



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN

**DESARROLLO DE UN SISTEMA WEB DE CONTROL
DOCUMENTAL
(APLICADO A EMPRESA AUTOMOTRIZ)**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN INFORMÁTICA**

PRESENTA:

ALEJANDRO HERNÁNDEZ AGUILAR

ASESOR: L.C. CARLOS PINEDA MUÑOZ

CUAUTITLÁN IZCALLI, EDO. DE MÉX.

2018



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JEFA DEL DEPARTAMENTO



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
UNIDAD DE ADMINISTRACIÓN ESCOLAR
DEPARTAMENTO DE EXÁMENES PROFESIONALES

M. en C. JORGE ALFREDO CUÉLLAR ORDAZ
DIRECTOR DE LA FES CUAUTITLÁN
PRESENTE

ATN: I.A. LAURA MARGARITA CORTAZAR ECHEGUEROA
Jefa del Departamento de Exámenes Profesionales

de la FES Cuautitlán.
DEPARTAMENTO DE

EXÁMENES PROFESIONALES

U. N. A. M.
ASUNTO: VOTO APROBATORIO
FACULTAD DE ESTUDIOS
SUPERIORES-CUAUTITLÁN



Con base en el Reglamento General de Exámenes, y la Dirección de la Facultad, nos permitimos comunicar a usted que revisamos el: Trabajo de Tesis

DESARROLLO DE UN SISTEMA WEB DE CONTROL DOCUMENTAL
(APLICADO A EMPRESA AUTOMOTRIZ)

Que presenta el pasante: ALEJANDRO HERNÁNDEZ AGUILAR

Con número de cuenta: 41311164-1 para obtener el Título de la carrera: Licenciatura en Informática

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

ATENTAMENTE

“POR MI RAZA HABLARÁ EL ESPÍRITU”

Cuautitlán Izcalli, Méx. a 06 de noviembre de 2017.

PROFESORES QUE INTEGRAN EL JURADO

	NOMBRE	FIRMA
PRESIDENTE	L.C. Carlos Pineda Muñoz	
VOCAL	MCC. María Araceli Nivón Zaghi	
SECRETARIO	L.I. Oscar Vilchis Guerra	
1er. SUPLENTE	L.I. Rosalba Nancy Rosas Fonseca	
2do. SUPLENTE	L.I. María Guadalupe Vázquez Salazar	

NOTA: los sinodales suplentes están obligados a presentarse el día y hora del Examen Profesional (art. 127).

LMCF/ntm*

AGRADECIMIENTOS

A Dios:

Por permitirme llegar al lugar en el que me encuentro, siendo uno de los pilares más importantes en mi vida, el cual me permitió culminar con una de las etapas más importantes para mí.

A mi mamá:

Gracias a ti puedo decir que estoy en donde estoy, un párrafo y/o unas líneas no describen la cantidad de cosas que pienso y siento sobre ti. Gracias por todo el apoyo que he recibido por parte tuya, sin dudar eres la mejor persona y madre que nadie jamás podrá tener, me siento realmente bendecido de tener una madre como tú y es que mi vida nunca será suficiente para agradecerte todo lo has hecho por mí.

Recuerda que este logro no solamente es mío, también es tuyo debido a la gran confianza que en mí depositaste y el amor incondicional que va más allá de todas las cosas. Siendo el pilar más importante en vida en todo aspecto. Te amo.

A mi hermano:

Tu apoyo siempre ha sido importante para mí, sobre todo en esta etapa en la cual quiero ser un ejemplo para ti. De verdad, muchas gracias por siempre estar ahí en toda situación, ya sea buena o mala.

A mi asesor Carlos Pineda Muñoz:

Sin lugar a dudas usted ha sido una de las personas más importantes en mi vida académica, gracias a usted y a sus enseñanzas estoy cumpliendo una de las metas más importantes en mi vida. Muchísimas gracias.

A Rocío Lara:

Tu apoyo incondicional me ha sido de gran ayuda, no tengo palabras para demostrar la gratitud que te mereces.

A Iriz Yutzil:

Para mí eres más que una amiga, gracias por todo el apoyo que me has brindado, me siento realmente bendecido de tenerte en mi vida, te aprecio demasiado.

A mis demás familiares y amigos:

Gracias por siempre estar ahí en todo momento, en lo bueno y en lo adverso, sin ustedes y su apoyo no sé qué sería de mí, aprecio cada enseñanza, consejo, etc., que he recibido de ustedes, porque saben que el apoyo es recíproco.

Cualquier omisión no es falta de gratitud.

ÍNDICE

Introducción.....	1
Objetivo.....	3
Objetivos específicos.....	3
Antecedentes.....	4
Justificación.....	5
Metodología.....	6
CAPÍTULO 1. Software Libre.....	7
1.1. ¿Qué es el Software Libre?.....	7
1.2. Licencias.....	9
1.2.1. Copyleft.....	10
1.3. Software Libre en proyectos.....	10
1.4. Infraestructura LAMP.....	12
1.4.1. (L) Linux. Sistema Operativo.....	12
1.4.2. (A) Apache. Servidor Web.....	13
1.4.3. (M) MySQL/MariaDB. Gestor de Base de Datos.....	13
1.4.4. (P) PHP. Lenguaje de programación orientado a Web.....	14
CAPÍTULO 2. Análisis y Diseño de Sistemas.....	16
2.1. Ingeniería del Software.....	16
2.1.2. Modelos del ciclo de vida tradicionales.....	17
2.2. Nuevos modelos de desarrollo (Tecnologías Ágiles).....	20
2.3. Metodología Tradicional VS. Metodología Ágil.....	21
2.4. UX - Experiencia de Usuario.....	23
CAPÍTULO 3. Modelo de desarrollo MVC (Modelo Vista Controlador).....	24
3.1. Proceso del modelo de desarrollo MVC.....	26
3.2. Sistemas anteriores a MVC.....	28
3.3. Evolución de los sistemas.....	29
3.4. Modelo de desarrollo MVC en PHP.....	31
CAPÍTULO 4. Ofimática y Facturación Electrónica.....	34
4.1. Ofimática.....	34
4.2. Aplicaciones Informáticas & Ofimáticas.....	35
4.2.1. Gestión Contable y Finanzas.....	35
4.2.2. Soporte y Formación al Usuario.....	38
4.2.2.1. Suites Ofimáticas.....	38
4.2.2.2. Multimedia y Presentaciones.....	39
4.2.2.3. Seguridad.....	40
4.2.3. Sistemas ERP.....	40
4.3. Facturación Electrónica.....	43
4.3.1. Firma Electrónica y Firma Electrónica Avanzada.....	44
4.3.2. Facturación Electrónica en ERP.....	45
CAPÍTULO 5. Caso práctico.....	46
5.1. Planteamiento del problema.....	46
5.2. Análisis preliminar.....	47
5.2.1. BPMN (Business Process Modeling Notation).....	49
5.2.2. Proceso general (Bonita BPMN).....	50

5.3. Análisis y diseño del sistema.....	53
5.3.1. Estructura interna de la aplicación.....	54
5.3.1.1. Diagrama Entidad – Relación.....	54
5.3.1.2. Diccionario de Datos	55
5.3.1.3. Diagramas.....	58
5.3.2 Sistema Web de Control Documental	64
5.3.3. Implementación.....	83
Conclusiones.....	84
Referencias	86

ÍNDICE DE FIGURAS

Figura 1. Estructura del Modelo en Cascada.....	18
Figura 2. Representación del Modelo Espiral.	19
Figura 3. Representación del Modelo Incremental.....	19
Figura 4. Representación general del modelo de desarrollo MVC.	26
Figura 5. Esquema general MVC - PHP.....	31
Figura 6. Estructura de archivos manejada por MVC.	32
Figura 7. Proceso general.	51
Figura 8. Notación del diagrama general.	52
Figura 9. Diagrama Entidad – Relación.	54
Figura 10. Proceso General del uso del sistema Pte. 1.....	58
Figura 11. Proceso General del uso del sistema Pte. 2.	59
Figura 12. Proceso de validación de documentos.....	60
Figura 13. Proceso ABC Usuarios.....	61
Figura 14. Proceso ABC Proveedores.	62
Figura 15. Proceso en el sistema como Proveedor.	63
Figura 16. Pantalla inicial del sistema.....	64
Figura 17. Inicio de sesión para el personal de la Empresa Automotriz S.A. de C.V.	65
Figura 18. Inicio de sesión para los proveedores.	65
Figura 19. Pantalla de ‘Bienvenida’ una vez ingresado al sistema.....	66
Figura 20. Interfaz donde se aprecia el manual del administrador.	67
Figura 21. Interfaz para el registro de nuevos usuarios.....	68
Figura 22. Pantalla donde se visualizan los usuarios registrados en el sistema.	69
Figura 23. Formulario para la edición de algún usuario seleccionado.	69
Figura 24. Formulario para el registro de un nuevo proveedor.	70
Figura 25. Pantalla donde se visualizan los proveedores registrados en el sistema.	71
Figura 26. Formulario para la edición de algún proveedor seleccionado.....	71
Figura 27. Selección de proveedor para consultar sus documentos.	72
Figura 28. Documentos a revisar.	73
Figura 29. Documentos validados.	74
Figura 30. Documentos invalidados.	75
Figura 31. Pantalla de inicio para usuarios de Contabilidad y Cuentas por Pagar.	76
Figura 32. Pantalla de inicio del proveedor.	77
Figura 33. Interfaz de la sección de ‘Contacto’.....	77
Figura 34. Interfaz para el envío de documentos.....	78
Figura 35. Lista de los documentos enviados por el proveedor.....	79
Figura 36. Lista de los documentos validados.....	80
Figura 37. Lista de los documentos invalidados.....	81
Figura 38. Acceso Denegado.....	82

ÍNDICE DE TABLAS

Tabla 1. Las cuatro libertades esenciales.....	8
Tabla 2. Diferencias entre Metodologías ágiles vs. Metodologías tradicionales.....	22
Tabla 3. Diccionario de datos: ‘Usuarios’.....	55
Tabla 4. Diccionario de datos: ‘Proveedores’.....	56
Tabla 5. Diccionario de datos: ‘Archivos’.....	57

Introducción

El presente trabajo se fundamenta en la necesidad expresada por una empresa con giro automotriz y por razones de confidencialidad no se hará uso de su nombre, por lo que la denominaré: “Empresa Automotriz S.A. de C.V.”.

Se presenta el desarrollo de una aplicación web que funciona con Base de Datos para la gestión de documentos fiscales, específicamente facturas en formato PDF y XML. La aplicación se realizará utilizando un conjunto de sistemas, tecnologías y/o lenguajes que son útiles de manera conjunta para el desarrollo de aplicaciones web, siendo éstas de tecnología libre (Software Libre), en concreto se hace referencia a la tecnología LAMP (Linux, Apache, MySQL/MariaDB, PHP).

Para esta aplicación se debe tomar en cuenta el análisis y diseño de sistemas, con el cual se deduce que para cumplir los objetivos para el desarrollo de la aplicación se necesita un patrón de arquitectura de software especial, éste debe ser capaz de prolongar el ciclo de vida de la aplicación, por esta razón es necesaria la utilización del modelo de desarrollo MVC (Modelo-Vista-Controlador), el cual ayuda a que la estructura de la aplicación siempre se encuentre ordenada, facilitando el manejo y uso de información que ésta controla.

Con la aplicación se pretenden utilizar los principios de la Ofimática para eliminar el papel que se utiliza cada vez que las facturas entran a la organización receptora, es decir, digitalizar todo el proceso, solicitando que los documentos deben ser enviados estrictamente en formato digital, respetando las extensiones de archivo establecidas para las revisiones (.PDF y .XML).

Se ha visto como a lo largo de los años y en el transcurrir de los avances tecnológicos la necesidad de tener implementado un sistema ha crecido, ya sea sistema de escritorio o sistema web y es que se han vuelto tan imprescindibles, no solo por el hecho de querer estar a la vanguardia en cuanto a tecnología para la optimización de los procesos que la organización maneja, sino que ayuda de manera sustancial en la misma.

Aunque no tanto se trata de cuestiones tecnológicas o de competencia sobre cual empresa u organización es la que se encuentra más a la vanguardia, se trata de cuestiones administrativas

y legales, y es que se ha visto como organizaciones encargadas de llevar un seguimiento de la documentación fiscal de varias organizaciones (ejemplo el SAT), han actualizado sus normas sobre el manejo de la documentación; estos documentos ya se han estado manejando en formatos digitales desde hace algún tiempo atrás.

Sin embargo, actualmente las empresas optan por requerir esa información de manera digital, desafortunadamente algunas no han tenido la oportunidad de implementar sistemas que les ayuden a llevar un mejor control sobre esta documentación fiscal y así tener un avance tecnológico que los ayude en la simplificación de sus procesos, ya que hay veces en las que revisar documentación de este tipo conlleva bastante tiempo. Un factor importante son las horas hombre que se reducen cuando un sistema de control documental es implementado, haciendo que las revisiones de documentos se agilicen.

Objetivo

Desarrollar una aplicación web que sea capaz de gestionar y administrar la carga de documentación fiscal de proveedores (facturas: PDF, XML), para el control y seguimiento de los mismos, utilizando Software Libre, así como también la metodología de programación orientada a objetos en conjunto con el modelo de desarrollo MVC (Modelo-Vista-Controlador).

Objetivos específicos

1. Reestructurar los procesos de la gestión de documentos fiscales (facturas) de los proveedores;
2. Utilizar de Software Libre; y
3. Utilizar el modelo de desarrollo MVC.

Antecedentes

El uso de las tecnologías de la información obliga a cada una de las personas a que incurriere en el uso de éstas, sin importar el contexto en el que se aplique. Por esta razón, las empresas y/o instituciones se ven en la necesidad de automatizar sus procesos. Uno de los objetivos generales de la automatización de procesos es mejorar la productividad de la organización, reduciendo tiempos y mejorando la calidad de las mismas. De la misma manera para mejorar las condiciones de trabajo del personal.

Además, en México, el SAT juega un papel importante y estricto en cuanto a documentación fiscal se trata ya que desde un tiempo atrás a la fecha todos y cada uno de los contribuyentes ya sea persona física o moral están obligados a emitir facturación electrónica, con la finalidad de que los mismos tengan un control en el sistema de Registro Fiscal para que de esta manera el contribuyente cumpla con sus obligaciones fiscales.

Anteriormente cuando de facturas se hablaba no se pensaba otra cosa que no fuera en una hoja de papel con las firmas autógrafas correspondientes que la hacían válida ante cualquier entidad. Sin embargo, surge la facturación electrónica con la cual se logró la eliminación del papel que se utilizaba en la facturación tradicional, lo cual significó un gran avance, además de que el Servicio de Administración Tributaria logró mantener un control más estricto sobre todos los contribuyentes.

El realizar todo este tipo de movimientos algunos años atrás parecería imposible, ya que entran cuestiones de seguridad, confiabilidad, etc., situaciones que actualmente son muy bien controladas por este tipo de entidades (SAT) para el resguardo y manejo de la información, etc., pero gracias al avance tecnológico que se ha logrado, la población que hace uso de medios electrónicos para realizar cualquier tipo de movimiento se siente más confiada.

Justificación

La contribución del Lic. en Informática se ha vuelto trascendental cuando se trata de solucionar necesidades del tipo tecnológicas en cuanto al desarrollo de sistemas, haciendo hincapié en la visión con la que éste debe de contar para brindar el resultado deseado.

Actualmente, considerando el entorno nacional se deduce que el Sistema Tributario ha cambiado muchas de sus reglas de trabajo y ahora todo debe ser trabajado con facturas y sellos electrónicos, por esta razón las empresas se ven en la necesidad de estar a la vanguardia en cuanto al manejo de sus procesos que les ayude a manejar toda esa información.

Producto de la necesidad descrita anteriormente se justifica el desarrollo del sistema web para que la organización logre tener un estatus digital que la ayude a mejorar. Llevando un control más preciso sobre la documentación fiscal.

Se utilizará Software Libre para llevar a cabo este proyecto, ya que se estableció con la Empresa Automotriz S.A. de C.V. que no se necesitaría de ningún tipo de inversión que ayude al desarrollo de este sistema, además en el Software Libre se tienen un sin fin de herramientas las cuales ayudan a cumplir el objetivo especificado.

Metodología

Se inicia con el análisis para llevar a cabo el rediseño de los procesos manejados por la empresa, posteriormente se elige la metodología de desarrollo más adecuada para lograr la optimización de los mismos, para modelar los procesos se propone BPMN¹ (Business Process Model and Notation / Modelo y Notación de Procesos de Negocio) para la definición de los procesos más allá de la forma convencional mostrando modelos de negocios fáciles de entender.

Se propone utilizar el modelo de desarrollo MVC (Modelo-Vista-Controlador) para crear la estructura de la aplicación y tener un mejor control sobre ésta, así mismo se logra que el tiempo de vida de la aplicación se extienda, de la misma manera se requiere la utilización de la POO (Programación Orientada a Objetos) para la estructura lógica en conjunto con el lenguaje PHP de programación y el gestor de base de datos MariaDB para el almacenamiento de la información.

¹ <https://www.bonitasoft.com/>

CAPÍTULO 1. Software Libre

"Las únicas personas que tienen algo que temer de software libre son aquellos cuyos productos tienen un valor aún menor."

– David Emery

Actualmente se ha visto como grandes corporaciones tratan de monopolizar todo lo que les sea posible, este acto de querer tener el control incluye varios servicios que son utilizados diariamente; el Software no es la excepción y como se ha visto a lo largo de los años, las empresas que se dedican a la creación y distribución del software de pago lo llegan a vender en precios exorbitantes. Ante esta situación siempre ha existido y existirá una alternativa que va creciendo día con día, el Software Libre.

1.1. ¿Qué es el Software Libre?

Es importante conocer que el Software Libre no es otra cosa más que la libertad de utilizar un código o programas (en esencia un producto) según las necesidades, éste puede ser modificado con la seguridad de que no se incurre en acciones ilegales, ya sea para fines personales, laborales, académicos, etc., o simplemente por contribuir en la mejora de los programas y/o paquetes que están disponibles para toda la comunidad que hace uso de éstos, de la misma manera éste puede ser redistribuido para que alguien más si así lo desea haga sus propias modificaciones, versiones, etc., y así sucesivamente; podría ser interpretado como un círculo vicioso para el mejoramiento del Software Libre.

Analizando desde otra perspectiva, según la *Free Software Foundation*¹ dice que el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software.

¹ Organización no lucrativa dedicada a promover y defender el uso y desarrollo de la documentación y el software libre.

Para entender mejor el párrafo anterior se intuye que el software libre es denominado “libre” si éste garantiza ciertas libertades. Citando una parte de la filosofía de *Richard Stallman*² acerca del software libre en la cual menciona estas libertades³:

Libertad	Descripción
0	La libertad para ejecutar el programa sea cual sea nuestro propósito.
1	La libertad para estudiar el funcionamiento del programa y adaptarlo a tus necesidades –el acceso al código fuente es condición indispensable para esto.
2	La libertad para redistribuir copias y ayudar así a tu vecino.
3	La libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad –el acceso al código fuente es condición indispensable para esto.

Tabla 1. Las cuatro libertades esenciales.

Entonces, es software libre aquel que cumple estas cuatro libertades, si alguna de éstas no se cumple, el software deja de ser libre. No quiere decir que éste ‘no sea comercial’; como menciona *Richard Stallman* en su libro “*Software libre para una sociedad libre*”: –Cualquier programa libre estará disponible para su uso, desarrollo y distribución comercial⁴–.

Sin embargo, el software libre siendo una alternativa a todas aquellas plataformas, programas, aplicaciones, etc., de pago, es capaz de brindar un sin fin de soluciones que las proporcionadas por el software privativo; éste último no brinda la libertad de acceder al código fuente, por lo tanto no se le pueden hacer mejoras y si se trataran de hacer se corre el riesgo de tener problemas legales debido a su manejo estricto de licencias que incluso éstas llegan a ser personales, por ende, no se pueden hacer copias ni distribuciones del software.

² Presidente de la *Free Software Foundation*, además creador del proyecto GNU y fundador del movimiento del Software Libre.

³ Stallman, R. (2004). *Software libre para una sociedad libre*. pp 59, 60.

⁴ Stallman, R. (2004). *Software libre para una sociedad libre*. pp 46.

Se tiene en mente que al pagar un software se está adquiriendo con seguridad incluida, lo que implica que el usuario se confíe y lo utilice deliberadamente. Por esta razón día con día se han ido creando y mejorando alternativas de software para que los usuarios se ‘liberen’⁵ de estas situaciones que pueden llegar a ser perjudiciales en un futuro.

Se ha visto un impacto notorio en cuanto al uso del software libre, ya que además de brindar alternativas al software privativo, también es utilizado por un sin fin de organizaciones (incluso gubernamentales) que hacen uso de estos sistemas operativos libres que están a entera disposición, ya que brindan un alto grado de confiabilidad; un ejemplo claro es la NASA, que hace uso de estos sistemas en sus robots, satélites, supercomputadoras, etc., ya que necesitan un gran poder de procesamiento.

1.2. Licencias

Al hacer uso de cualquier distribución de sistema operativo, programa o aplicación con software libre también se hace uso de una licencia o licencias que regulan a éstos, ya que el que sean libres no quiere decir que no dependan de alguna licencia que los regule, por lo tanto, las licencias libres especifican a los usuarios qué es lo que se está adquiriendo, y la libertad que se tiene para usar el producto. Pueden ser encontradas varias categorías de licencias que se utilizan para regular el software libre, pero la gran mayoría por su relevancia son clasificadas dentro de los siguientes tipos:

1. BSD (Berkeley Software Distribution)⁶;
2. MPL (Mozilla Public License)⁷; y
3. GPL (General Public License)⁸.

⁵ Libertad de los usuarios, que sean ellos los que tengan el control absoluto sobre el Software que utilizan.

⁶ Llamadas así las distribuciones de código fuente hechas en la Universidad de Berkeley, California.

⁷ Desarrollada por la *Fundación Mozilla*.

⁸ Utilizada para la mayoría de los programas GNU, y para más de la mitad de los paquetes de Software Libre.

De las tres, la más popular es la licencia GPL perteneciente al proyecto GNU, el cual a su vez está basado bajo el concepto de ‘Copyleft’ el cual establece los criterios de distribución del software libre mientras las libertades citadas anteriormente se cumplan.

1.2.1. Copyleft

El Copyleft⁹ es una licencia o regla contraria al Copyright, ya que no exige que el software debe contener derechos reservados, cuando se utiliza éste se establece que las libertades centrales deben de cumplirse, además el proyecto GNU utiliza esta licencia para proteger legalmente el uso de estas libertades para todos, ya que exige que no se deben de agregar nuevas restricciones.

1.3. Software Libre en proyectos

A partir de las grandes ventajas que tiene el software libre sobre las plataformas de paga también proporciona herramientas que son de gran utilidad al momento de emprender algún proyecto, ya que no encadena al usuario a seguir un régimen de normas como los que establecen algunos programas que se utilizan para la realización de proyectos dentro del software privativo.

Actualmente, los proyectos (de todo tipo) han tomado cierta importancia sobre todo para las empresas que desean mejorar su estatus digital, ante esta situación se puede encontrar un sin fin de herramientas que son de gran utilidad al momento de iniciar un proyecto, ya que existe software que se encarga de llevar un control sobre el proyecto, que van desde la planeación del proyecto hasta el desarrollo del mismo.

⁹ Extraído de *Software libre para una sociedad libre*, el 10 de 07 de 2017, pp 125.

La cantidad de proyectos de software libre que circulan al rededor del mundo ha crecido bastante, y cuando el usuario hace uso de éstos se percata de las facilidades que proveen, ya que el apoyo por parte de todos los desarrolladores que contribuyen con estos proyectos se encuentran constantes para realizar nuevos aportes y mejoras al software. Por ende, estos proyectos se encuentran bajo licencia GNU/GPL por el hecho de las libertades que los desarrolladores y usuarios tienen.

Los proyectos de software libre más utilizados a nivel mundial son los siguientes:

- **Linux**¹⁰: Sistemas operativos; en esencia el núcleo de trabajo;
- **Apache**¹¹: Servidor web;
- **MySQL/MariaDB**¹²: Gestor de base de datos;
- **PHP**¹³: Lenguaje de programación orientado a web;
- **Sendmail**: Medio de transporte de correos electrónicos vía SMTP (Simple Mail Transport Protocol);
- **GNOME**¹⁴: Entorno de escritorio para Linux;
- **Firefox**¹⁵: Navegador web de código abierto; y
- **JavaScript**¹⁶: Lenguaje de programación interpretado.

Éstas son solo algunas de las herramientas realmente poderosas que han sido desarrolladas para el fomento del software libre, y así mismo coadyuvar a los proyectos que se realizan con éstas. Siendo amigables con el usuario que las utiliza.

¹⁰ <https://www.linux.org/>

¹¹ <https://www.apache.org/>

¹² <https://mariadb.org/>

¹³ <http://php.net/>

¹⁴ <https://www.gnome.org/>

¹⁵ <https://www.mozilla.org/es-MX/firefox/>

¹⁶ <https://www.javascript.com/>

1.4. Infraestructura LAMP

La infraestructura LAMP cuyo acrónimo se desglosa más adelante; ofrece herramientas sumamente poderosas, cuando se decide hacer un proyecto que implica el desarrollo de una aplicación web ésta puede ser la solución, y es que en conjunto brinda un sin fin de posibilidades para la culminación exitosa de dichos proyectos.

Esta arquitectura es de gran utilidad para la implementación de algún servidor web dentro de una organización ya que se evitan costos por tener que realizar alguna renta mensual/anual de servicio web, y es que se ha comprobado que la infraestructura LAMP tiene una velocidad que va más allá que la de sus competidores de paga, por el simple hecho de que carga los recursos necesarios para el proceso necesario, caso contrario con su competencia que carga todos los recursos a la espera de cual se necesita.

1.4.1. (L) Linux. Sistema Operativo

Linux además de ser el sistema operativo con el que se ejecutan aplicaciones alternas a las privativas, es el núcleo de todo el funcionamiento de la infraestructura LAMP. La elección de utilizar Linux es simple ya que ofrece velocidad y confiabilidad, además de que no existen preocupaciones por tener que pagar por dicho sistema, sin embargo, se pueden realizar donaciones. Actualmente hay demasiadas distribuciones de Linux disponibles para la comunidad que se adecuan a cada una de las necesidades que se tienen, además de la gran adaptación con el Hardware ya que los requerimientos para la instalación de un sistema operativo de este tipo son mínimos.

Linux empezó siendo un pasatiempo de Linus Torvalds cuando él se encontraba estudiando en la universidad de Helsinki, Finlandia. Linus empezó trabajando con un pequeño sistema de Unix llamado Minix; en 1991 comenzó el desarrollo de un sistema operativo similar a Minix que fuera más allá de sus estándares; publicando así la versión la primera versión del Kernel Linux 0.02. Desde ese entonces, Linux ha crecido considerablemente, formando una comunidad de desarrolladores que están dispuestos a contribuir en este proyecto que inicio hace ya más de dos décadas. Además de que Linux ha mejorado considerablemente desde que inició.

1.4.2. (A) Apache. Servidor Web

Apache es un servidor web HTTP de código abierto, es uno de los más utilizados a nivel mundial por la rapidez que éste ofrece. La integración con sistemas Linux es de las más estables que se conocen además, según el popular sitio Netcraft informa que a la fecha apache es utilizado en el 44.9% de los servidores web a nivel mundial, además de que sigue ganando popularidad y cada día son más los servidores que se actualizan o son creados con apache. La ventaja que tiene apache es que funciona tanto para páginas web estáticas como páginas web dinámicas.

Por otra parte, las vulnerabilidades que han sido encontradas en el servidor apache únicamente pueden llegar a afectar de manera local, es decir, que no pueden ser aprovechadas por usuarios malintencionados de manera remota, aunque se puede dar el caso, pero es algo que muy rara vez puede llegar a suceder.

1.4.3. (M) MySQL/MariaDB. Gestor de Base de Datos

MySQL es el gestor de base de datos relacionales más utilizado en el mundo por la gran robustez que éste ofrece, además de que hace una dupla increíble con el lenguaje PHP al momento de consultar información. Este gestor permite almacenar bases de datos con grandes cantidades de información al ser tan flexible en cuanto a los tipos de datos que puede almacenar, permitiendo así que el sitio o sistema web sea más dinámico en cuanto al manejo y uso de la información.

Desafortunadamente MySQL que actualmente es de Oracle¹⁷ pasó de ser un gestor de base de datos Open Source a ser ahora Open Core, es decir, que no todas las extensiones de MySQL son libres, se volvieron de “paga” por parte de Oracle.

¹⁷ <https://www.oracle.com/es/index.html>

Por otro lado, MariaDB es una derivación de MySQL completamente gratis que es de código abierto, ésta derivación hace posible que se hagan cambios entre gestores de bases de datos fácilmente, es decir, una migración de MySQL a MariaDB y viceversa puede llegar a ser una tarea fácil, además de que ambos gestores comparten características similares como la misma interfaz de usuario, consultas similares, etc.

1.4.4. (P) PHP. Lenguaje de programación orientado a Web

PHP es de los lenguajes orientados a la programación web más populares que trabaja del lado del servidor para generar páginas web dinámicas, ya que simplifica el cómo mostrar algún resultado esperado; con HTML pudiera ser logrado, pero se necesita de mucho código y a veces el resultado no es el deseado.

Lo interesante de este lenguaje es la adecuación que realiza con el gestor de base de datos MariaDB, es posible mostrar resultados obtenidos de una base de datos y éstos ser interpretados de manera dinámica, lo cual es de suma ayuda ya que el código php al ser incrustado en páginas HTML ocupa pocas líneas de código, haciendo que la aplicación en esencia sea más ligera. Una de las partes más importantes de este lenguaje es que, aunque se consulte el código fuente de la página, la sección o secciones con código PHP únicamente mostrarán el resultado dinámico obtenido, jamás se mostrará la estructura lógica del código que hace posible la obtención del resultado.

En otra perspectiva, el utilizar LAMP facilita la resolución de las necesidades que se lleguen a presentar, por el simple hecho de que toda la infraestructura es de código abierto y no tiene que haber preocupaciones en aspectos económicos, incluso de seguridad, ya que cada componente ofrece un nivel de seguridad que si se es configurado adecuadamente hace que ofrezca un grado de confiabilidad mayor.

El uso de LAMP en este trabajo profesional surgió por la confiabilidad que esta arquitectura representa, además de que las configuraciones al momento de instalar cada uno de los componentes es sencilla, y de existir algún error éste puede ser corregido fácilmente. Por otro lado, el utilizar LAMP incita a que el usuario tenga que hacer uso de otras herramientas de

código abierto anexas a esta infraestructura, por ejemplo, en la planeación de un proyecto se puede utilizar ‘Planner ®’¹⁸; en el modelado de base de datos se puede utilizar ‘MySQL Workbench’¹⁹ que es una herramienta que se puede sincronizar con la base de datos que se tiene instalada y configurada, etc. Siempre teniendo una opción libre para aquellas herramientas de pago que suelen ser costosas.

¹⁸ Herramienta para la planeación, programación, y seguimiento a proyectos para el escritorio GNOME.

¹⁹ Herramienta visual para el diseño de bases de datos.

CAPÍTULO 2. Análisis y Diseño de Sistemas

“Si no lo puedes explicar con simplicidad, es que no lo entiendes bien.”

– Albert Einstein

Al momento del desarrollo de un sistema es preciso analizar qué requerimientos se necesitan para la exitosa culminación de éste, al hablar de requerimientos no únicamente se habla de analizar las necesidades del sistema, se trata de sacar un análisis general en cuanto al lugar en donde este sistema se va a implementar y es necesario saber a qué usuarios está destinado el sistema, para así llegar a la metodología correcta para el desarrollo del mismo.

2.1. Ingeniería del Software

Citando a *Ian Sommerville* el cual define a la Ingeniería del Software como: –Una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza¹–.

En el desarrollo de un proyecto de Software es importante tomar en cuenta a la Ingeniería del Software, ya que se puede obtener la metodología adecuada para el desarrollo del mismo ya que comienza partiendo del análisis de requerimientos que determinan cuales necesidades son las que se tienen que satisfacer, de la misma manera, los tipos de usuarios que van a hacer uso del sistema.

Anteriormente al hacer uso de la ‘Ingeniería del Software’ se tenía que seguir un enfoque de una manera tradicional, es decir, la(s) metodología(s) que se ocupaban implicaban la utilización de bastante documentación la cual tenía que ser realizada en tiempo y forma para darle continuidad al desarrollo del software.

¹ Sommerville, I. (2005). *Ingeniería del software, Séptima edición*. pp 6.

Además, de todas las fases del desarrollo de un proyecto de software la del análisis de los requerimientos, planeación y control debían tener un seguimiento estricto en cuanto a la documentación y los procesos que se realizaban en cada una de estas fases, para que el proyecto terminara con éxito y sin retraso alguno.

Sin embargo, de cierta manera el tener que realizar bastante documentación implicaba que se pudiera retrasar el proyecto por la falta de la documentación correcta, es por eso que para evitar estas situaciones se tenían que hacer adecuaciones a las normas de trabajo, y así como se vigilan los procesos descritos anteriormente también se hacía con todo el proyecto para que el software a obtener fuera predecible.

Así como la tecnología avanza diariamente, sucede lo mismo con las metodologías de desarrollo las cuales se han ajustado a la época actual, simplificando la documentación que debe ser entregada en cada fase del proyecto.

2.1.2. Modelos del ciclo de vida tradicionales

Los modelos de ciclo de vida se encargan de definir o establecer las fases que son necesarias en la realización del desarrollo del software. A continuación, se aprecian algunos de los modelos más utilizados y/o conocidos en el desarrollo de un software.

Modelo en Cascada:

Es uno de los modelos más destacados que se conocen para el desarrollo de un software, actualmente se usa poco, aunque no quiere decir que esté en desuso, ya que de aplicarse correctamente se pueden obtener resultados favorables; este modelo está compuesto de las siguientes fases:

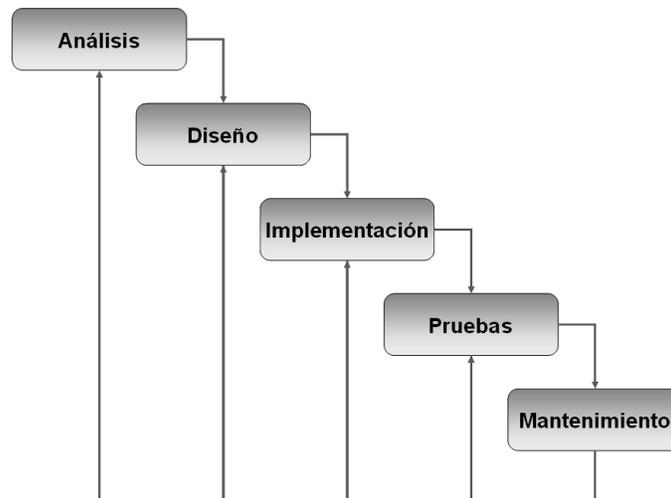


Figura 1. Estructura del Modelo en Cascada.

Por lo tanto, cada una de las fases debe ser realizada de manera lineal, es decir, no se puede empezar con la siguiente fase si la fase actual no ha sido concluida, por ende, al terminar una fase y comenzar con la siguiente no se puede regresar a la fase anterior. La controversia sucede al momento de querer regresar a una fase anterior, por ejemplo, al estar en la fase del desarrollo del sistema y si el cliente requiere modificaciones adicionales que impactan desde la fase del diseño del sistema se debería tomar la decisión de si es conveniente reiniciar el proyecto o terminarlo y evaluar el resultado final contra el resultado esperado.

Modelo en Espiral:

Este modelo es completamente diferente al descrito anteriormente, ya que centra sus fases en la obtención del análisis de riesgos y es que el recorrido que sigue atraviesa cada una de sus fases una y otra vez, las veces que sea necesario hasta lograr reducir los riesgos que se pueden presentar, además de que existir riesgos en la fase en la que se encuentre actualmente, no se podrá continuar con la siguiente fase hasta que los riesgos latentes hayan sido resueltos. El modelo en espiral presenta las siguientes fases:



Figura 2. Representación del Modelo Espiral.

Modelo Incremental:

El modelo incremental surge en base al *modelo en cascada* y al *modelo de prototipos*². En este modelo lo que destaca es la forma de realización de las fases del desarrollo, ya que lo hace de manera escalonada mientras el tiempo en base al calendario:

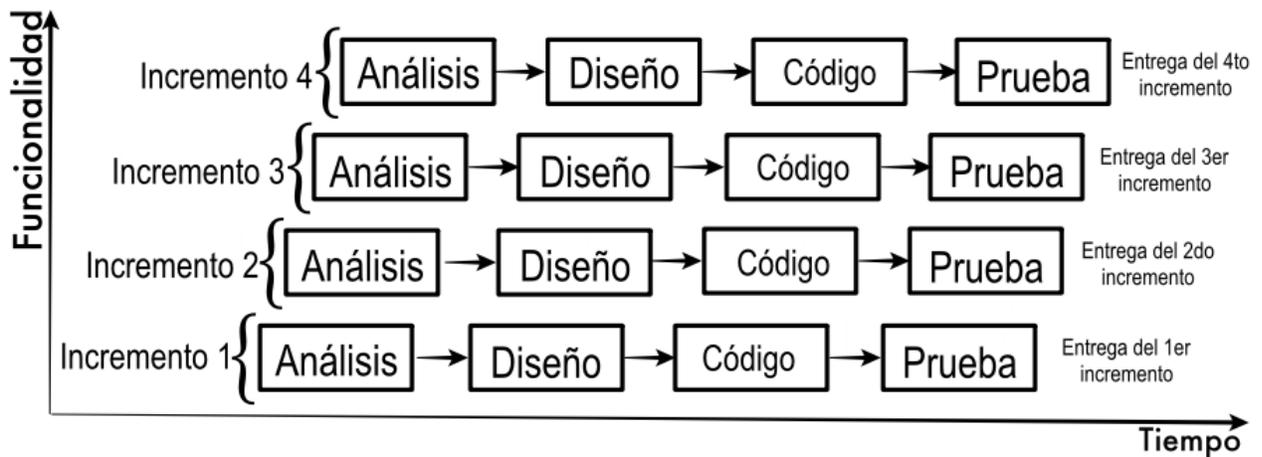


Figura 3. Representación del Modelo Incremental.

² Modelo basado a prueba y error hasta obtener el resultado que el usuario espera.

Cada fase que es concluida en este modelo produce un incremento en el software, la parte esencial en este modelo es lograr la construcción de la primera fase o secuencia ya que es considerada como el núcleo del software, además de que este modelo resulta ser de manera interactiva, ya que al final de cada incremento realizado el producto es operacional. Los primeros incrementos pueden ser realizados por un grupo reducido de personas, pero conforme se avanza en el proyecto es necesario añadir más, además de que este modelo en cada incremento se pueden prever errores técnicos.

2.2. Nuevos modelos de desarrollo (Tecnologías Ágiles)

Los modelos de desarrollo también se actualizan constantemente, aunque la mayoría de estos utilizan metodologías tradicionales que son combinados con técnicas modernas que resultan ser útiles al momento de ser empleados en el desarrollo de algún software.

Por lo tanto, los modelos de desarrollo ‘modernos’ que son los más utilizados actualmente son:

1. SCRUM³:

Ésta define un marco para la gestión de proyectos. Está destinada a los proyectos que cambian de requisitos constantemente, ya que unas de sus principales características son el desarrollo de software que es realizado en iteraciones (también denominadas sprints) con duración de 30 días. Al terminar cada iteración es un incremento utilizable que el cliente puede ver. Y la otra característica se basa en las reuniones a lo largo del desarrollo que se tienen diariamente con duración de 15 minutos para la coordinación de integración del proyecto.

Crystal Methodologies⁴:

Son un conjunto de metodologías para el desarrollo de software caracterizadas por estar enfocadas en los integrantes que forman el equipo y la reducción al máximo del número de artefactos producidos.

³ www.controlchaos.com

⁴ www.crystallmethodologies.org

DSDM – Dynamic Systems Development Method⁵:

Esta metodología define el marco para el desarrollo para la producción del software, sus características son el acoplamiento entre un proceso iterativo e incremental y el equipo de desarrolladores y el usuario final.

ASD – Adaptive Software Development⁶:

Éste es orientado a los componentes del software además de ser iterativo en las tareas y tolerante a los cambios. Éste es iterativo y más enfocado a los componentes del software que a las tareas, además de ser tolerante a los cambios.

2.3. Metodología Tradicional VS. Metodología Ágil

En la siguiente tabla se listan las diferencias que son más sobresalientes entre los tipos de metodología, ya sea tradicional o ágil. Cabe aclarar que no solo hace referencia únicamente al proceso, sino que engloba también al equipo de desarrollo, a la organización, etc.

⁵ www.dsdm.org

⁶ www.adaptivesd.com

<i>Metodología Ágil</i> ⁷	<i>Metodología Tradicional</i> ⁸
Pocos Artefactos. El modelado es prescindible, modelos desechables.	Más Artefactos. El modelado es esencial, mantenimiento de modelos.
Pocos Roles, más genéricos y flexibles.	Más Roles, más específicos.
No existe un contrato tradicional, debe ser bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Orientada a proyectos pequeños. Corta duración (o entregas frecuentes), equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio.	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas y/o usadas en proyectos grandes y con equipos posiblemente dispersos.
La arquitectura se va definiendo y mejorando a lo largo del proyecto.	Se promueve que la arquitectura se defina tempranamente en el proyecto.
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo.	Énfasis en la definición del proceso: roles, actividades y artefactos.
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Se esperan cambios durante el proyecto.	Se espera que no ocurran cambios de gran impacto durante el proyecto.

Tabla 2. Diferencias entre *Metodologías ágiles* vs. *Metodologías tradicionales*.

⁷ Se enfocan en desarrollar más a detalle el software dejando un poco de lado toda la documentación.

⁸ Predomina el enfoque disciplinar, incluyendo documentación excesiva; centradas en cumplir un plan de proyecto.

2.4. UX - Experiencia de Usuario

El término ‘UX’ (User eXperience) para hacer referencia a la *experiencia de usuario* suele ser poco conocido, aunque la realidad es que éste concepto es considerado como una de las partes esenciales en el proceso que acompaña al diseño web. Desde otro punto de vista, UX es descrito por *Dan Saffer*⁹ como – “Lo que el cliente percibe al usar o probar un producto y una forma de ver estos productos de manera integral desde el punto de vista de un usuario que probablemente no le importa cómo se hacen esos productos, sólo el producto en sí mismo”¹⁰ –.

Por otra parte, el ‘Diseño UX’ se centra en la resolución de necesidades específicas que son requeridas por los usuarios finales para así obtener el mejor resultado y a la vez adquiriendo una mejor experiencia de uso con el mínimo esfuerzo. Cada una de las fases por las que atraviesa el Diseño UX es fundamentada en base a la capacidad del usuario, necesidades, etc.

UX cuando es contemplado en el proceso del desarrollo de software tiene establecido la forma de trabajo, y lo generaliza en tres puntos:

1. Conocer a fondo a los usuarios finales, normalmente utilizando investigación cualitativa y/o cuantitativa;
2. Diseñar un producto que resuelva sus necesidades y se ajuste a sus capacidades expectativas y motivaciones; y
3. Poner a prueba lo diseñado, usando test de usuario.

Además de seguir con los puntos que establece el Diseño UX, se caracteriza por que no hay un seguimiento específico para la utilización de éste, no sigue un procedimiento específico, lo que se tiene que tener en cuenta es que se deben conocer herramientas y técnicas, así como saber en qué momento y cómo utilizarlas. El propósito principal del Diseño UX se centra en formar y diseñar la experiencia del usuario en todo momento, desde el inicio hasta el final.

⁹ Líder en diseño de productos, autor de cuatro libros relacionados con el diseño.

¹⁰ Extraído de *PMQuality*, el 20 de 07 de 2017 <https://pmqlinkedin.wordpress.com/about/que-es-el-diseno-ux/>

CAPÍTULO 3. Modelo de desarrollo MVC (Modelo Vista Controlador)

“Primero resuelve el problema. Entonces,
escribe el código.”

– John Johnson

Cuando se trata de desarrollar un sistema web surge la pregunta ¿cómo realizarlo? Ya que es necesario desarrollar un sistema que sea de fácil uso y entendimiento, de igual manera que su ciclo de vida sea más duradero y a su vez que el mantenimiento que se le aplique sea más sencillo, además de que permita la reutilización de código el cual puede ser implementado en los mantenimientos al sistema.

MVC es un modelo de desarrollo de software descrito por primera vez en el año de 1979 por *Trygve Reenskaug*¹ el cual puede ser utilizado cuando se trata del desarrollo e implementación de sistemas que requieren el uso de interfaces de usuario, el cual se caracteriza por mantener la estructura de la aplicación siempre ordenada, fundamentándose en dividir el código en tres capas diferentes:

1. *Model* (modelo);
2. *View* (vista); y
3. *Controller* (controlador).

Lo anterior, es separar el código de manera que los datos son autónomos de la aplicación, la interfaz de usuario por otro lado y en la tercera separación la lógica de control.

¹ Científico computacional y profesor de la Universidad de Oslo, Noruega.

Modelo:

En esta capa es donde se establecen todas las funciones que tienen que ver con las acciones referentes al manejo de los datos, los cuales generalmente son extraídos de las bases de datos; únicamente en esta capa se realizan consultas de todo tipo hacia el gestor de base de datos (ej. SELECT, INSERT, UPDATE, DROP, TRUNCATE, etc.), por ende, el modelo no conoce más allá de sí mismo, no tiene conocimiento sobre cómo están estructurados los controladores ni las vistas.

Vista:

En esta capa se encuentra todo el código relacionado con la interfaz gráfica la cual se encarga de representar los datos obtenidos en las salidas resultantes enviados desde el modelo. La información solicitada en éstas vistas tiene que pasar por el controlador, aunque en situaciones se puede suscitar que la vista tiene una relación directa con el modelo sin necesidad de pasar por algún controlador.

Controlador:

Esta capa es el medio de comunicación entre las peticiones que pide la vista hacia el modelo, centrando toda interacción entre la vista y el modelo, por lo general el código que maneja es para mostrar otras vistas, información que envía el modelo hacia la vista, etc., jamás tiene interacción directa con los datos que manipula el modelo. Esta capa depende de las necesidades que se requieran en la aplicación, y por ende el envío de las peticiones al modelo y el retorno del modelo hacia la vista dependerán de esas necesidades.

La utilización del modelo de desarrollo MVC es más frecuente en el desarrollo de aplicaciones web donde las interfaces pueden llegar a ser más complejas, en donde las vistas por lo general son paginas HTML, los modelos están relacionados con las bases de datos ya que la relación entre estos dos es muy estrecha, y por otro lado el controlador el cual se encarga de percibir los eventos de entrada desde la vista que son generados por el usuario.

3.1. Proceso del modelo de desarrollo MVC

Así como todas las metodologías y modelos observados en los capítulos anteriores, MVC también sigue cierto proceso el cual es necesario para que el software/aplicación quede funcional completamente, el siguiente diagrama ejemplifica la forma usual de cómo es la comunicación entre las tres capas (modelo, vista, controlador) y el usuario que hace las peticiones:

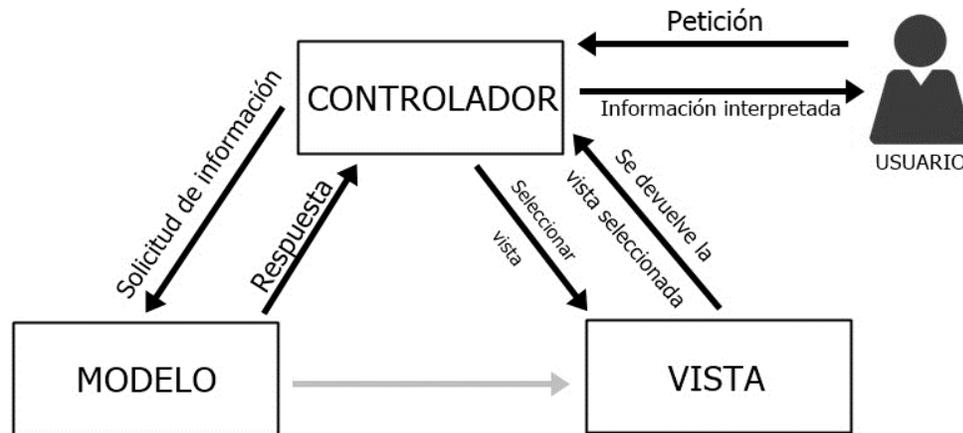


Figura 4. Representación general del modelo de desarrollo MVC.

Proceso:

1. El Usuario comienza realizando una petición en el sistema web misma la cual será enviada a través del controlador;
2. El Controlador recibe la petición que el usuario realiza; se encarga de la comunicación que hay entre el modelo y la vista; el controlador realiza la petición de los datos al modelo (desde consultas hasta ediciones de los datos), posteriormente selecciona la vista a mostrar y a su vez interpreta la información;
3. En ocasiones, la vista solicitará más información al modelo por lo tanto el controlador tendrá que interactuar hacia el modelo y de nuevo a la vista y así sucesivamente; todo depende de cómo se implemente la aplicación; y

4. Las Vistas mostrarán la información que el usuario requirió, aunque puede darse el caso de que al momento de mostrar una vista con datos provenientes de un modelo ésta vuelva a enviar una petición al controlador para obtener más datos.

Lógica de Negocio:

De forma general, la *lógica de negocio* establece ciertas reglas a ser cumplidas en el momento en el que se hace uso de algún sistema, por ejemplo, aplicándolo al modelo de desarrollo MVC se conoce que el usuario, al tener una interfaz gráfica se comunicará con el sistema para la obtención de información, por ende, al manipular la interfaz se ejecutan ciertos procesos en base a las acciones en el sistema. Aunque, por otro lado, también puede establecer los criterios permitidos y no permitidos para el manejo del sistema, actualmente conocidos como reglas de negocio.

Por lo tanto, en MVC la lógica de negocio se le atribuye a los modelos que contenga el sistema, ya que cada uno de los modelos se encarga de la comunicación con la Base de Datos ya que están programados de tal manera que saben la operación de cada acción, cada consulta, cada resultado, etc., durante la ejecución de los procesos en el sistema.

Lógica de Aplicación:

Así como el Modelo tiene sus reglas establecidas a seguir, lo mismo sucede con el controlador siendo ahora *lógica de aplicación*, ya que en ésta se establecen las reglas de comunicación para el controlador, argumentando que cuando el usuario realiza la petición desde la vista, ésta última se encarga de enviar dicha petición al controlador, el controlador envía la solicitud requerida por el usuario al modelo, obteniendo así la información y a la vez realiza el llamado a la vista correspondiente que mostrará la salida, puede que en la salida obtenida se realice el mismo proceso, es decir, llamar al modelo para pedir más información y mostrar en la misma vista o en alguna otra vista relacionada con la actual la salida correspondiente.

Además, la lógica de la aplicación ayuda a comprender más a detalle la estructura del sistema, ya que podemos conocer en qué parte de la aplicación se escribe o escribirá el código.

3.2. Sistemas anteriores a MVC

Anterior a la utilización del modelo de desarrollo MVC en los sistemas web, la estructura del código era completamente desordenada, ya que en código HTML se acostumbraba anexar el código PHP con procesos de lógica y en situaciones extremas también anexar código JavaScript, al tener estos tres tipos de códigos en un solo archivo se lograba que el código fuera denominado como: “*Código Espagueti*”.

La problemática se presentaba al momento en que un integrante del equipo con carecimiento de programación de la parte lógica de la aplicación interactuaba con el código de la aplicación (ej. el diseñador web), ya que su función es la del diseño de la interfaz del sistema/aplicación, por lo tanto él solo se encarga de manipular el código HTML, pero a la vez tiene acceso a las demás líneas de código que ahí hubiese (código PHP y/o JavaScript).

Tomando como base el breve ejemplo descrito en el párrafo anterior; resulta que cuando se hacían las páginas HTML con etiquetas de estilo incluidas, por ejemplo:

```
<font color="#FFFFFF">Texto aquí</font>
```

Y esa misma etiqueta se tenía en varias páginas y/o líneas más y si había que realizar cambios en el estilo por ende se tenían que editar todas y cada una de las etiquetas, lo cual hacía que el trabajo se demorara más. Ante esta situación se creó el código CSS² para englobar estilos por separados en archivo alterno al código HTML y así no tener que editar todas y cada una de las etiquetas necesarias dentro del archivo HTML.

² Lenguaje utilizado para describir la presentación de documentos HTML.

Lo anterior es un ejemplo que demuestra el por qué es importante tener el código separado y ordenado. Otro de los problemas que solían ser comunes era que una sola carpeta era la contenedora de todo del proyecto, es decir, demasiados archivos en una sola carpeta, sin importar de qué tipo fueran (.css, .js, .php, .html, etc.), lo cual de cierta manera dificultaba el rápido acceso hacia algún archivo que necesitara modificación y resultaba peor cuando habían archivos con un nombre similar.

No obstante, el desarrollo de los sistemas/aplicaciones web hace unas décadas atrás era totalmente diferente a como los conocemos actualmente, ya que no existía el lenguaje PHP, todos los desarrollos enfocados a la web se realizaban en Perl³, el cual existió tiempo antes de que el Internet fuera accesible para todo el mundo, sin embargo, poco después se creó el lenguaje PHP el cual logró que el desarrollo de sistemas y/o aplicaciones web tuviera un mayor auge.

3.3. Evolución de los sistemas

Con el pasar del tiempo, los sistemas se han ido actualizando conforme a las reglas de negocio así lo establezcan, uno de los factores principales del porqué de estas actualizaciones es la demanda de uso, la cual obliga a que cada sistema web sea más robusto, seguro y confiable. Por otra parte, el usuario es el factor clave más importante a tomar en cuenta, ya que las necesidades por parte de los usuarios se incrementan también conforme a la tecnología lo hace, por ende, los sistemas deben cubrir las necesidades que se lleguen a plantear.

³ Lenguaje de programación basado en un estilo por bloques, creado en 1987.

Pero no solo se trata de actualizaciones a los sistemas web, sino que también a todo lo relacionado con éstos, por ejemplo los servidores que se utilizan, los navegadores con los cuales accedemos a dichos sistemas, los gestores de bases de datos los cuales albergan la información, etc., y adaptaciones para nuevos componentes para sistemas que se encargan de almacenar el contenido multimedia como información.

Actualmente, cuando se desarrolla un sistema web se pueden encontrar herramientas bastante potentes, las cuales simplifican la forma de estructurar el sistema e incluso la forma en cómo escribir el código para lograr el propósito deseado. Existen herramientas tanto de Front-End⁴ como de Back-End⁵ que ‘simplifican’ la creación de ciertas secciones de la aplicación.

Frameworks:

Los Frameworks (marcos de trabajo) son herramientas que se han vuelto populares en el desarrollo de cualquier sistema, de los cuales se pueden encontrar bastantes opciones que se acoplan al usuario, una analogía muy certera es compararlo con MVC, ya que MVC separa todos sus componentes en tres capas: “datos”, “operaciones”, “presentación” y por otro lado, un Framework es capaz de definir los nombres de sus archivos, establecer las reglas de negocio para la codificación del sistema y la estructura misma.

Sin embargo, cabe aclarar que por lo general estos Frameworks no están ligados con algún lenguaje en especial, aunque no significa que no puedan hacerlo, tal es el caso en el lenguaje ‘*Ruby on Rails*’ en dónde: ‘*Ruby*’ es el lenguaje de programación y ‘*Rails*’ es el Framework. Lo interesante de estos Frameworks es que dependiendo de la necesidad que se tenga es como se utilizan, por ejemplo, se pueden detallar estructuras muy complejas para la creación de algún sistema y en otros casos solo utilizarlos en alguna sección de la aplicación.

⁴ Funciona del lado del cliente, es decir, lo que visualiza el usuario en el navegador (HTML, JavaScript, CSS).

⁵ Funciona del lado del servidor, es decir, interacción con bases de datos, lenguajes como PHP, Python, etc.

Lo anterior depende de qué tan necesario sea el Framework y para cumplir qué propósito, por ejemplo, *Ruby on Rails* es un Framework MVC que se dedica completamente al desarrollo total de sistemas web; mientras que *JavaServer Faces* (otro Framework) únicamente se centra en las interfaces gráficas para el usuario. Así como estos dos ejemplos se pueden encontrar muchos más, con diferentes paradigmas que resultan útiles.

3.4. Modelo de desarrollo MVC en PHP

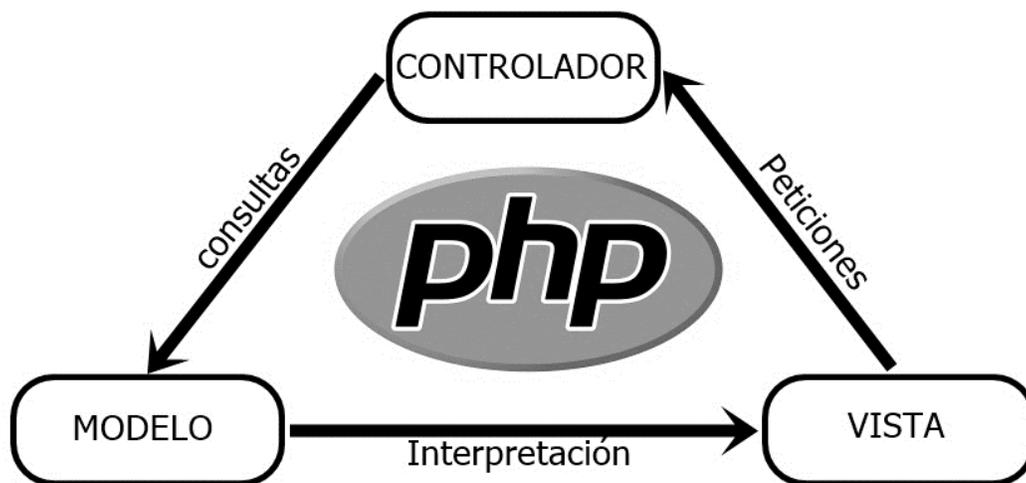


Figura 5. Esquema general MVC - PHP.

Implementar el modelo de desarrollo MVC en conjunto con PHP resulta sencillo, además de ser la combinación idónea al momento de realizar algún sistema web que realiza consultas con alguna Base de Datos. Lo anterior albergado en un servidor que provee un servicio de hosting⁶ y el albergue del dominio⁷ web.

Para empezar a describir la estructuración de MVC en conjunto con el lenguaje PHP de manera nativa se tiene que crear una estructura de archivos similar a la siguiente:

⁶ Servicio web en el cual se puede asociar un dominio.

⁷ Nombre único con el que se identifica a un sitio web en Internet.

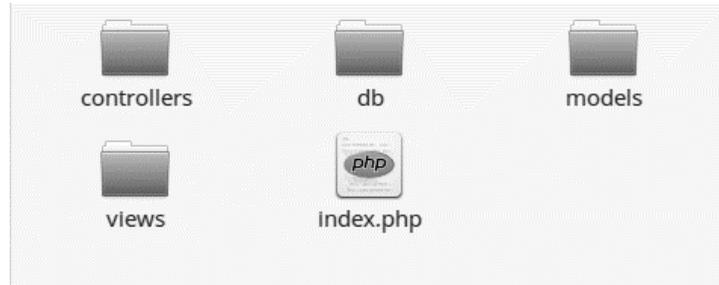


Figura 6. Estructura de archivos manejada por MVC.

De cierta manera es el primer paso para mantener la aplicación ordenada, en la cual:

1. **'models'** → Albergará las funciones encargadas con la interacción de la Base de Datos (consultas);
2. **'views'** → Todos y cada uno de los elementos de la interfaz gráfica (HTML, JavaScript, CSS) que el usuario visualizará, es recomendable que dentro de ésta carpeta se tengan carpetas que identifiquen a qué página se hará referencia, así como con los estilos tenerlos en su propia carpeta, etc;
3. **'controllers'** → Elementos que funcionarán como intermediarios controlando la lógica entre los modelos y las vistas;
4. **'db'** → Archivos correspondientes a la Base de Datos, tales como la conexión; e
5. **'index.php'** → La vista inicial al momento de acceder al sistema; desde ésta se redirecciona a la vista correspondiente dentro de *views*.

A partir de la estructura descrita anteriormente se puede empezar con la programación de cada una de las partes que respecta a la aplicación, siempre y cuando se respeten las reglas de negocio que establecen mantener separada la lógica de la aplicación de la interfaz, con la finalidad de lograr que una vez teniendo una aplicación funcional, y si se desea ir más allá de sus funciones actuales, es decir, expandir la aplicación, se tenga amplio conocimiento sobre en donde realizar las ediciones pertinentes para así lograr el resultado deseado.

Cabe aclarar que el modelo de desarrollo MVC con PHP puede ser creado de manera ‘pura’, es decir, que el programador cree las funciones a su manera, variables, parámetros, etc., ya que es una forma de trabajo que ayuda al programador a llevar un seguimiento en la estructura de la aplicación, de igual manera para tener un control sobre la misma y cuando se realicen mejoras o mantenimiento tener presente las partes del código a tratar. Además de que no es forzoso que se utilice la estructura de archivos mostrada anteriormente, la estructura mostrada es la misma que se utilizó para el sistema del cual se hablará posteriormente.

CAPÍTULO 4. Ofimática y Facturación Electrónica

“XML no es más lenguaje de programación que unas notas sobre una servilleta de papel.”

– Charles Simonyi

Anteriormente, jamás se hubiera pensado que la tecnología llegaría a ser una parte imprescindible en las empresas u organizaciones. Actualmente en una oficina se ha requerido de software especializado para la realización de las labores, sin importar que tipo de oficina o trabajo se cumpla, cumpliendo así con los nuevos estándares que manejan las empresas sobre la automatización de procesos, además contar con algún sistema o aplicación de este tipo ayuda sustancialmente al desarrollo de la misma organización.

4.1. Ofimática

Ofimática no es únicamente el utilizar software especializado para resolver ciertas tareas dentro de la oficina, sino que también engloba al hardware necesario para soportar la carga de este software. Por lo general, las aplicaciones destinadas al uso en oficina son actualizadas año tras año lo cual asegura que se tenga una mejor eficacia y eficiencia en la organización. A su vez, destaca la cantidad de herramientas que están a disposición de los usuarios, mismas que ayudan a agilizar el trabajo diario.

Teniendo en cuenta la importancia de la tecnología en la oficina surge la pregunta: ¿Desde cuándo se vuelve prescindible para la oficina? Simple, desde que la primera computadora es colocada en una oficina, ya que se derivan nuevas necesidades que impulsan hacia el uso de la tecnología, por ende, el colocar una computadora para un ambiente de trabajo derivó el desarrollo de aplicaciones que simplificaran el mismo, un claro ejemplo de aplicación son los procesadores de texto, hojas de cálculo, etc., que actualmente son presentados con interfaces bastante intuitivas que facilitan su uso.

La ofimática surge alrededor de los años 70 ya que era necesario “automatizar” los procesos que en ese tiempo se realizaban, comenzó con la implementación de máquinas capaces de tratar texto (máquinas de escribir) para las secretarías principalmente. Sin embargo, el concepto de

“automatizar” (en la época del mecanógrafo) tuvo que irse transformando desde el momento en que salieron al mercado las Computadoras Personales (PC).

Oficina Virtual

La analogía principal para describir a éstas Oficinas Virtuales es que tratan de emular las actividades que se realizan en una oficina convencional. Principalmente son utilizadas por Organismos Públicos en los cuales se brinda atención especializada a las personas que los consultan, principalmente para agilizar trámites que pueden ser realizados desde Internet sin tener que presentarse en alguna entidad que realice el mismo trámite. Son de gran ayuda ya que se encuentran funcionando 24/7 y la atención que brindan es completamente fácil de comprender, además de que cuentan interfaces intuitivas para el usuario.

4.2. Aplicaciones Informáticas & Ofimáticas

Como se mencionó en líneas anteriores, existen aplicaciones que están destinadas a ciertas tareas, por ejemplo, no sería prudente que un diseñador gráfico utilizara aplicaciones relacionadas con el procesamiento de texto para realizar su trabajo. Actualmente, existen aplicaciones para todo tipo de profesión, y la evolución de estas aplicaciones es notable que podemos encontrarnos con aplicaciones como procesadores de texto, hojas de cálculo, etc., disponibles para utilizar vía web.

4.2.1. Gestión Contable y Finanzas

Uno de los sectores principales en toda organización son los contables y los que se relacionan con éstos, por ende, contar con una aplicación informática que ayude a gestionar los procesos contables y financieros es de gran ayuda para tener control sobre estos recursos. Pero, ¿por qué es importante el contar con una aplicación informática en este sector? Sencillo, para una mejor toma de decisiones por parte de los directivos de la organización, ya que la información tratada en este tipo de sistemas es decisiva.

¿Cómo funciona un sistema de gestión contable?

Se debe tener una completa flexibilidad y relación entre el ‘costo - beneficio’, por lo tanto, se necesita un sistema que puede ser básico y a la vez puede ser un sistema muy complementado y bien planificado, lo que logra que estos sistemas de gestión contable, sin importar cuál sea el que utilice cada organización debe de cumplir con tres etapas que son cruciales ya que hacen uso de los recursos relacionados a las actividades financieras; siendo éstas:

1. Registro de la actividad financiera en la empresa;
2. Clasificación de la información; y
3. Resumen de la información.

Si bien otro requerimiento a cumplir es la información que debe ser registrada en estos sistemas, ya que a partir de la información registrada los ejecutivos toman las decisiones convenientes para la empresa.

Registro de la actividad financiera en la empresa:

Toda actividad comercial que es llevada a cabo en un día laboral debe ser registrada para que la gestión contable cumpla su propósito. Se sabe que en toda organización se realizan actividades comerciales de diferentes tipos, toda actividad se refleja en dinero, por ende, ésta debe ser registrada en su correspondiente cuenta contable; no todas las actividades comerciales influyen en el reflejo monetario.

Clasificación de la información:

Tener registros completos de toda actividad comercial realizada es fundamental en esta etapa de la gestión contable, por lo tanto, se requiere de bastantes datos los cuales servirán para la toma de las decisiones por parte de los directivos, es por esto que se deben de clasificar todas las actividades registradas y éstas separarlas por conceptos que las identifiquen más fácilmente, por ejemplo, separar las transacciones que emiten dinero de las que únicamente lo reciben.

Resumen de la información:

En la etapa anterior se menciona que se requiere de bastante información para la correcta toma de decisiones, pero la fase crucial se presenta en ésta, ya que aquí se resume la información a

tal grado de que es viable para que los directivos o personas encargadas tomen las decisiones favorables para la organización.

Software Libre Contable y Financiero:

A continuación, se muestran algunos ejemplos sobre software libre enfocado a la gestión contable y financiera:

- *FacturaScripts: Factura y Contabilidad de software libre para PyMES¹*.

Una aplicación realmente completa y poderosa para llevar la contabilidad y facturaciones que a la vez también cuenta con más de 80 plugins que se integran de manera ejemplar con la aplicación lo que logra que ésta sea más completa. El desarrollador de este proyecto pone al público en general el código fuente de su aplicación, así como guías de instalación, soporte, requerimientos necesarios para poder hacer uso del sistema, videotutoriales, actualizaciones, etc., además de ser multiplataforma.

- *GnuCash: Free Accounting Software²*.

Ésta es una herramienta libre para el manejo de finanzas y contabilidad para Micro y Pequeñas Empresas, ofreciendo herramientas necesarias tales como la ‘Contabilidad por partida doble’, ‘Cotizaciones’, ‘Transacciones Multidivisas’, etc. Este software además de que es multiplataforma también está incluido dentro de la Suite Ofimática ‘GnomeOffice’.

¹ <https://www.facturascripts.com/software-contabilidad>

² <https://www.gnucash.org/>

4.2.2. Soporte y Formación al Usuario.

Con el pasar del tiempo, el trabajo de oficina se ha vuelto cada vez más y más utilizado, por ende, la necesidad de tener aplicaciones o sistemas que ayuden a tener un control sobre toda la información que se maneje es esencial. Sin embargo, el contar una aplicación que simplifica alguno de los procesos de oficina implica que se debe tomar en cuenta los usuarios que harán uso de éstas aplicaciones, ya que son bastantes los programas o herramientas que pueden ser utilizados para mejorar los procesos que la empresa maneja.

4.2.2.1. Suites Ofimáticas

Dentro de la ofimática, los paquetes más importantes con los que se realiza el trabajo del día a día son los paquetes o suites ofimáticas. Una Suite Ofimática contiene una serie de programas o herramientas los cuales se utilizan para diferentes funciones, tales como la creación de documentos, organización de la información, impresiones, modificaciones, etc. Es común encontrar que en cada suite ofimática se tenga un procesador de texto y una hoja de cálculo, independientemente del paquete se pueden encontrar más programas, por ejemplo, como los que se utilizan para realizar presentaciones o incluso un sistema básico para bases de datos, además de otras herramientas que ayudan en las comunicaciones o en trabajos gráficos, correo electrónico, agendas, etc.

Suites Ofimáticas:

Dentro de las suites ofimáticas populares encontramos las siguientes:

- *OpenOffice*³: La primer suite ofimática libre creada por “The Document Foundation” utilizando el estándar OpenDocument; está conformada por: Writer (procesador de texto), Calc (hoja de cálculo), Impress (presentaciones), Draw (dibujo vectorial), Base (base de datos) y Math (editor de expresiones matemáticas).

³ <https://www.openoffice.org/es/>

- LibreOffice⁴: Es una versión alterna a OpenOffice, misma que tiene un código fuente más limpio y mejorado, incluye las mismas herramientas que OpenOffice pero anexándole un “LibreOffice” antes, es decir: LibreOffice Writer, LibreOffice Calc, LibreOffice Impress, LibreOffice Draw, LibreOffice Math; desarrollada por el mismo equipo de The Document Foundation.
- GnomeOffice⁵: Suite ofimática creada por el proyecto GNOME, ésta contiene más herramientas que las dos anteriores, ej. GnuCash, Dia.

4.2.2.2. *Multimedia y Presentaciones*

Actualmente la mayoría de las organizaciones que se dedican a la venta de algún producto o servicio lo hace mediante contenido multimedia profesional; ejemplos:

- GIMP⁶: Un editor de imágenes, perteneciente al proyecto GNU. GIMP es una alternativa libre a programas de edición de imágenes que pueden llegar a ser costosos, además de que la utilización de este editor es bastante sencilla, logrando resultados impresionantes.
- Natron⁷: Es una herramienta cinematográfica de código abierto de igual manera es una alternativa al software comercial, éste software es capaz de crear animaciones y efectos visuales de manera profesional. La forma de trabajo de este software es bastante diferente a los otros softwares conocidos que trabajan con la superposición de capas y edición de las mismas; éste trabaja con una composición basada en nodos.

⁴ <https://es.libreoffice.org/>

⁵ <https://wiki.gnome.org/Projects/GnomeOffice>

⁶ <https://www.gimp.org/>

⁷ <https://natron.fr/>

4.2.2.3. Seguridad

La seguridad es una de las partes más importantes en toda computadora, servidor, aplicación, etc. Se listan algunos a continuación:

- Wireshark⁸: Herramienta de código abierto que proporciona un análisis completo sobre nuestra red, con el fin de rastrear la navegación que se genera en nuestra red.
- Nikto2⁹: Es una herramienta que se dedica al escaneo de servidores web, trabaja realizando pruebas exhaustivas contra otros servidores web, para así determinar fallas, vulnerabilidades, aplicaciones que se encuentran desactualizadas.

4.2.3. Sistemas ERP

Enterprise Resources Planning (ERP) son sistemas dedicados a la planeación de recursos empresariales (de ahí el acrónimo ERP), con el fin de gestionar y controlar todo proceso, operación y recurso que la empresa utiliza y/o automatizar otros procesos existentes en la organización que requieren ser actualizados. Este tipo de sistemas se pueden ajustar en cada tipo de empresa sin importar si ésta es pequeña, mediana o grande, por el hecho de que están compuestos por módulos que se adecuan a cada tipo de empresa integrando cada departamento de la misma.

Algunos de los módulos que pueden ser integrados a los sistemas ERP se listan a continuación:

⁸ <https://www.wireshark.org/>

⁹ <https://cirt.net/Nikto2>

- Nomina;
- Bancos;
- Compras;
- Pagos y cobros;
- Inventarios;
- **Facturación Electrónica;**
- Almacenes; etc.

Los módulos mencionados anteriormente son solo algunos de los que pueden ser integrados a un sistema ERP, depende que enfoque se le quiera dar, por otra parte, todos estos módulos pueden estar conectados a una única base de datos o a varias dependiendo el departamento o la información a tratar.

Una clasificación general de los ERP es que éstos hay de dos tipos, los privativos o propietarios (de paga) y los gratuitos, los privativos son desarrollados por empresas que venden el servicio a un alto costo pagando en esencia por la licencia para hacer uso de éstos, además del pago de la licencia se le suma el pago de la implementación, contrario a estos existen los gratuitos, los cuales son desarrollados por comunidades de programadores los cuales liberan las versiones de los ERP que van construyendo sin costo alguno; a veces puede presentarse el caso que siendo gratuito se necesita la intervención de un “experto” que se relacione con el ERP para que lo deje funcional en la organización.

Implementar un ERP de manera correcta desde un inicio impacta en la productividad de la empresa y un aprovechamiento de tiempos mayor, esto último gracias a que los ERP ayudan en la creación de reportes que en ciertas ocasiones lleva demasiado tiempo elaborarlos. Por otro lado, la integración para la obtención de reportes de diferentes tipos, estadísticas, flujos, etc., todo reporte, estadística y demás se logra sin problema alguno aun así cuando la información cambia constantemente.

La importancia de los ERP influye en la toma de decisiones sin importar para qué fecha se tomen gracias a la integridad de la información y que siempre se encuentra disponible, es decir, la

información de tiempo atrás puede ser tomada como referencia para decisiones que vayan a ser tomadas.

Sistemas ERP de Open Source:

En ocasiones es preciso adentrarnos en los tipos de software que se pueden encontrar, por ejemplo, para la selección de algún sistema ERP también se tienen opciones alternas a los ERP que obligan a pagar una licencia con validez de cierta duración. Al querer implementar un sistema ERP en la empresa se deben tener en cuenta todas y cada una de las necesidades que tiene la empresa para tomar la decisión correcta en cuanto a qué ERP se elegirá. Algunos de éstos son:

- *webERP*¹⁰:

Sistema Open Source el cual está destinado a las PyMES (pequeñas y medianas empresas). Este ERP debe ser instalado en un servidor web (ej. apache) para que sea utilizado mediante navegador web.

Lo que hace que este ERP destaque por encima de otros más es el módulo contable con el que cuenta, además del control de inventarios avanzado que maneja, gestionando productos, precios, etc., reportes de varios tipos con importación a hojas de cálculo, archivos PDF. Por otra parte, la gestión de usuarios que hacen uso del ERP, ya que ajusta permisos para que el usuario únicamente acceda a las áreas que le corresponden.

- *PostBooks*¹¹:

No es más que otro ERP de código abierto el cual como en el caso anterior está destinado a las PyMES, éste tiene un cambio respecto al mencionado “*webERP*” y es que *PostBooks* no se instala en un servidor web, sino que debe ser instalado en una red local.

¹⁰ <http://www.weberp.org/>

¹¹ <https://www.xtuple.com/products/postbooks>

Una de las ventajas que tiene este ERP es la total integración de los módulos contables ya que alberga todas las cuentas contables que la empresa pueda tener, además de los reportes detallados que emite y la administración de cuentas bancarias.

Actualmente implementar un sistema ERP para la gestión empresarial resulta conveniente, ya que la toma de decisiones se simplifica, pero no solo eso, sino que brinda un status tecnológico a la organización que lo utiliza.

4.3. Facturación Electrónica

En México la facturación tradicional, es decir, facturas emitidas en papel fueron cambiando de acorde a las nuevas reglas establecidas por el sistema tributario de nuestro país (el SAT – Servicio de Administración Tributaria), volviéndose más exigente sobre el control de la facturación en México. La Factura Electrónica tiene la misma validez que la factura emitida en papel que sirve para avalar la adquisición de bienes o prestación de un servicio o producto, sin embargo, ésta debe ser emitida y recibida en formato electrónico.

¿Por qué se empezó a utilizar la Facturación Electrónica?

La Facturación electrónica comienza en México al rededor del año 2004 en donde el proceso de creación de los CFD (Comprobante Fiscal Digital) comenzó como una medida de seguridad para evitar la evasión fiscal; en el año 2005, el SAT era la única entidad que emitía las primeras Facturas Electrónicas. Mientras más contribuyentes incursionaban en la facturación electrónica ésta debía actualizarse, ante dicha situación en el año 2010 se crean los CFDI (Comprobante Fiscal Digital por Internet), y al mismo tiempo los Proveedores Autorizados de Certificación (PAC).

Otra de las razones por las cuales la facturación electrónica ha exigido cada vez más en México es evitar la evasión fiscal ya que un reporte emitido por el SAT detalla que entre los años 2007 y 2009 se presentó una evasión fiscal que ascendió a los 45,852 mdp. Además de que había un registro de 1,367 mdp de operaciones ilusorias, mismas que los contribuyentes no registraban. Por otra parte el proceso administrativo de emisión y recepción de facturas se simplificó con la

incursión de la factura electrónica gracias a que se logra la reducción en cuanto a la utilización de papel y tiempos de facturación.

Los formatos establecidos para la emisión y adquisición de las facturas electrónicas principalmente son dos, PDF y XML la importancia radica en que ambos archivos son indispensables cuando se haga uso de la contabilidad electrónica. El archivo XML contiene toda la información fiscal del emisor y receptor, así como la operación correspondiente ya sea de ingreso o egreso, conceptos, montos; el contenido en general de los CFDI depende de las reglas que el SAT establezca para su emisión. El archivo PDF es una representación impresa del CFDI (XML); no tiene validez fiscal.

4.3.1. Firma Electrónica y Firma Electrónica Avanzada

México, 2004 comenzaba con la facturación electrónica y a la par sucedió lo mismo con la Firma Electrónica, sin embargo, ésta surge de la necesidad de que las empresas quieren ampliar la seguridad para ciertos de sus procesos que empezaban a tener un manejo mediante dispositivos electrónicos (ej. para facturación), a esto se le suman los costos que pueden ser reducidos mediante la eliminación del papel.

Cuando se emitían facturas tradicionales era necesario que llevara una firma autógrafa para que tuviera validez, cuando se implementó la facturación electrónica se empezó a hacer uso de la denominada “Firma Electrónica”, ésta tiene la misma validez tanto jurídica como autógrafa de la persona que hace uso de ella. Al contar con una firma electrónica es posible realizar la facturación electrónica.

Posteriormente el nombre cambia a Firma Electrónica Avanzada (FIEL), la cual en un inicio presenta las mismas características que su antecesora; se actualiza para brindar más beneficios en cuanto a seguridad, por ejemplo, en esta nueva versión las transacciones electrónicas se vuelven más seguras, ya que la procedencia y autor puede ser verificado para comprobar que no se hayan presentado modificaciones durante la transacción.

4.3.2. Facturación Electrónica en ERP

Contar con un sistema de facturación electrónica es lo más demandado actualmente gracias a la cantidad de ventas que se realizan en las empresas. Se pueden encontrar demasiados sistemas los cuales siempre brindan una solución para realizar la facturación electrónica. Sin embargo, si se tiene un ERP, es indispensable contar con un módulo de facturación electrónica, y es que la contratación de un ERP en la actualidad más que nada se basa en la vinculación con la que estos cuentan para la integración de la facturación electrónica.

Además, el tener contratado un ERP con la facturación electrónica integrada es de gran ayuda ya que éste debe de mantenerse actualizado por parte del proveedor del ERP, adecuándose a las normas que establezca el Servicio de Administración Tributaria, en este caso en México.

CAPÍTULO 5. Caso práctico

“Si la depuración es el proceso de eliminar errores, entonces la programación debe ser el proceso de introducirlos.”

- Edsger Dijkstra

La “Empresa Automotriz S.A. de C.V.”, recurrió al L.C. Carlos Pineda Muñoz, para el posible desarrollo de un sistema que les ayudara en el control de cierta documentación que en ese entonces se llevaba en papel; recurriendo a mí para el desarrollo del mismo.

5.1. Planteamiento del problema

En la “Empresa Automotriz S.A. de C.V.”, se desea resolver la problemática de tiempo, es decir, que se beneficie tanto la empresa como a los proveedores de la misma, por lo tanto, se vuelve necesaria la creación de un sistema web con base de datos el cual realice la carga de las facturas generadas por la empresa, así mismo se realice la verificación y validación de las mismas. En otras palabras, que sea un intermediario entre el proveedor y el personal de la Empresa Automotriz S.A. de C.V.

El proceso general usando el sistema debe contar con las siguientes fases, además de manera general se describe todo el proceso que también incluye al personal correspondiente:

1. Carga y envío de los PDF y XML por parte de los proveedores al sistema de la Empresa Automotriz S.A. de C.V.;
2. Análisis de las facturas cargadas al sistema por parte del personal de *Cuentas por pagar* y *Contabilidad*;
3. Programación de pago; y
4. Pago y/o transferencia en tesorería.

A continuación, se describe el flujo del uso del sistema:

1. El proveedor tendrá que iniciar sesión en el portal mediante el usuario y contraseña que se le proporcionará por parte del personal del área de *Cuentas por pagar*;
2. El proveedor realizará el envío al portal web de la(s) factura(s) en formato PDF y XML, además del soporte de la factura;
3. Una vez obtenidos los documentos mencionados en el punto anterior, se procede al análisis de la factura, (duración aproximada de **un** mes);
4. Los documentos tienen que ser validados por las áreas de *Cuentas por pagar* y *Contabilidad*, pero se debe restringir dicha validación con que *Contabilidad* **NO** puede validar los documentos si *Cuentas por pagar* no ha validado los documentos;
5. Si las validaciones son exitosas, se programa la fecha de pago, misma que el proveedor podrá consultar en el sistema; *Cuentas por pagar* son quienes programan esta fecha; y
6. Una vez reflejada la fecha de pago, ésta tendrá que ser realizada en el área de *Tesorería* por parte del proveedor.

5.2. Análisis preliminar

La primer fase para la creación del sistema, era la realización de una investigación en conjunto con el personal solicitante de la creación del sistema que se involucraría (*Cuentas por pagar*) de la empresa, por tanto semana a semana, se realizaba una visita a la Empresa Automotriz S.A. de C.V., para tener en claro los requerimientos que el sistema debe ejecutar, uno de los procesos más importantes antes de empezar con la codificación del sistema era entender el flujo de todo el proceso para la recepción de la documentación fiscal.

Cada visita que se realizaba se obtenía una versión de mejorada de un documento en el cual se planteaban todas y cada una de las necesidades a solventar, así como la información necesaria para modelar la diagramación correspondiente a entregar con el sistema, por ejemplo, el primer diagrama que se tuvo que presentar fue el que explica todo el proceso de cómo se manejan las compras involucrando al proveedor y al sistema producto final como intermediario en el diagrama.

¿Por qué es necesaria la diagramación de todo el proceso?

A pesar de que en el proceso interfiere el factor humano, debe de realizarse para obtener el requerimiento adecuado para la programación del sistema, ya que incluso la tarea manual más mínima que pareciera no tener importancia se vuelve tan necesaria para el entendimiento del flujo de cómo se manejará el sistema.

Además, no solo este tipo de diagramas son indispensables, sino que todos los que serán mostrados más adelante son de suma importancia por si en algún futuro se desea realizar alguna modificación al sistema, ya sea mantenimiento, o agregar nuevos módulos que simplifiquen alguna otra necesidad que el personal desee solventar.

Para llevar a cabo el correcto análisis preliminar en la creación de este sistema se deben responder las siguientes preguntas:

1.- ¿Qué se busca resolver?

Al tener un sistema que ayude con el control de estos documentos se pretende solucionar el factor **tiempo** que conlleva la revisión de los mismos, y es que revisar estas facturas, tanto PDF, el XML y el “Soporte” de la factura (PDF) es un proceso tardado el cual para completarse se ocupa alrededor de un mes completo, sin embargo las horas hombre también son reducidas con la ayuda del sistema, ya que como se comentó con el personal de *Cuentas por pagar*, los proveedores tienen que acudir personalmente a las instalaciones de la empresa para hacer entrega de los documentos correspondientes.

Los proveedores, acuden y hacen entrega como se comentaba en el párrafo anterior, había situaciones en las que el tiempo de espera se extendía hasta dos horas, no por negligencia burocrática, sino por las mismas revisiones y/o aclaraciones con otros proveedores que se llegaban a extender bastante.

2.- ¿Cómo se realiza actualmente?

De forma breve, el proceso se lleva a cabo iniciando con la realización de un pedido (compra) al proveedor mediante los medios de contacto que se tengan para localizarlo, el proveedor genera la factura y hace la entrega del material en el almacén, mientras que *Compras* al validar la recepción del material procede a firmar el soporte de la factura, posteriormente se cita al proveedor para que haga entrega de la factura y el soporte con el área de *Cuentas por pagar* desde este punto, el proceso puede ser tardío, ya que la visita de más proveedores, termina provocando que los tiempos de espera para los tiempos de entrega de sus documentos se prolongue.

Cuando es la recepción de los documentos, es decir que el proveedor presenta la factura en PDF impresa y más aparte el Soporte de la factura que también es impreso, se debe realizar una revisión rápida pero minuciosa de ambos documentos, para así notificarle al proveedor que día puede acudir de nuevo para que se le informe la fecha en la que *Tesorería* realizará el pago correspondiente.

3.- ¿Por qué se desea solventar esta situación?

La Empresa Automotriz S.A. de C.V., principalmente requiere de este sistema para mejorar la calidad de servicio que les proporciona a sus proveedores, pero no solo eso, sino que también para tener un estatus tecnológico frente a su competencia y así tener una mejor eficiencia en los tiempos que se invierten para la recepción de este tipo de documentación.

5.2.1. BPMN (Business Process Modeling Notation)

*BPMN*¹ (Business Process Modeling Notation) o Modelado y Notación de Procesos de Negocios, se utiliza para demostrar gráficamente y de manera explícita todas y cada una de las tareas, mensajes, etc., describiendo la lógica de los pasos a tomar en consideración para la correcta descripción de los procesos de negocio. La utilización de esta notación es de gran ayuda para la coordinar las secuencias del proceso, por ejemplo, controlar los mensajes que surgen y fluyen para todos los participantes involucrados con sus respectivas actividades, reflejando la complejidad que se tiene en los procesos.

El fin de tener controlados los flujos de los mensajes y las actividades es para que todos los procesos y las partes involucradas se encuentren en constante comunicación de forma clara y concisa. Además, utilizar BPMN es recomendable ya que es un estándar de modelado aceptado a nivel internacional, además de que éste no depende de ninguna metodología para el modelado de procesos. Asimilándose a lo que viene siendo un Diagrama de Procesos de Negocio, BPD (Business Process Diagram).

5.2.2. Proceso general (Bonita BPMN)

A continuación, se presenta el diagrama general del proceso de un pedido/compra en la Empresa Automotriz S.A. de C.V., éste incluye el factor humano (tareas) indispensable para la operación, y como intermediario el sistema a desarrollar.

¹ <https://www.bonitasoft.com/>

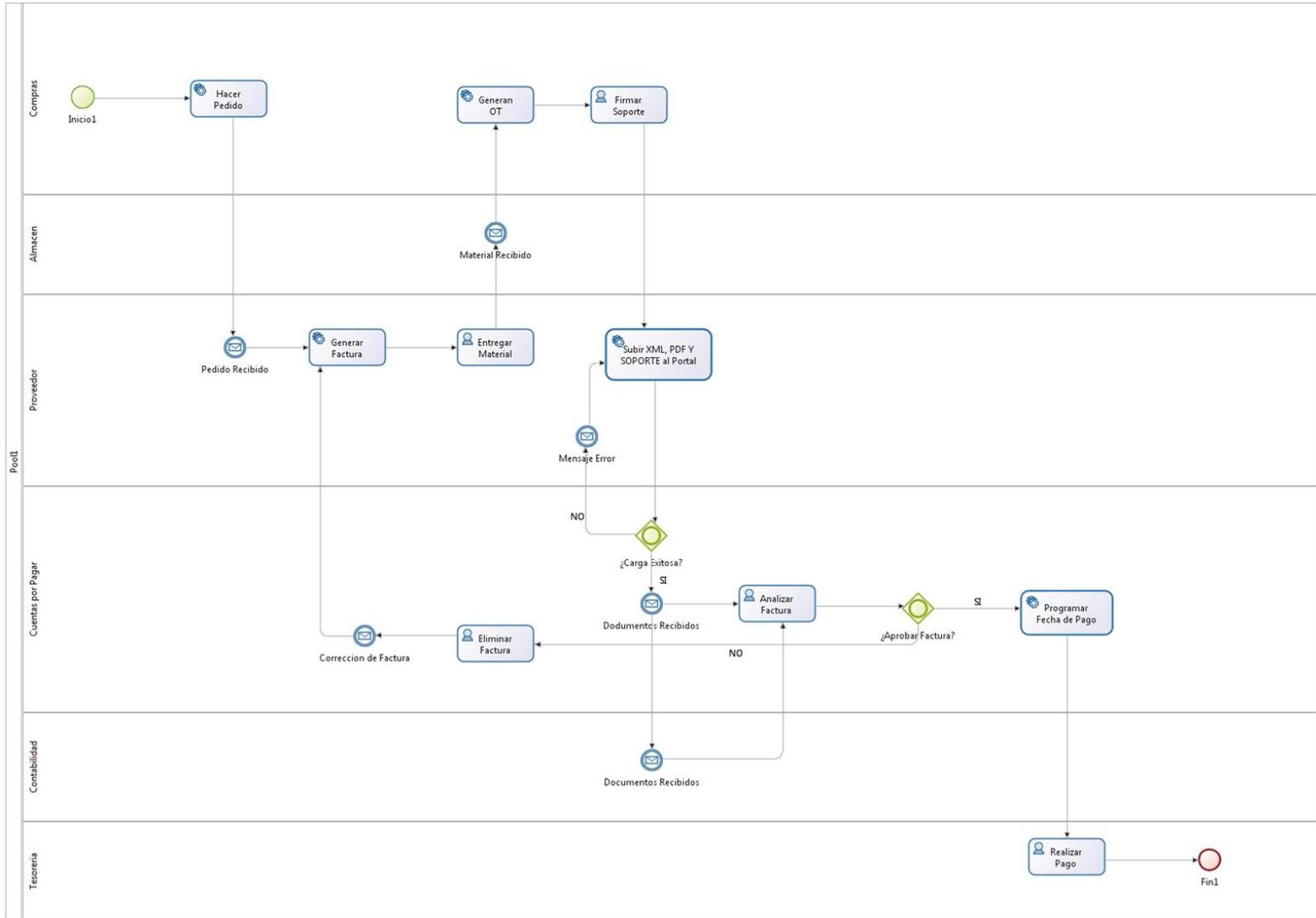


Figura 7. Proceso general.

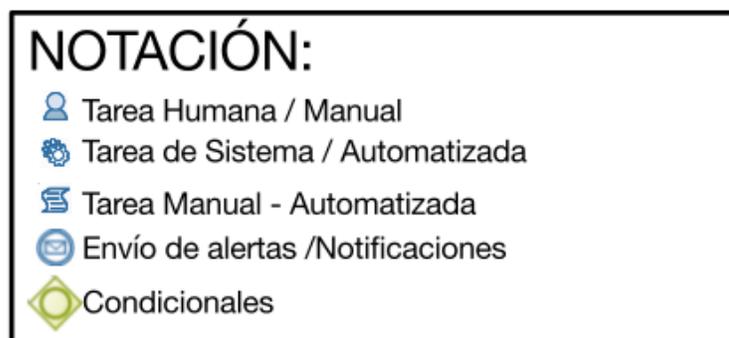


Figura 8: Notación del diagrama general.

Si bien, el proceso completo se describe a continuación:

1. El departamento de compras se encarga de realizar un pedido/compra;
2. Llega una notificación al proveedor, ya sea vía correo electrónico, mensajería, etc.;
3. El proveedor genera la factura correspondiente;
4. El proveedor hace entrega del material solicitado;
5. Una vez con el material en el almacén, éste notifica al departamento de compras sobre el material ya ingresado en el almacén;
6. Compras genera una OT (orden de trabajo) y a la vez firman el soporte de la factura;
7. Una vez obtenido el soporte de la factura con la firma correspondiente, el proveedor procede con la carga de la factura (archivos *pdf* y *xml* y soporte *pdf*) al portal que controlará el área de *cuentas por pagar* y de cierto modo *contabilidad*;
8. Si todo el proceso de carga resultó ser correcto, el personal de *cuentas por pagar* empieza con la revisión de la factura; de lo contrario se le notifica al proveedor mediante el sistema que existe algún problema en la carga de los archivos;
9. Durante el proceso de revisión de la factura, si se presentaron inconsistencias en ésta, *cuentas por pagar* elimina la factura, y hace la corrección de la misma al momento, notificando al proveedor;
10. Caso contrario al punto anterior, la revisión se realizó sin ninguna inconsistencia, entonces *cuentas por pagar* otorga el V.º B.º y prosigue que *contabilidad* haga su revisión pertinente;
11. Si *contabilidad* detecta algún error notificará a *cuentas por pagar* para que se realice la revisión correspondiente;
12. Caso contrario, *contabilidad* otorga el V.º B.º de la factura;

13. El área de *cuentas por pagar* programa la fecha en la cual el proveedor recibirá su pago correspondiente;
14. El proveedor deberá estar pendiente al sistema para que consulte y verifique cuándo recibirá su pago correspondiente por el servicio prestado; y
15. *Tesorería* realiza el pago.

Se especificó que únicamente se manejaran facturas, ya que Notas de Crédito querían mantenerlas controladas fuera de este sistema. Éstas notas de crédito las aplican en casos extraordinarios como en la devolución del material, situaciones que rara vez suceden.

5.3. Análisis y diseño del sistema.

Una de las tareas más importantes antes de que este proyecto fuera llevado a cabo era el tener en cuenta qué metodologías, herramientas, software, etc., eran de gran utilidad para la culminación del mismo. Sin embargo, lo más importante es la metodología empleada en este proyecto la cual es la “**Metodología de Desarrollo Ágil**”. Es de suma importancia especificar todas y cada una de las herramientas utilizadas en la creación de éste sistema de control documental, por lo tanto se listan a continuación:

1. SOFTWARE

1. Infraestructura LAMP.

1. Sistema Operativo Linux;
 2. Servidor Apache versión 2.4.18 (distribuciones derivadas de ubuntu);
 3. Lenguaje PHP en su versión 5.6;
 4. Motor de Base de Datos MariaDB versión 10.0.32; y
 5. Herramienta phpMyAdmin versión 4.5.4.1.
2. Modelo de desarrollo MVC;
 3. Diagramación BPMN.

5.3.1. Estructura interna de la aplicación.

Una vez instaladas las herramientas de trabajo descritas anteriormente, se procede a la realización del sistema.

5.3.1.1. Diagrama Entidad – Relación

Se detalla a continuación la Base de Datos utilizada en el sistema, ésta se estructuró con tres tablas las una de ellas se encarga de almacenar los datos de los *usuarios* de la empresa, mientras que la otra almacena los *proveedores* junto con los *archivos* que éstos envían para su respectiva revisión.

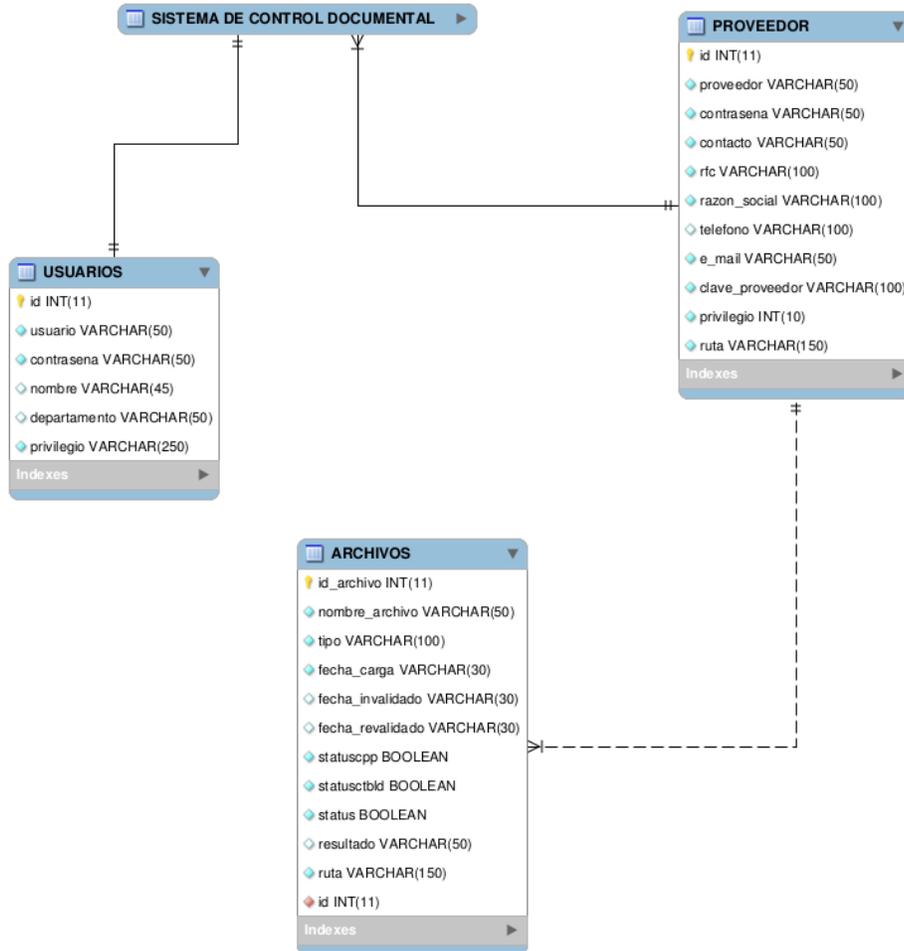


Figura 9. Diagrama Entidad – Relación.

5.3.1.2. Diccionario de Datos

El diccionario de datos que se mostrará a continuación nos ayudará a definir todos y cada uno de los campos establecidos en las tablas utilizadas en la Base de Datos del sistema, ya que éste diccionario de datos cumple la función de brindar información sobre los datos que se manejan, con a la finalidad de evitar ambigüedades y/o mala interpretación de los datos. El sistema de control documental, únicamente cuenta con tres tablas con las cuales cumple toda la operación deseada, siendo éstas:

USUARIOS			
CAMPO	TIPO	CONDICIÓN	DESCRIPCIÓN
id	int(11)	No Nulo, Auto incremental	Identificador único del usuario.
usuario	varchar(50)	No Nulo, Único	Nombre de usuario que se le asigna al usuario para que éste ingrese en el sistema, éste nombre de usuario debe ser único y no puede existir uno similar.
contrasena	varchar(50)	No Nulo	Contraseña de acceso al sistema que se le asigna al usuario.
nombre	varchar(250)	Nulo	Nombre completo del usuario que está haciendo uso del sistema.
departamento	varchar(50)	Nulo	Departamento al cual pertenece el usuario, además de que ya existe el campo privilegio que funciona de la misma manera, éste es menos restrictivo y solo es un campo informativo.
privilegio	int(10)	No Nulo	Privilegio que el usuario tiene para acceder al sistema, están definidos en la siguiente escala: <ol style="list-style-type: none"> 1. Privilegio 1. Se trata de un usuario con privilegios administrativos; 2. Privilegio 2. Se trata de un usuario del área de <i>Cuentas por pagar</i>, éste no cuenta con privilegios Administrativos; 3. Privilegio 3. Se trata de un usuario del área de <i>Contabilidad</i>.

Tabla 3. Diccionario de datos: 'Usuarios'.

PROVEEDORES			
CAMPO	TIPO	CONDICIÓN	DESCRIPCIÓN
id	int(11)	No Nulo, Auto Incremental	Identificador único del proveedor.
proveedor	varchar(50)	No Nulo, Único	Nombre de usuario con el cual el proveedor ingresa al sistema, éste nombre de usuario debe ser único e irrepetible
contrasena	varchar(50)	No Nulo	Contraseña que se le asigna al proveedor para que haga uso del sistema.
contacto	varchar(50)	No Nulo	Persona con la cual se mantiene un el contacto para cualquier detalle en relación con sus facturas.
rfc	varchar(100)	No Nulo, Único	R.F.C. del proveedor, no pueden existir registros de proveedores que tengan un mismo R.F.C.
razon_social	varchar(100)	No Nulo	Razón social a la cual pertenece el proveedor.
telefono	varchar(100)	Nulo	Número de teléfono de contacto del proveedor, ya sea algún número fijo o un número de celular
e_mail	varchar(50)	No Nulo	Cuenta de correo electrónico del proveedor, misma que también sirve de medio de contacto en dado caso de que la comunicación con él vía telefónica falle.
clave_proveedor	varchar(100)	No Nulo	Clave asignada por el personal de <i>Cuentas por pagar</i> ; ésta es manejada internamente.
privilegio	int(10)	No Nulo	Privilegio con el cual el proveedor accede a su página de inicio, por lo general los proveedores cuentan con privilegio 0 .
ruta	Varchar(150)	No Nulo	Ruta en la cual se guardan todos y cada uno de los archivos que el proveedor envía mediante el sistema, ésta ruta se construye en base al R.F.C. del proveedor, lo que hace que todos los documentos cargados por cada proveedor se mantengan en una ruta conocida a la cual los demás proveedores no tengan acceso.

Tabla 4. Diccionario de datos 'Proveedores'.

ARCHIVOS			
CAMPO	TIPO	CONDICIÓN	DESCRIPCIÓN
id_archivo	int(11)	No Nulo, Auto Incremental	Identificador único del archivo.
nombre_archivo	varchar(50)	No Nulo	Nombre del archivo, mismo que se obtiene al momento de ser enviado al servidor.
tipo	varchar(100)	No Nulo	Formato con el que cuenta el archivo, el sistema está restringido únicamente a aceptar los siguientes formatos: <ol style="list-style-type: none"> 1. PDF; y 2. XML.
fecha_carga	varchar(30)	No Nulo	Fecha y hora en la cual el proveedor realiza la carga de los documentos en el sistema.
fecha_invalidado	varchar(30)	Nulo	Fecha y hora en la cual el documento es invalidado por el personal de la empresa.
statuscpp	tinyint(1)	0	El valor predeterminado de este campo es 0 ya que indica que se trata de un documento nuevo el cual tiene que ser validado por el área de <i>Cuentas por Pagar</i>
statusctbld	tinyint(1)	0	El valor predeterminado de este campo es 0 ya que indica que se trata de un documento nuevo el cual tiene que ser validado por el área de <i>Contabilidad</i> , aunque contabilidad no podrá hacer la validación correspondiente hasta que <i>Cuentas por pagar</i> haya validado.
status	tinyint(1)	No Nulo	Estatus del documento; rangos: <ol style="list-style-type: none"> 1. 1 → Activo; 2. 2 → Invalidado.
resultado	text	No Nulo	Campo informativo en el cual se agregará el resultado obtenido en base a la revisión que se realiza de los documentos, puede agregarse texto sin limitación, fechas escritas manualmente, observaciones, comentarios, cancelaciones, etc.
ruta	varchar(150)	No Nulo	Ruta en la cual se encuentra el documento después de que fue cargado por el proveedor, nótese que una vez cargado el archivo éste no podrá ser modificado.
id	int(11)	No Nulo	Relación con el proveedor que realizó la carga, sirve para el filtrado de los archivos que realizan todos y cada uno de los proveedores registrados en el sistema.

Tabla 5. Diccionario de datos: 'Archivos'.

5.3.1.3. Diagramas

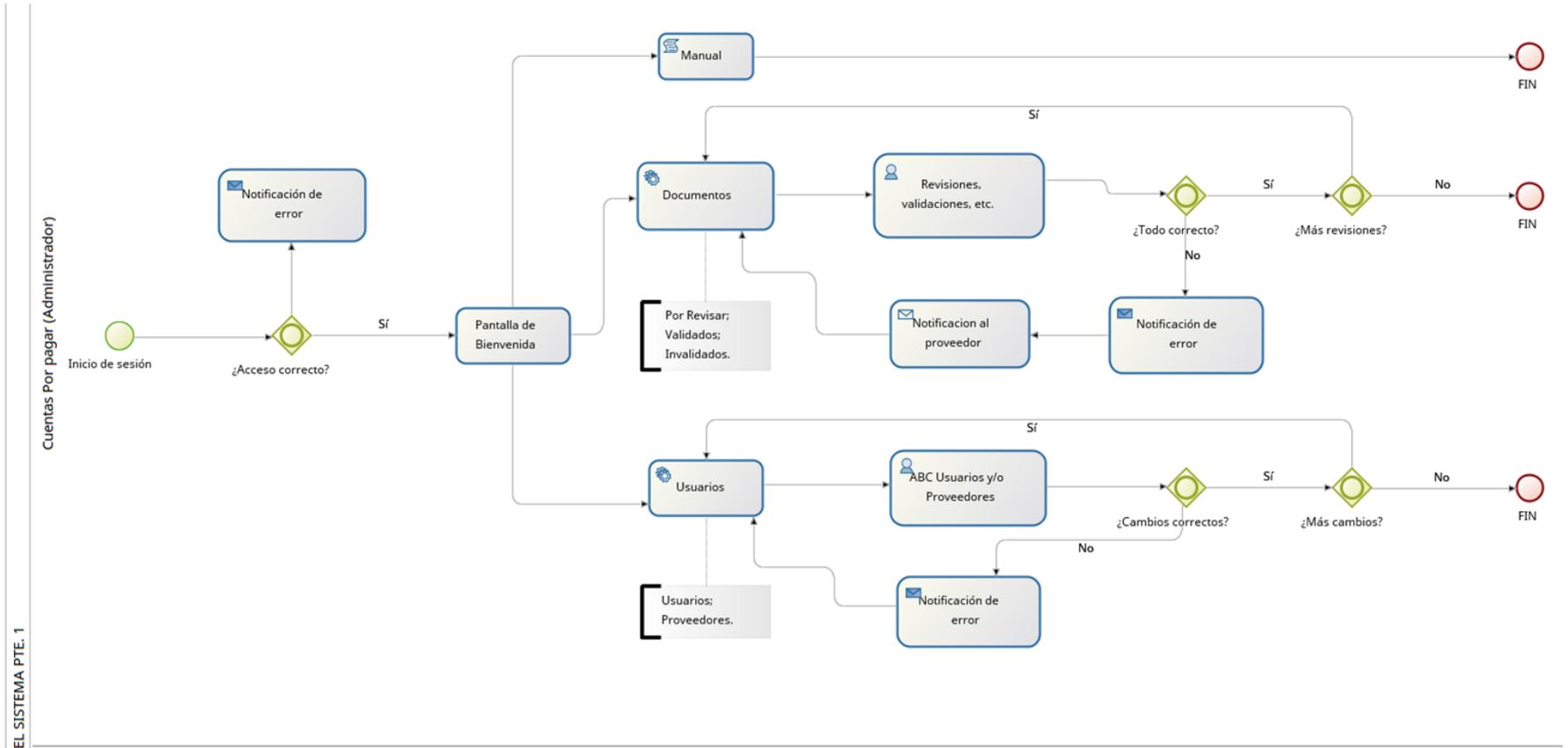


Figura 10. Proceso General del uso del sistema Pte. 1.

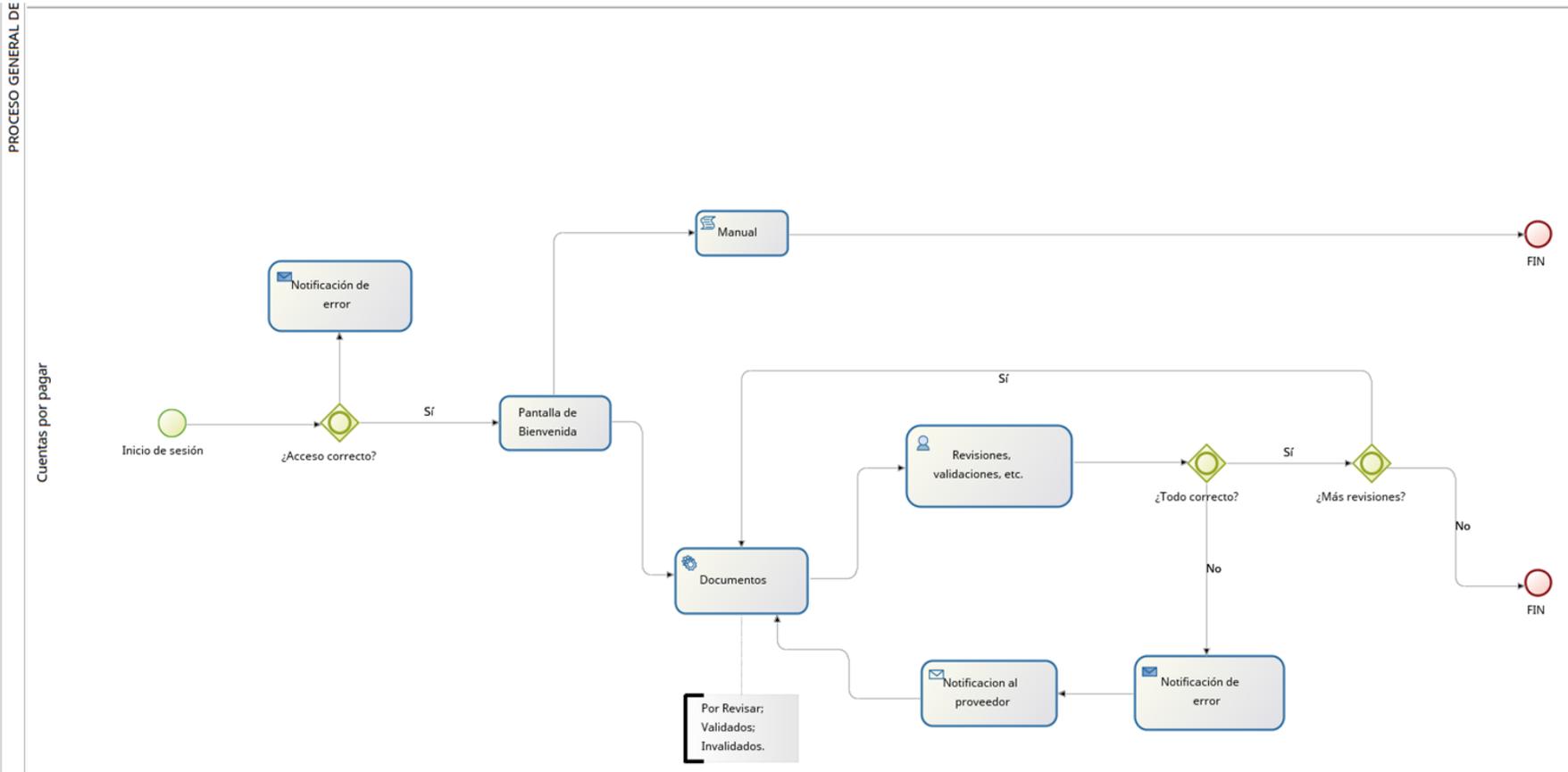


Figura 11. Proceso General del uso del sistema Pte. 2.

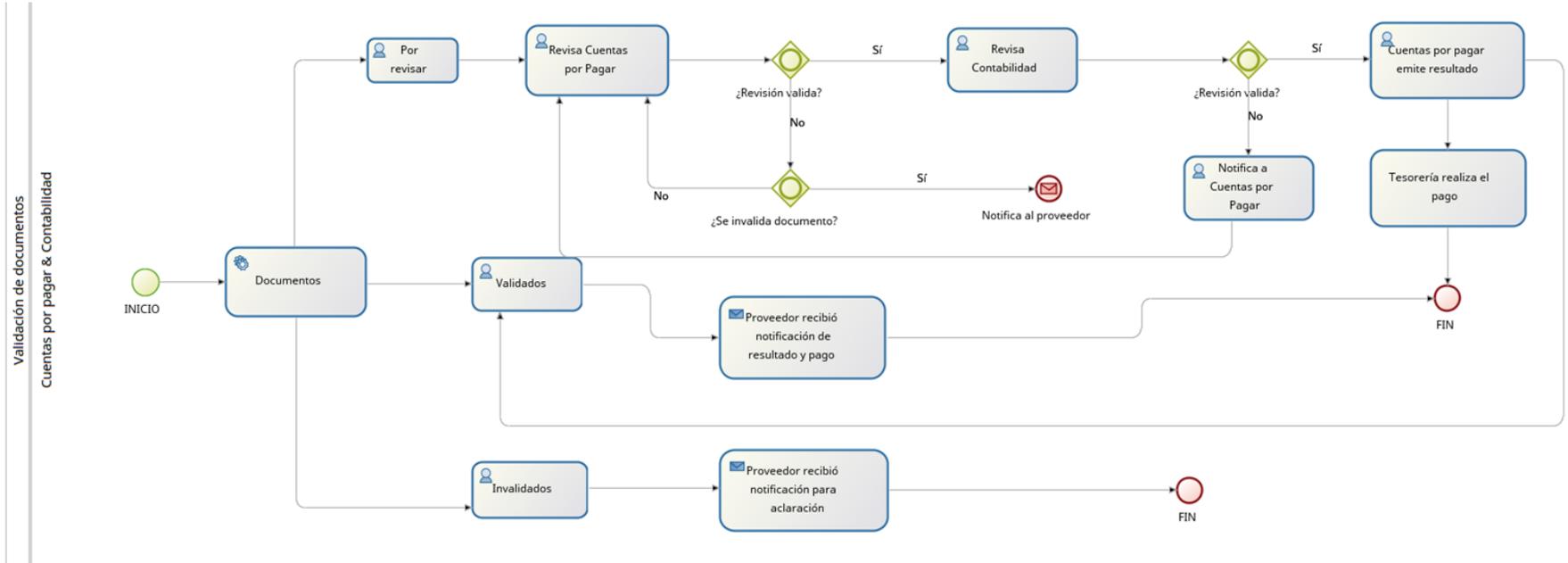


Figura 12. Proceso de validación de documentos.

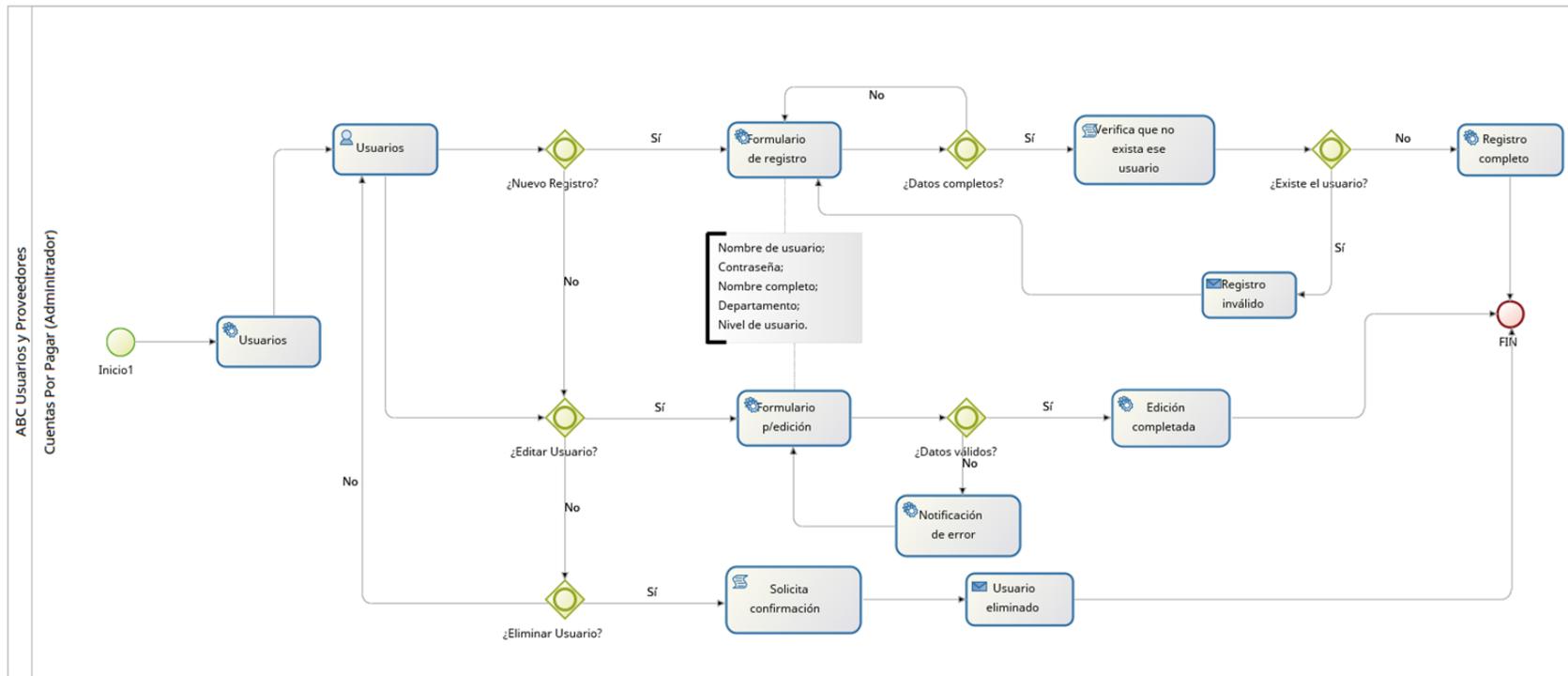


Figura 13. Proceso ABC Usuarios.

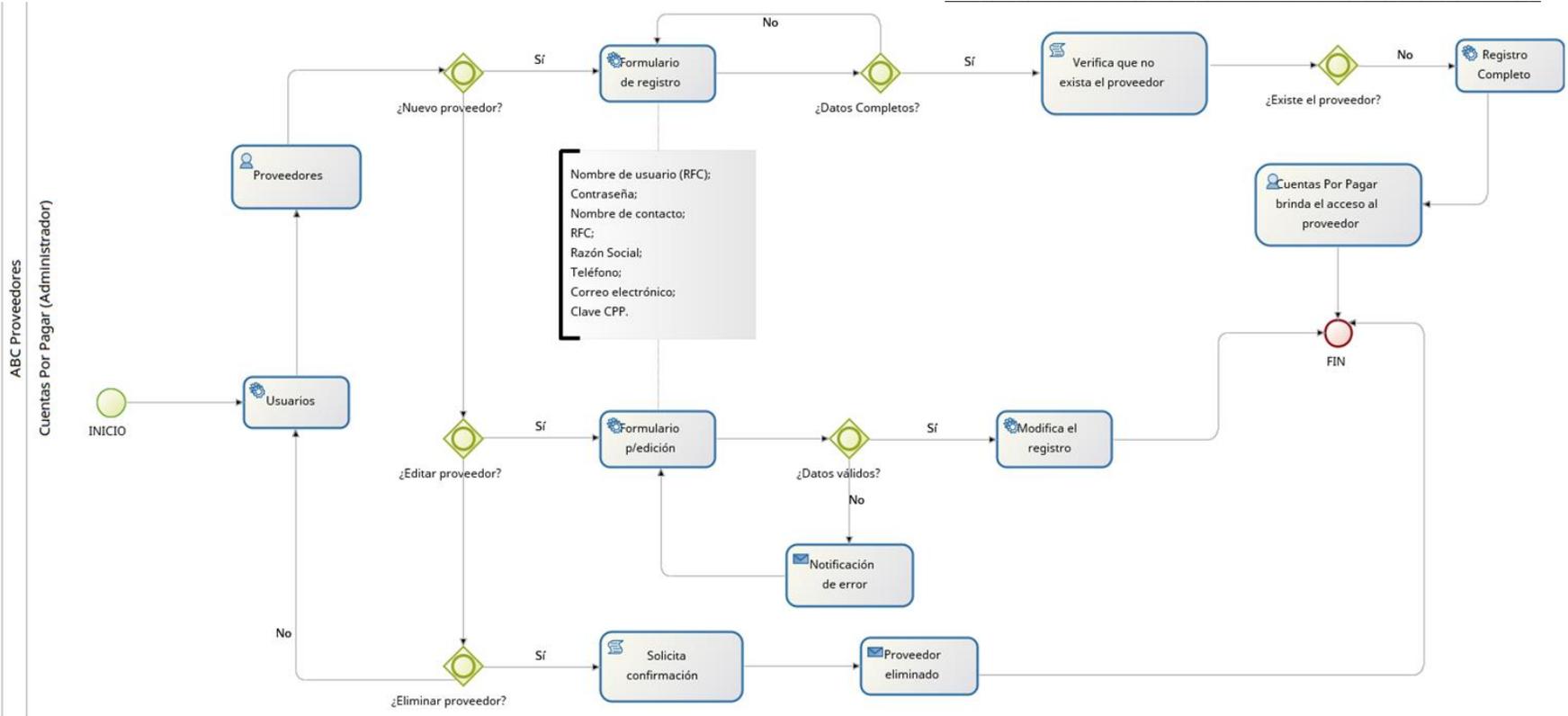


Figura 14. Proceso ABC Proveedores.

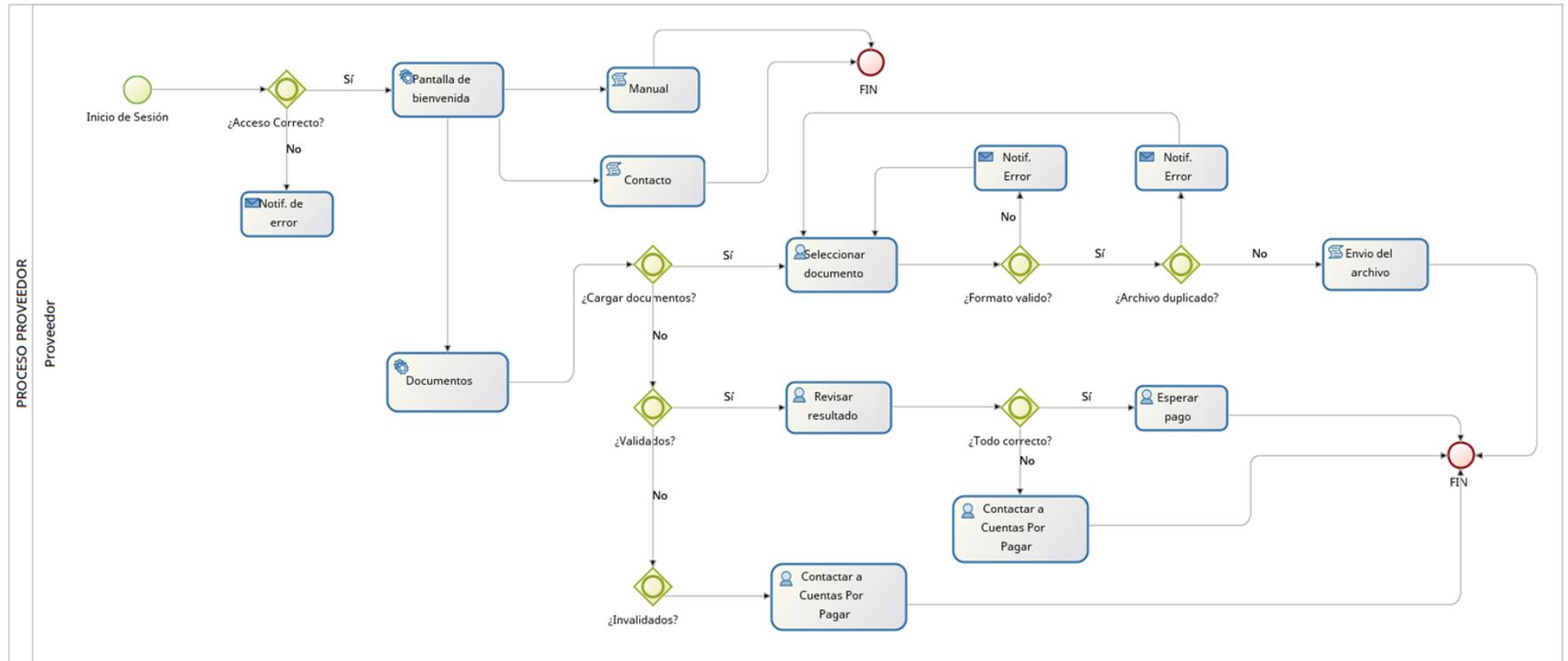


Figura 15. Proceso en el sistema como Proveedor.

5.3.2 Sistema Web de Control Documental

A continuación, se presentan las interfaces a las cuales el personal de la Empresa Automotriz S.A. de C.V. y los proveedores pueden hacer uso, es decir, se incluyen las interfaces que componen el perfil administrativo, el usuario de cuentas por pagar, el usuario de contabilidad y el proveedor.

Inicios de Sesión

Página inicial del sistema, en la cual se podrá seleccionar quién va a ingresar al sistema, es decir, en la sección “ADMINISTRADORES” Entra todo personal de la Empresa Automotriz S.A. de C.V. y en el botón “PROVEEDORES”, ingresan únicamente los proveedores.

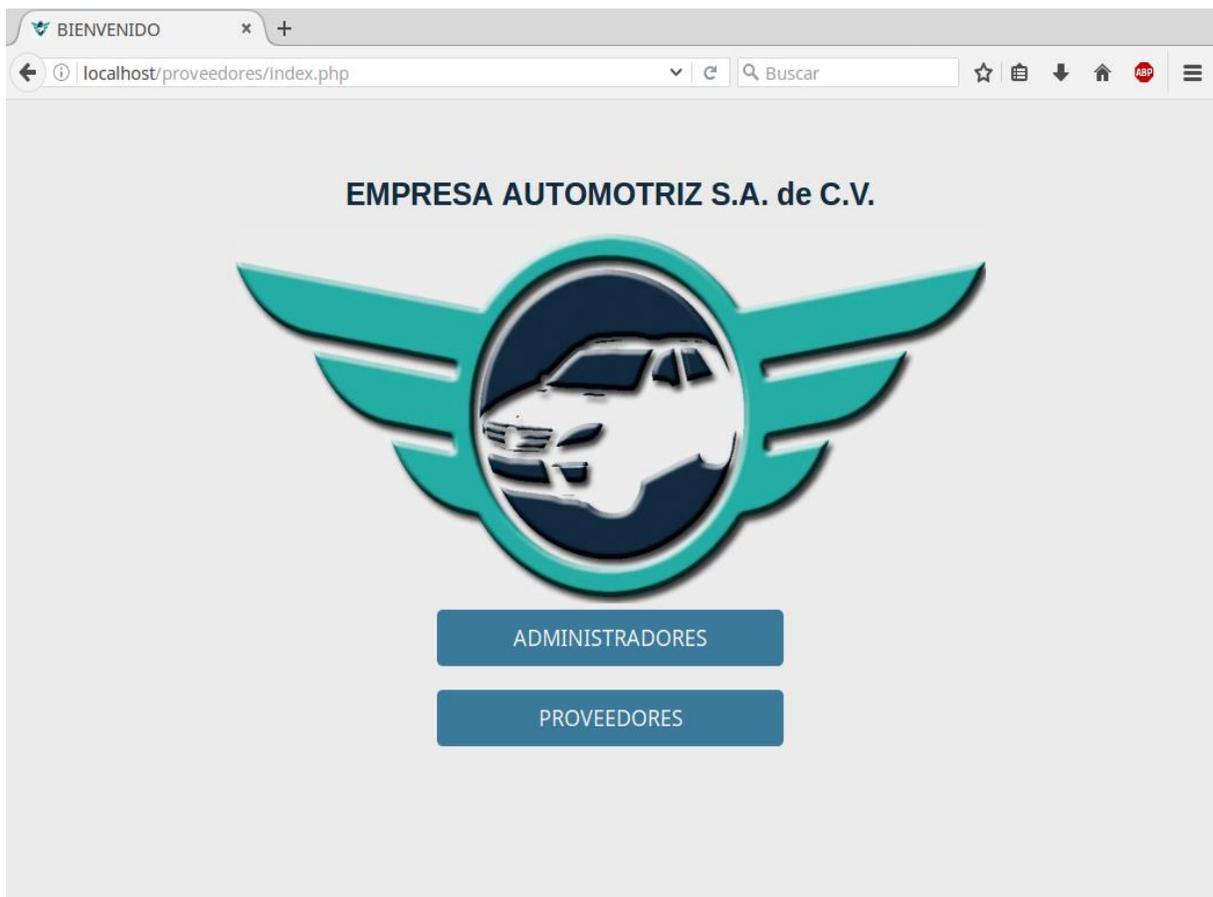


Figura 16. Pantalla inicial del sistema.

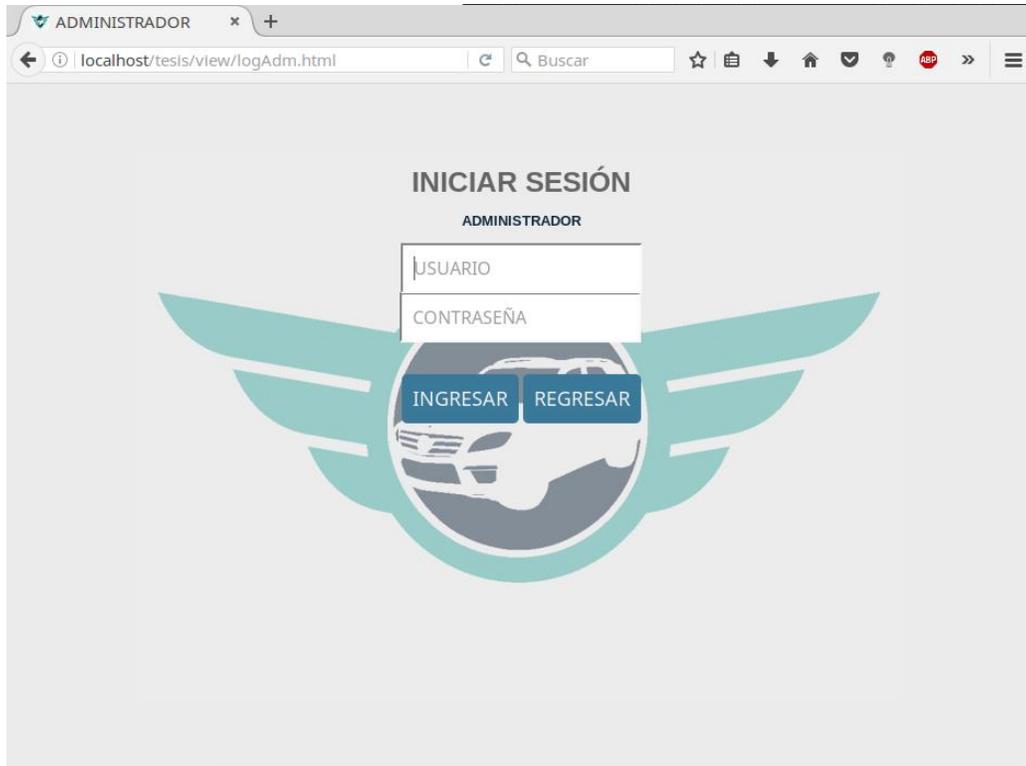


Figura 17. Inicio de sesión para el personal de la Empresa Automotriz S.A. de C.V.

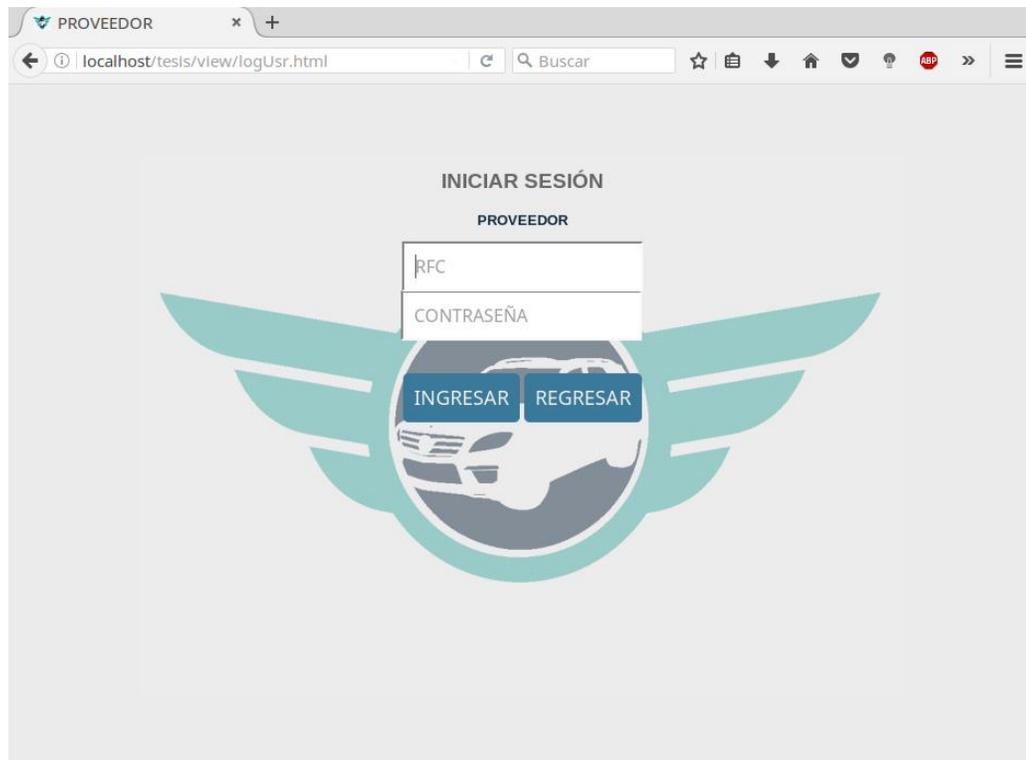


Figura 18. Inicio de sesión para los proveedores.

Bienvenida

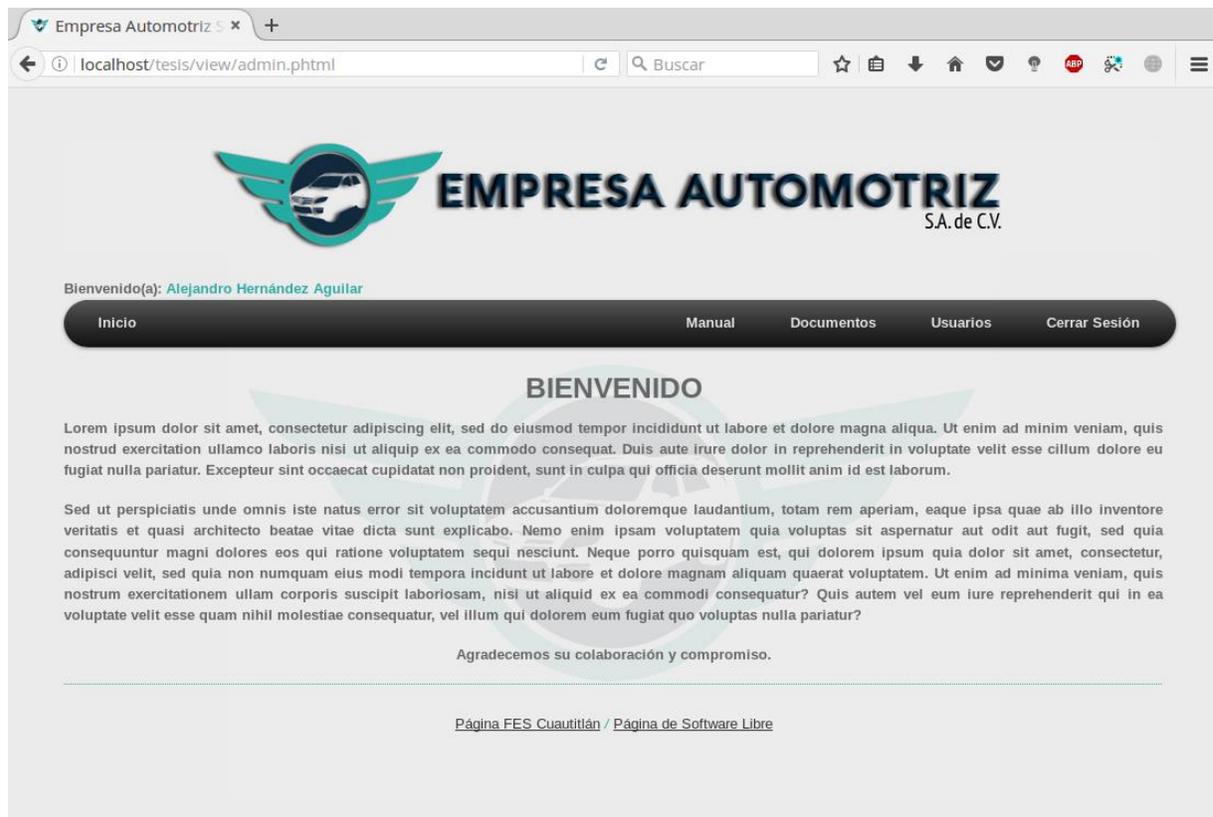


Figura 19. Pantalla de 'Bienvenida' una vez ingresado al sistema.

La pantalla de 'Bienvenida' se puede configurar al gusto, para ejercicio de este trabajo, se colocó el famoso texto *Lorem Ipsum*. El botón en la barra de menú 'Inicio' como tal redirecciona a la página mostrada en la figura anterior. Además, el pie de página fue editado por razones de confidencialidad, ya que en sí muestra hipervínculos a las páginas de la empresa que solicitó la realización de este sistema.

Manual

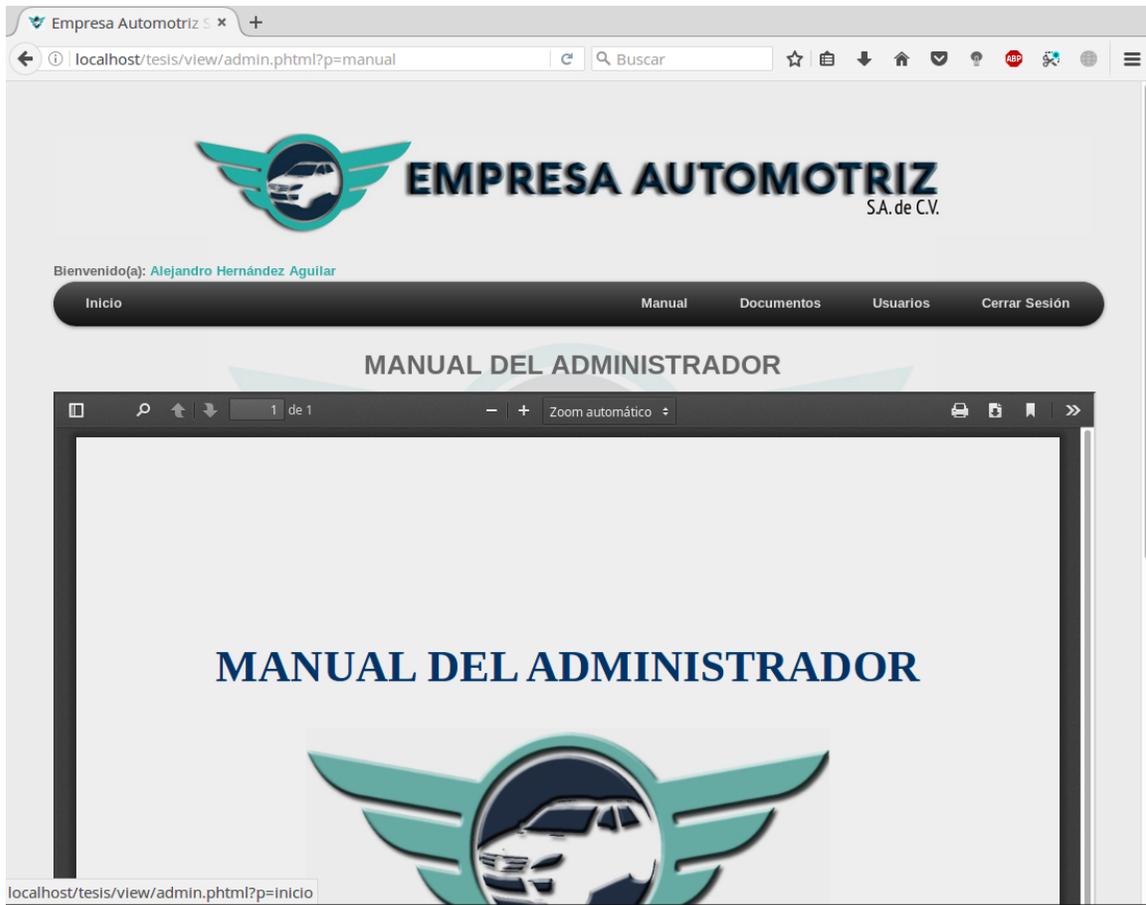


Figura 20. Interfaz donde se aprecia el manual del administrador.

Esta pantalla está dedicada exclusivamente para la documentación del sistema que los usuarios finales pueden consultar, es decir, los manuales de usuario en los cuales se establecen las métricas a seguir para el adecuado funcionamiento del sistema, además, dudas en cuanto a la operación también se encuentran detalladas en el mismo manual, para este sistema existen 4 tipos de manuales, siendo estos:

1. **Manual del Administrador.** En esencia es para el usuario de Cuentas por Pagar que tiene privilegios administrativos;
2. **Manual del usuario Cuentas por Pagar.** Éste no cuenta con los permisos administrativos que le permiten editar, agregar y/o eliminar usuarios y proveedores del sistema;
3. **Manual del usuario Contabilidad.** Únicamente para la validación de esa área; y
4. **Manual del proveedor.** Lo guía con la carga de los documentos, consultas, etc.

Usuarios → Usuarios → Registrar Usuario

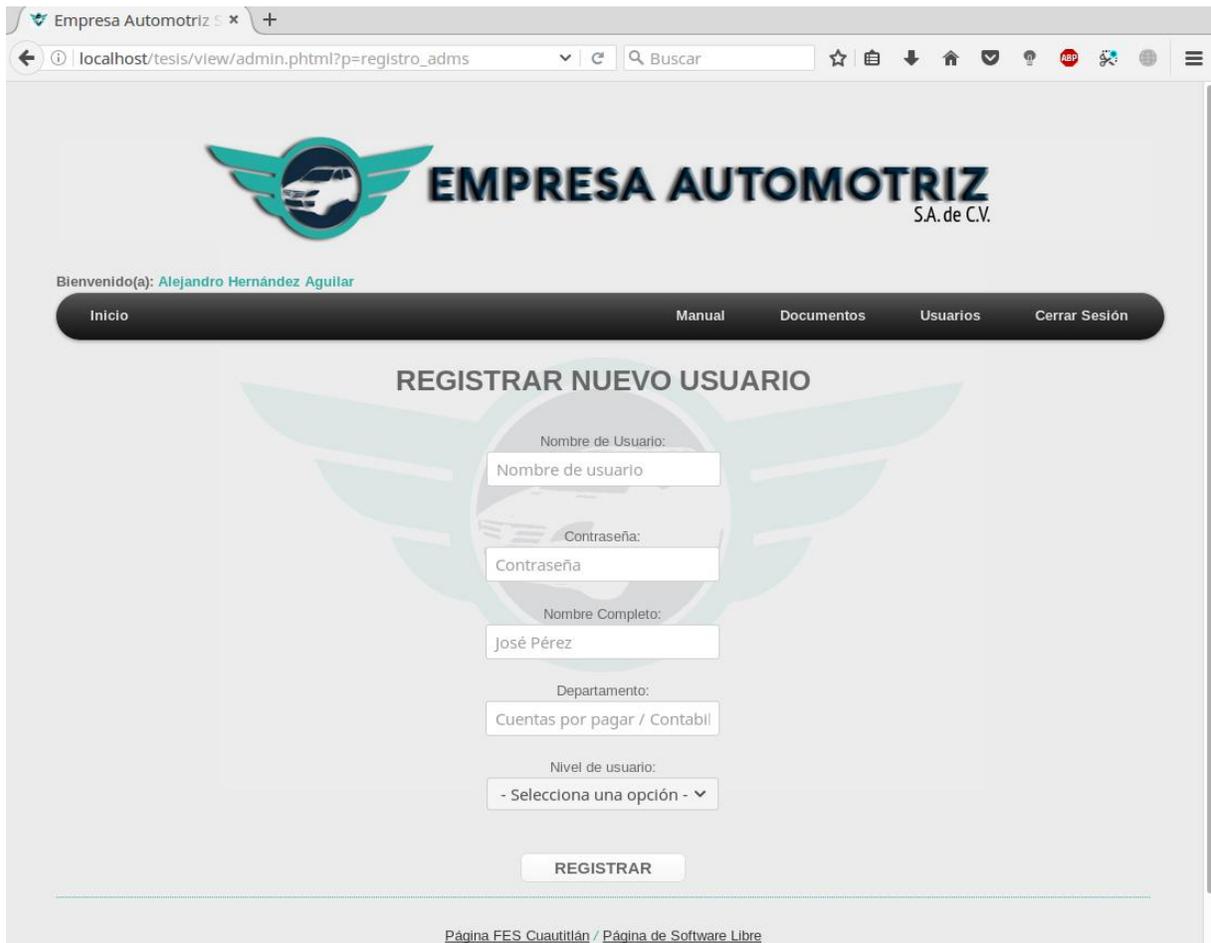


Figura 21. Interfaz para el registro de nuevos usuarios.

Esta interfaz está diseñada para el registro de nuevos usuarios “administradores”, es decir, para el personal de la Empresa Automotriz S.A. de C.V., cabe aclarar que el campo del nombre de usuario realiza una verificación previa al registro para comprobar que no exista un usuario con el mismo nombre de usuario; todos los campos son requeridos para completar el registro.

Usuarios → Usuarios → Usuarios Registrados

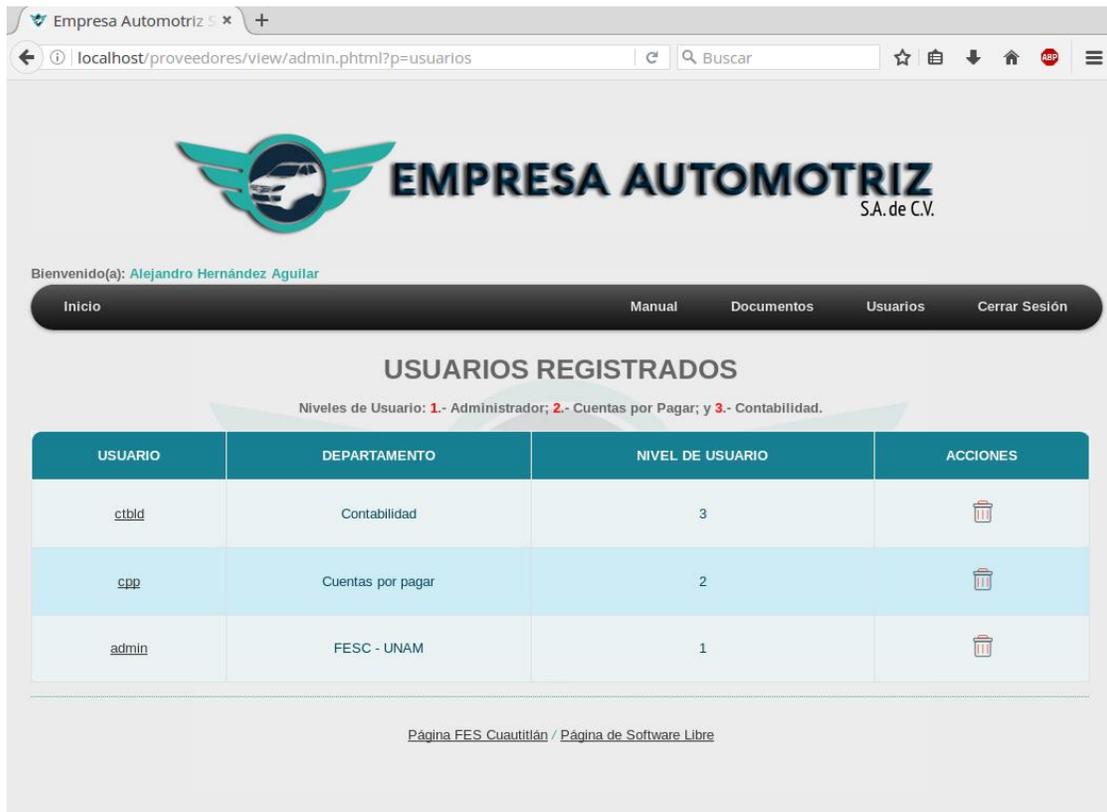


Figura 22. Pantalla donde se visualizan los usuarios registrados en el sistema.

En esta pantalla, se listan todos y cada uno de los usuarios registrados en el sistema, mismos que son pertenecientes a la Empresa Automotriz S.A. de C.V., desde aquí se gestiona a los usuarios, es decir, se puede realizar la eliminación y modificación de los usuarios.

EDITAR USUARIO

Nombre Usuario:

Contraseña:

Departamento:

Nombre:

Nivel de Usuario:

Figura 23. Formulario para la edición de algún usuario seleccionado.

Usuarios → Proveedores → Registrar proveedor

Empresa Automotriz S x +

localhost/tesis/view/admin.phtml?p=registro_prov

REGISTRAR PROVEEDOR

Datos para ingresar al Sistema.

Nombre de Usuario:
De preferencia RFC

Contraseña:
Contraseña

Información adicional del proveedor.

Contacto:
Contacto

RFC:
RFC

Razón Social:
Razón Social

Teléfono:
5555-5555

Correo Electrónico:
contacto@ejemplo.com

Clave CDC Proveedor:
Clave CDC Proveedor

REGISTRAR

Figura 24. Formulario para el registro de un nuevo proveedor.

El formulario para el registro de los proveedores realiza la validación de revisar en la Base de Datos que el nombre de usuario no exista previamente, se recomienda que éste sea el R.F.C. del proveedor, incluso el campo “R.F.C.” realiza la validación de que ambos campos (nombre de usuario y R.F.C.) contengan los mismos caracteres, ya que en base a éste se crea la carpeta en el servidor, carpeta en la cual se almacenarán todos y cada uno de los archivos que el proveedor envíe.

Usuarios → Proveedores → Proveedores registrados.



Figura 25. Pantalla donde se visualizan los proveedores registrados en el sistema.

Al igual que con los usuarios sucede lo mismo con los proveedores, desde esta pantalla se administran los proveedores registrados.

EDITAR PROVEEDOR

DATOS DE ACCESO AL SISTEMA:

Nombre Usuario
HEAA931104FX2

Contraseña
123

DATOS DE CONTACTO:

Contacto
José Pérez

RFC
HEAA931104FX2

Razón Social
Ejemplo Dos S.A. de C.V.

Teléfono
1234-4321

E-Mail
contacto@dos.mx

Clave CDC
LEPRF23

ACTUALIZAR CANCELAR

Figura 26. Formulario para la edición de algún proveedor seleccionado.

Documentos → Por revisar



The screenshot shows a web browser window with the URL `localhost/proveedores/view/admin.phtml?p=Ilistar_proveedores`. The page header includes the company logo and name "EMPRESA AUTOMOTRIZ S.A. de C.V.". Below the header, a navigation bar contains links for "Inicio", "Manual", "Documentos", "Usuarios", and "Cerrar Sesión". The main content area is titled "SELECCIONAR PROVEEDOR" and displays a table with two rows of provider data. Each row includes an "ACCIONES" column with a magnifying glass icon.

RFC	RAZÓN SOCIAL	CLAVE CDC	ACCIONES
HEAA931104FX2	Ejemplo Dos S.A. de C.V.	LEPRF23	
HEAA931104XXX	Ejemplo Uno S.A. de C.V.	FX1003	

[Página FES Cuautitlán](#) / [Página de Software Libre](#)

Figura 27. Selección de proveedor para consultar sus documentos.

Interfaz en la cual se listan los proveedores que pueden ser consultados para revisión de sus documentos previamente cargados.

Documentos → Por revisar → Consultar



Figura 28. Documentos a revisar.

En la figura 28 se pueden apreciar listados los documentos que entran a revisión, mismos que son enviados por los proveedores al sistema, cada uno de estos documentos debe ser validado por las áreas correspondientes, es decir, primero tiene que validar *Cuentas Por Pagar* y después *Contabilidad* para así programar una fecha pago, aunque en ese campo se pueden agregar observaciones extras, es decir, si se encontró un fallo en las facturas puede ser escrito en ese campo, mismo que el proveedor consultará periódicamente.

Se aprecian dos botones en “Acciones”, uno es para realizar la validación del documento por parte del departamento de *Cuentas Por Pagar* y el otro realiza la invalidación completa del documento; se estableció que únicamente *Cuentas Por Pagar* podría realizar las invalidaciones de los documentos.

De la misma manera, no se mostrará el botón para asignar el “resultado” hasta que el documento tenga las validaciones correspondientes.

Documentos → Validados



The screenshot shows a web browser window with the URL `localhost/proveedores/view/admin.phtml?p=validados&id=4&r=Ejemplo`. The page header includes the logo for 'EMPRESA AUTOMOTRIZ S.A. de C.V.' and a navigation menu with 'Inicio', 'Manual', 'Documentos', 'Usuarios', and 'Cerrar Sesión'. The main content area is titled '- Ejemplo Uno S.A. de C.V. - DOCUMENTOS VALIDADOS -' and contains a table with the following data:

DOCUMENTO	FECHA	CUENTAS POR PAGAR	CONTABILIDAD	RESULTADO DE LA REVISIÓN
FEJEM001.pdf	24-09-2017 17:03:27	✓	✓	Se realizará el pago el 25/09/2017.

Below the table is a 'REGRESAR' button and a footer with the text 'Página FES Cuautitlán / Página de Software Libre'.

Figura 29. Documentos validados.

Una vez que los documentos tuvieron validaciones exitosas tanto por el departamento de *Cuentas Por Pagar* y *Contabilidad*, se mostrarán en esta sección, la cual únicamente será de referencia para consultar qué documentos han sido validados y de la misma manera el resultado que se tuvo como consecuencia de la revisión de cada documento.

Documentos → Invalidados



Figura 30. Documentos invalidados.

Si durante las revisiones de los documentos se encuentran inconsistencias, el departamento de *Cuentas Por Pagar* invalidará los documentos, mismo que podrán ser consultados en esta sección, para el personal se le mostrará únicamente como se aprecia en la figura 30, mientras que al proveedor se le mostrará un mensaje predefinido. Sin embargo, de existir equivocación el documento puede ser revalidado de nuevo.

Usuario CUENTAS POR PAGAR y CONTABILIDAD.

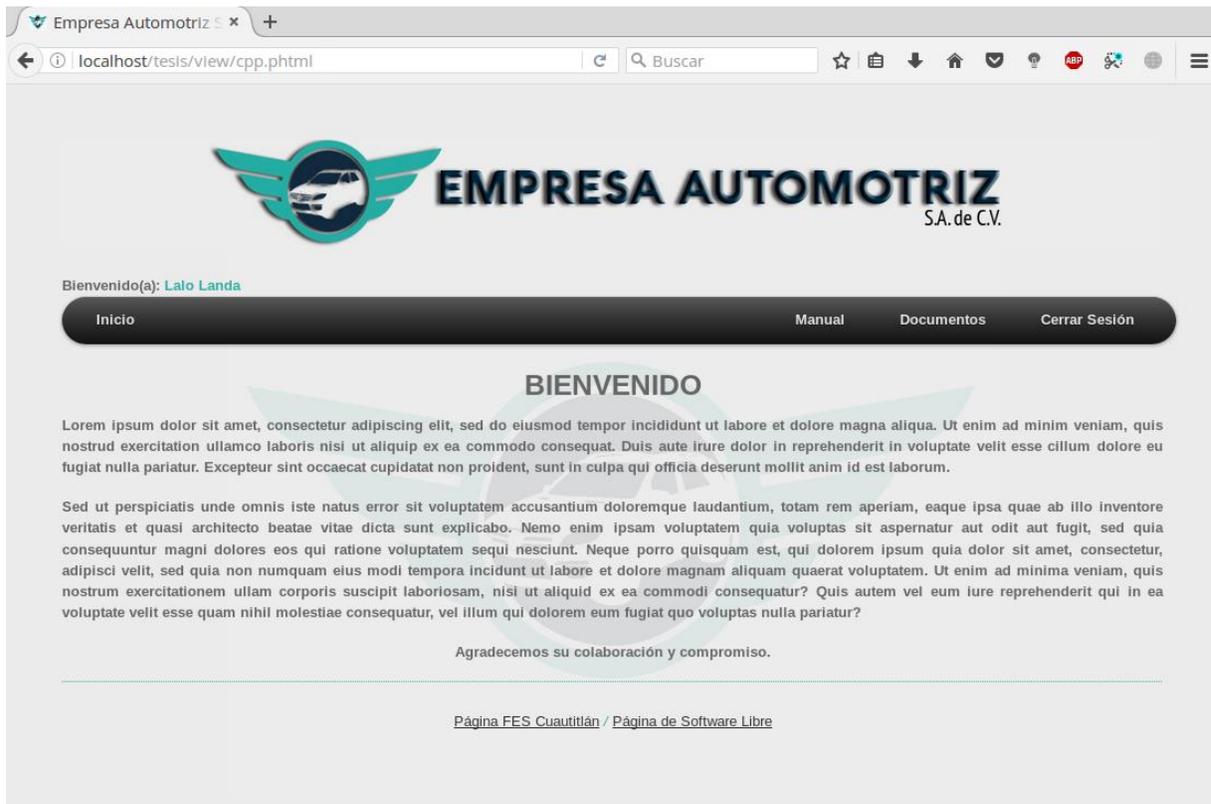


Figura 31. Pantalla de inicio para usuarios de Contabilidad y Cuentas por Pagar.

Para los usuarios de *Cuentas Por Pagar* y *Contabilidad* se les muestra la misma interfaz, ya que éstos usuarios no cuentan con privilegios administrativos, de ahí en fuera el flujo de operación sigue siendo el mismo, a diferencia de que el usuario de *Cuentas Por Pagar*, sin privilegios administrativos puede colocar el resultado obtenido durante la revisión, de la misma manera realizar la invalidación de los documentos, mientras que el usuario de contabilidad únicamente se dedica a validar los documentos que el proveedor carga al sistema.

PROVEEDOR

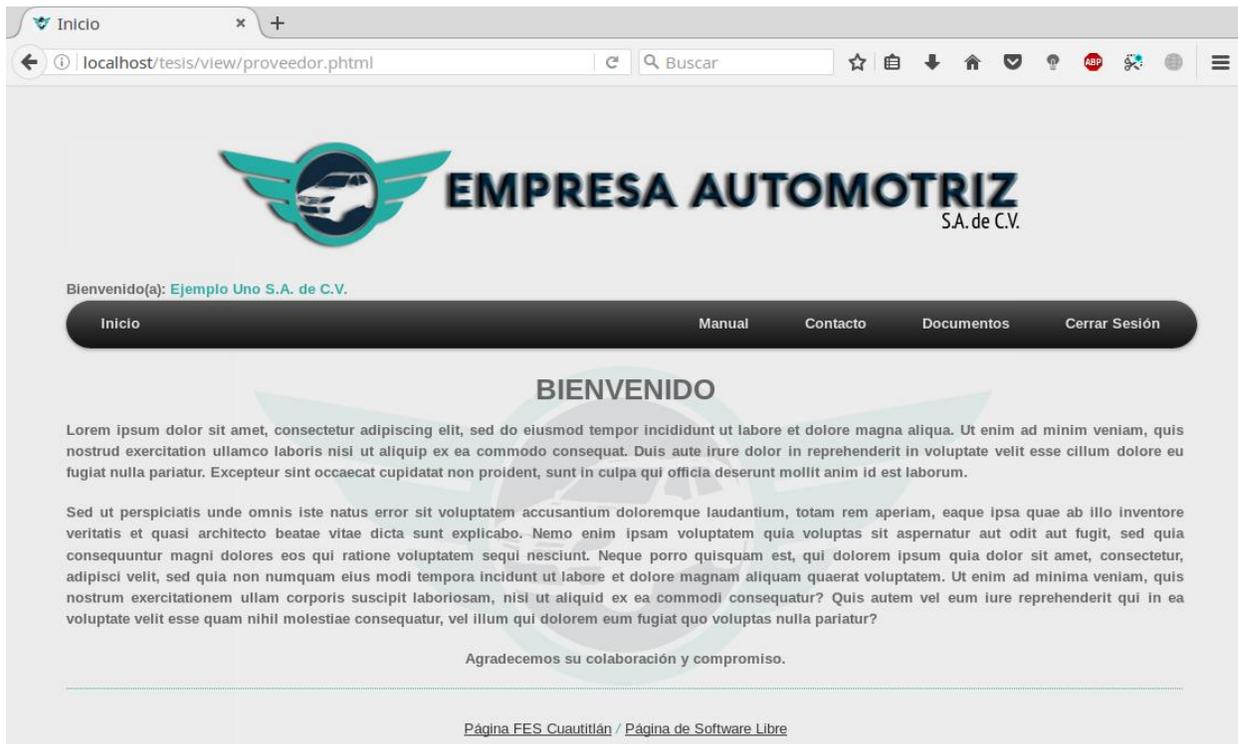


Figura 32. Pantalla de inicio del proveedor.

Básicamente la interfaz general sigue siendo la misma que se ha venido mostrando anteriormente, sin embargo, el proveedor cuenta con dos secciones que el personal de la Empresa Automotriz S.A. de C.V. no cuenta, siendo ‘Contacto’ y ‘Documentos’.



Figura 33. Interfaz de la sección de ‘Contacto’.

Documentos → Cargar documentos

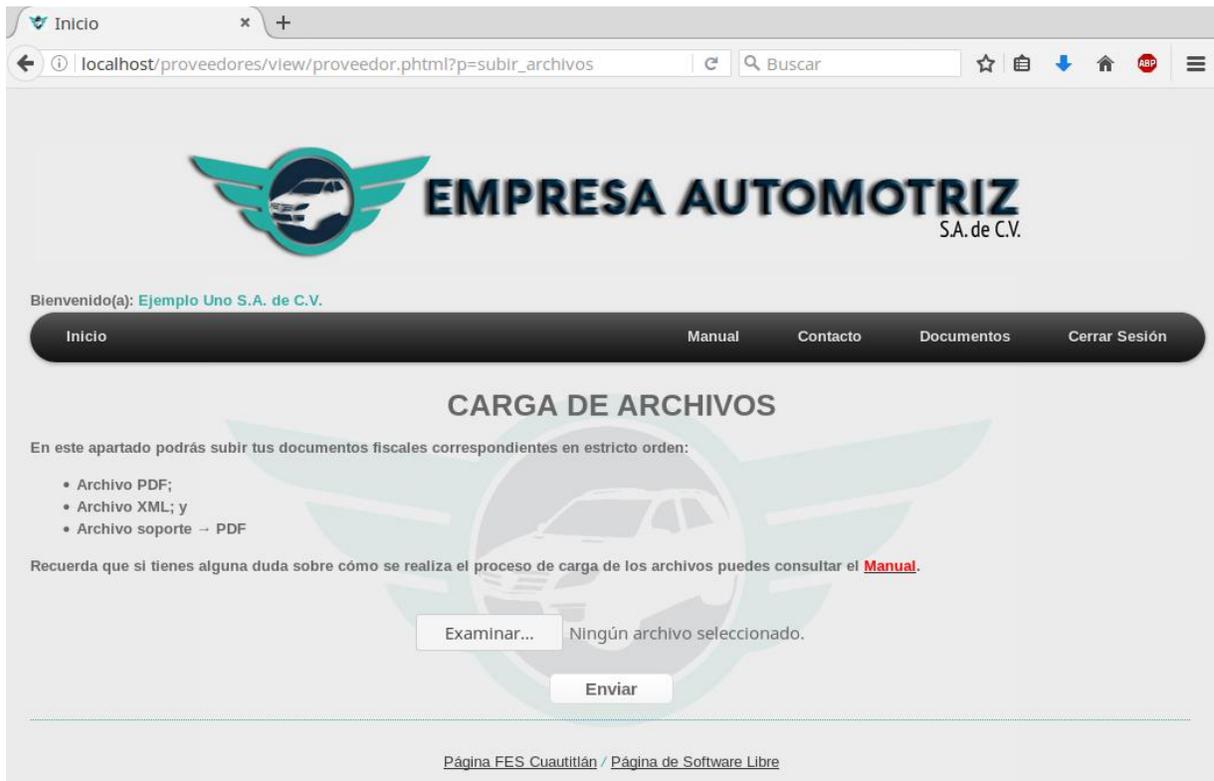
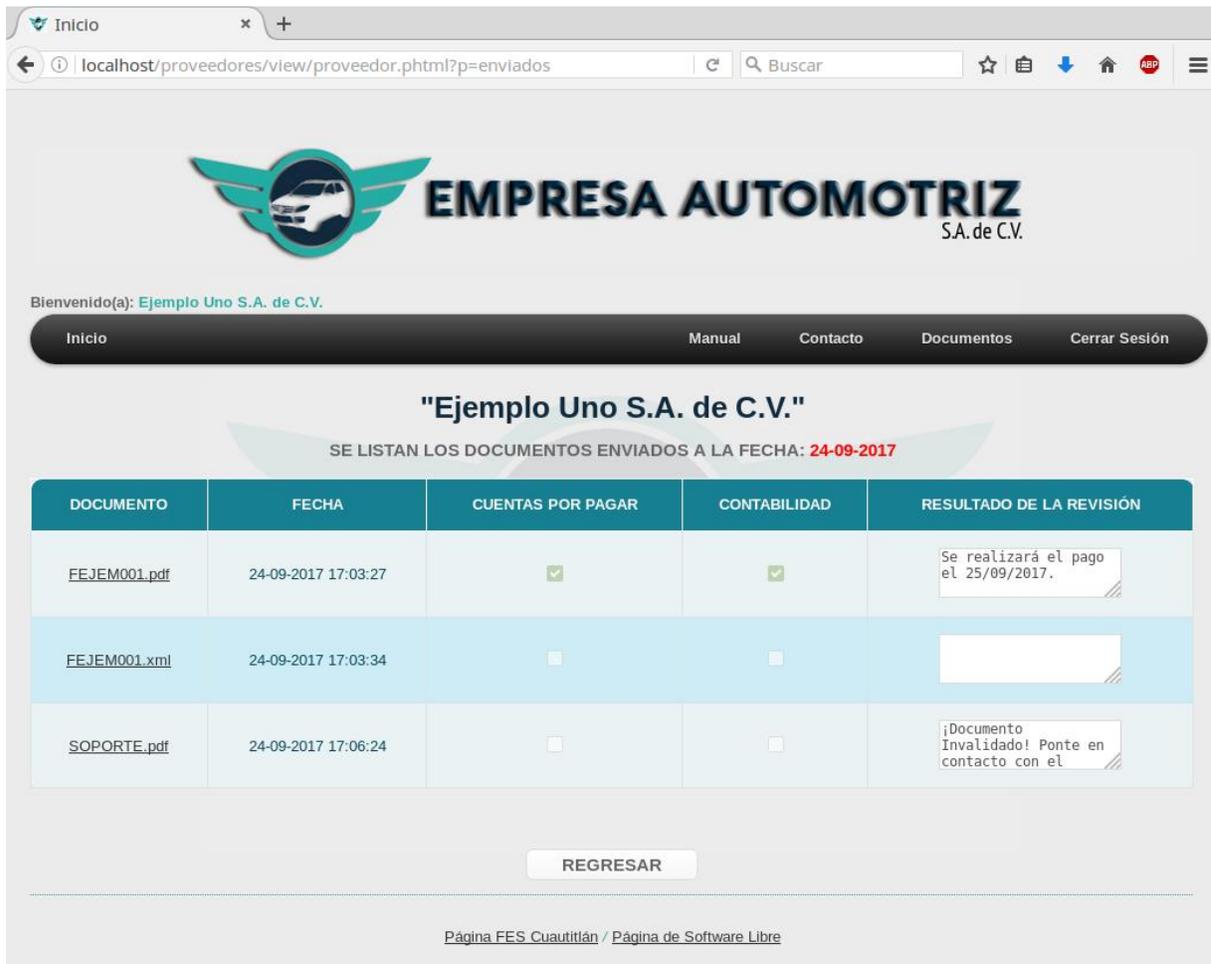


Figura 34. Interfaz para el envío de documentos.

La función primordial de esta pantalla es el envío de los documentos que el proveedor tiene que enviar a la Empresa Automotriz S.A. de C.V. para su revisión, el proveedor debe tener cuidado de qué envía, ya que una vez cargado el(los) PDF y/o XML no habrá manera de modificar, ya que el archivo se envía a la carpeta que se crea al momento de registrar al proveedor en el sistema.

Además, están restringidos los formatos de archivos, únicamente se aceptan PDF y XML, por lo que el intento de algún otro archivo con extensión que no sea la comentada anteriormente provocará un mensaje de error.

Documentos → Documentos enviados



Bienvenido(a): [Ejemplo Uno S.A. de C.V.](#)

Inicio Manual Contacto Documentos Cerrar Sesión

"Ejemplo Uno S.A. de C.V."

SE LISTAN LOS DOCUMENTOS ENVIADOS A LA FECHA: **24-09-2017**

DOCUMENTO	FECHA	CUENTAS POR PAGAR	CONTABILIDAD	RESULTADO DE LA REVISIÓN
FEJEM001.pdf	24-09-2017 17:03:27	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Se realizará el pago el 25/09/2017.
FEJEM001.xml	24-09-2017 17:03:34	<input type="checkbox"/>	<input type="checkbox"/>	
SOPORTE.pdf	24-09-2017 17:06:24	<input type="checkbox"/>	<input type="checkbox"/>	¡Documento Invalidado! Ponte en contacto con el

REGRESAR

[Página FES Cuautitlán](#) / [Página de Software Libre](#)

Figura 35. Lista de los documentos enviados por el proveedor.

Al igual que el personal de la Empresa Automotriz S.A. de C.V., el proveedor también puede consultar los documentos que ha enviado al sistema, así como también el conocer en qué fase se encuentran, es decir, qué área es la que ya validó cual documento, por ejemplo, en esta pantalla, cuando los tres documentos sean validados correctamente, el personal de *Cuentas Por Pagar* asignará la fecha de pago correspondiente en la cual el área de *Tesorería* se encargará de efectuar el pago.

Sin embargo, en esta sección también se ven reflejados los documentos invalidados, validados y por validar, etc., de la misma manera se cuenta con las secciones correspondientes.

Documentos → Validados

The screenshot shows a web browser window with the URL `localhost/proveedores/view/proveedor.phtml?p=validados`. The page header features the logo for 'EMPRESA AUTOMOTRIZ S.A. de C.V.' and a navigation menu with links for 'Inicio', 'Manual', 'Contacto', 'Documentos', and 'Cerrar Sesión'. The main content area displays the title '"Ejemplo Uno S.A. de C.V."' and a subtitle 'SE LISTAN LOS DOCUMENTOS VALIDADOS A LA FECHA: 24-09-2017'. Below this is a table with the following data:

DOCUMENTO	FECHA	PDF	XML	RESULTADO DE LA REVISIÓN
FEJEM001.pdf	24-09-2017 17:03:27	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Se realizará el pago el 25/09/2017.

At the bottom of the table area is a 'REGRESAR' button. The footer contains the text 'Página FES Cuautitlán / Página de Software Libre'.

Figura 36. Lista de los documentos validados.

El proveedor puede consultar en esta sección únicamente los documentos validados, así como el resultado pertinente a la revisión de cada documento.

Documentos → Invalidados

DOCUMENTO	FECHA DE CANCELACIÓN	RESULTADO
SOPORTE.pdf	24-09-2017 17:28:07	¡Documento Invalidado! Ponte en contacto con el personal de Cuentas por Pagar

Figura 37. Lista de los documentos invalidados.

En esta sección el proveedor puede consultar los documentos invalidados, se muestra un enlace al documento para que consulte de cual se trata, de la misma manera la fecha y hora en la que se invalidó el documento y un texto que le indica que tiene que ponerse en contacto con el personal de la Empresa Automotriz S.A. de C.V.

Acceso Denegado

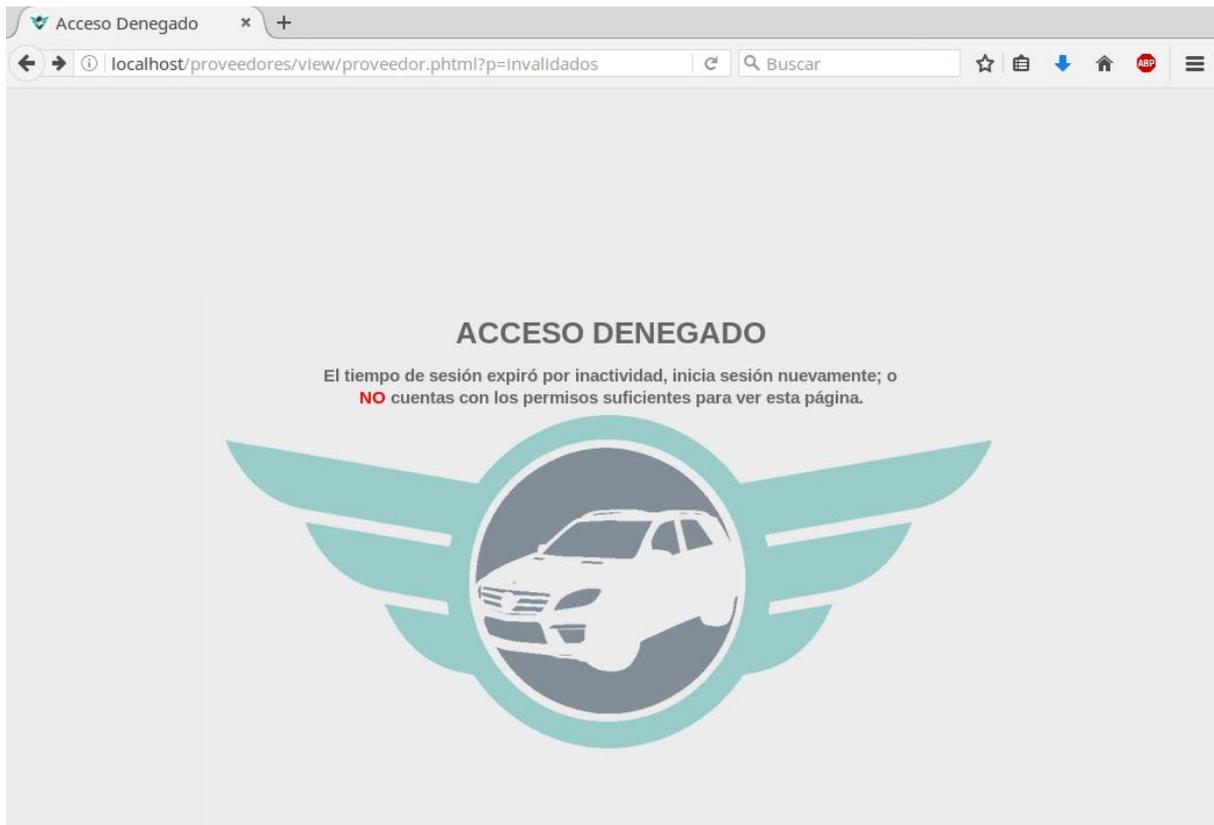


Figura 38. Acceso Denegado.

Esta pantalla se muestra en los siguientes casos:

1. El usuario no cuenta con los permisos para ver la página a la cual quiere acceder;
2. El usuario cerró sesión e intentó regresar a la página anterior; y
3. Cuando no hay interacción con el sistema durante un tiempo determinado, éste cierra la sesión por seguridad.

5.3.3. Implementación

Una vez finalizado el sistema web en un ambiente de pruebas, se prosiguió con la implementación de dicho sistema en los servidores de la Empresa Automotriz S.A. de C.V., para dicha implementación la colaboración de las siguientes áreas fue de suma importancia:

1. Área de *Sistemas*;
2. Área de *Cuentas Por Pagar*; y
3. Área de *Contabilidad*.

El área de *Sistemas* se encargó de proporcionar los accesos correspondientes a su servidor para la carga del sistema. La tarea de la implementación tomó alrededor de una semana ya que los parámetros de seguridad de la empresa impedían la carga correcta del sistema a su servidor, además, cuestiones técnicas en cuanto al motor de la Base de Datos también generó problema ya que éste tuvo que recibir actualización a una versión más reciente, ya que la versión instalada al momento de querer realizar la integración no permitía la creación de la Base de Datos.

Posteriormente a la carga del sistema y de la base de datos en el sistema y continuando con la fase de la implementación, se continuó con la capacitación sobre el uso del sistema, para lograr esto se realizaron cuentas pertinentes para el personal de *Cuentas Por Pagar* y *Contabilidad*, mismas que fueron validadas por el área de *Sistemas* la cual realizaba un seguimiento en su Base de Datos para el cotejo de la correcta inserción de datos. Y así comenzar el periodo de pruebas el cual consistió en enviar documentos ficticios en los formatos requeridos por el sistema.

Al término de la implementación, se realizaron varias reuniones con personal de alto mando en la Empresa Automotriz S.A. de C.V., para conocimiento de las actividades que fueron elaboradas en ese periodo, así mismo para la obtención de la aprobación correspondiente sobre el uso del sistema y los beneficios que se obtienen con la utilización del mismo.

Conclusiones

Actualmente, el contar con un sistema de información el cual permita gestionar información de personas que forman parte indirecta en las organizaciones (en este caso de los proveedores), es de gran ayuda para la simplificación de los procesos que son de suma importancia para la organización, y es que todos y cada uno de los procesos juegan un papel importante, y que mejor cuando existe de por medio un sistema que ayuda a mejorar dichos procesos, es decir, que cuenta con la capacidad de reducir tiempos, trabajo, administración de documentos, etc.

El desarrollo de este sistema para la Empresa Automotriz S.A. de C.V., logró cumplir con las expectativas, ya que las áreas involucradas en el manejo de éste se vieron completamente beneficiadas, más que nada por el hecho de contar con un sistema que les ayude a controlar este tipo de documentación que día a día se vuelve más requerida, sin embargo, el sistema se encuentra preparado para que se le realicen mejoras y así ofrecer una experiencia grata tanto al personal de la empresa como a los proveedores que también forman parte de este sistema.

Desarrollar este tipo de sistemas deja una experiencia inigualable, sobre todo en la vía de aprendizaje la cual aumenta considerablemente, y es que el desarrollar no solo es enfocarse en el problema de una persona, sino realizar un análisis minucioso de la necesidad general para así obtener el resultado deseado, involucrando a todas las personas que se beneficiarán del sistema que se llegue a desarrollar, tal es el caso de la Empresa Automotriz S.A. de C.V., el poder controlar este tipo de documentación les es de gran ayuda, ya que agilizan los tiempos que invierten, pero no solo ellos, también los proveedores salen beneficiados.

Personalmente me queda una gran satisfacción con la realización de este proyecto, porque sé que pude contribuir a que una empresa de tal magnitud como lo es ésta en el giro automotriz, pudo mejorar algunos de sus procesos, porque a futuro el beneficio es notable, sobre todo en el tiempo que se invierte tanto por el personal de la empresa como el de los proveedores, además del tiempo en el que las revisiones son ejecutadas. Por lo anterior, es recomendable que este sistema sea actualizado conforme al avance de las tecnologías, para que el mismo sea duradero y más robusto.

El papel que juega el Licenciado en Informática se ha vuelto determinante para el impulso de las empresas, ya que éste apoya de manera significativa en las nuevas tecnologías que se van presentando con el avance del tiempo y de las mismas.

Finalmente, considero que como desarrollo de este trabajo se lograron los objetivos que se plantearon, es decir, el utilizar la infraestructura LAMP en donde la empresa no tuvo que realizar ningún tipo de inversión; utilizar BPMN para una mejor descripción de los procesos con los que la empresa ya contaba y extender el panorama de cómo serían estos procesos con la utilización de la aplicación desarrollada, además la utilización de MVC como modelo de desarrollo ayudó a que la aplicación tenga un ciclo de vida más extenso, obteniendo una aplicación robusta.

Referencias

1. Torres, M. A. (2015). *Desarrollo de aplicaciones web con PHP y MySQL*. Lima, Perú: Macro.
2. Spona, H. (2010). *Programación de bases de datos con MySQL y PHP*. México: Alfaomega.
3. Kendall, K. E., Kendall, J. E., Kenneth, E. K., Julie, E. K., Pungitore, J. L. L. P., Nino, P., ... & Gómez Vieites, A. (2011). *Análisis y diseño de sistemas*. México: Pearson Educación.
4. Negrino, T. (2007). *JavaScript & Ajax para diseño web*. Madrid; México: Pearson Educación.
5. Fernández, Y. & Díaz Y. (2012). *Patrón Modelo-Vista-Controlador*. TELEM@TICA, 11, pp.47-57.
6. Bahit, E. (2011). *POO y MVC en PHP*. Argentina: Open Libra.
7. Remón, T. & Ángel, M. (2014). *Diseño web con: HTML5 y CSS3*. Lima, Perú: Empresa Editorial Macro.
8. Montañez, F. (2012). *Ofimática y proceso de la información: grado superior*. España: McGraw-Hill.
9. Rosebrock, E., Hughes, J. & Filson, E. (2004). *Setting Up LAMP: Getting Linux, Apache, MySQL, and PHP Working Together*. 1st ed. [ebook] San Francisco, CA · Londres: SYBEX, pp.1-100. Disponible en: <http://ebooks.cawok.pro/Sybex.Setting.Up.LAMP.Getting.Linux.Apache.MySQL.and.PHP.Working.Together.Jul.2004.eBook-DDU.pdf> [Consultado el 25 Feb. 2017].
10. Stallman, R. (2004). *Software libre para una sociedad libre*. Madrid: Traficantes de sueños.
11. Sommerville, I. (2005). *Ingeniería del software. Séptima edición*. Madrid: PEARSON EDUCACIÓN, S.A.