www.muycomputerpro.com/2016/09/07/paypal-mastercard-tiendas-fisicas>. [Último acceso: 12 11 2018].

- [27] PayPal, «¿Cómo se acredita un reembolso de PayPal?,» 2019. [En línea]. Available: <https://www.paypal.com/ec/smarthelp/article/%C2%BFc%C3%B3mo-se-acredita-un-reembolso-de-paypal-faq1083>. [Último acceso: 03 04 2019].
- [28] mike42, «PHP receipt printer library for use with ESC/POS-compatible thermal and impact printers.,» [En línea]. Available: <https://github.com/mike42/escpos-php>. [Último acceso: 09 04 2019].
- [29] A. Linux, «AMI de Amazon Linux,» 2019. [En línea]. Available: <https://aws.amazon.com/es/amazon-linux-ami/>. [Último acceso: 05 01 2019].
- [30] Amazon, «Tutorial: Instalación de un servidor web LAMP con la Amazon Linux AMI,» Amazon, 2019. [En línea]. Available: https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/install-LAMP.html. [Último acceso: 09 04 2019].
- [31] «Guía Digital,» [En línea]. Available: <https://www.guiadigital.gob.cl/articulo/pruebas-de-interfaces-y-contenidos.html>. [Último acceso: 19 05 2019].
- [32] PHPConf.Asia, «html_entity_decode,» [En línea]. Available: <https://www.php.net/manual/es/function.html-entity-decode.php>. [Último acceso: 10 04 2019].

6. ANEXOS

ANEXO I. Entrevista a la empresa mr. Toner.

ANEXO II. Encuesta de satisfacción.

ANEXO III. Código fuente.

ANEXO IV. Manual de usuario.

Los anexos se han incluido en un CD adjunto a este documento.

ORDEN DE EMPASTADO