

**INSTITUTO SUPERIOR TECNOLÓGICO**  
**“SAN GABRIEL”**



**ESPECIALIDAD:** INFORMÁTICA MENCIÓN ANÁLISIS DE SISTEMAS

**TEMA:**

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA EL CONTROL DE TICKETS DEL CENTRO DE SERVICIO TÉCNICO DE LA EMPRESA @PC, DESARROLLADA EN PHP ORIENTADA A OBJETOS Y CON UN GESTOR DE BASE DE DATOS MYSQL.

**TRABAJO DE INVESTIGACIÓN**

**PREVIA A LA OBTENCIÓN DEL TÍTULO:**

“TECNÓLOGA EN INFORMÁTICA MENCIÓN ANÁLISIS DE SISTEMAS”

**AUTORA:**

María Fernanda Manzano Calderón

**TUTOR:**

WILLIAM ADRIANO

**RIOBAMBA – ECUADOR**

**2020**

## **CERTIFICACIÓN**

Certifico que la Sra. **MARÍA FERNANDA MANZANO CALDERÓN**, con N° de Cédula 0604970673 ha elaborado bajo mi Asesoría el Proyecto de Investigación titulado:

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA EL CONTROL DE TICKETS DEL CENTRO DE SERVICIO TÉCNICO DE LA EMPRESA @PC, DESARROLLADA EN PHP ORIENTADA A OBJETOS Y CON UN GESTOR DE BASE DE DATOS MYSQL.**

Por lo tanto, autorizo la presentación para la calificación respectiva.

---

**Ing. William Adriano**

**DOCENTE DEL ITSGA**

“El presente Proyecto de Investigación constituye un requisito previo para la obtención del Título de **Tecnólogo en Informática Mención Análisis de Sistema**”



“Yo, **MARÍA FERNANDA MANZANO CALDERON** con N° de Cédula 0604970673, declaro que la investigación es absolutamente original, autentica, personal y los resultados y conclusiones a los que se han llegado es de mi absoluta responsabilidad.”

---

MARÍA FERNANDA MANZANO CALDERÓN

**INSTITUTO TECNOLÓGICO SUPERIOR**

**“SAN GABRIEL”**

**ESPECIALIDAD INFORMÁTICA**

**TESIS DE GRADO**

**PREVIA A LA OBTENCION DEL TÍTULO DE:**

**TECNÓLOGA EN INFORMÁTICA MENCIÓN ANÁLISIS DE  
SISTEMAS**

**TEMA:**

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA EL CONTROL DE  
TICKETS DEL CENTRO DE SERVICIO TÉCNICO DE LA EMPRESA @PC,  
DESARROLLADA EN PHP ORIENTADA A OBJETOS Y CON UN GESTOR DE  
BASE DE DATOS MYSQL.

**APROBADO:**

**ASESOR DE TESIS DE GRADO**

.....

**PRESIDENTE DEL TRIBUNAL**

.....

**MIEMBRO DEL TRIBUNAL**

.....

**MIEMBRO DEL TRIBUNAL**

.....

## **FIRMAS DE RESPONSABILIDAD**

<b>NOMBRES</b>	<b>FECHA</b>	<b>FIRMA</b>
<b>DIRECTOR DE TESIS</b>	.....	.....
<b>PRESIDENTE DEL TRIBUNAL</b>	.....	.....
<b>MIEMBRO DEL TRIBUNAL</b>	.....	.....
<b>MIEMBRO DEL TRIBUNAL</b>	.....	.....
<b>NOTA FINAL DE LA TESIS.....</b>		

## **DEDICATORIA**

Dedico esta tesis a mi DIOS que me dio la oportunidad de vivir y de regalarme una maravillosa familia.

Con mucho cariño a mis padres quienes me dieron la vida y han estado conmigo en todo momento. A mi esposo por ser el apoyo incondicional. A mi querido hijo por ser el motor que me motiva cada día. A mis hermanos y sobrinos por compartir momentos significativos conmigo. A todos ellos se los agradezco desde el fondo de mi alma. Para todos ellos hago esta dedicatoria.

*Fernanda*

## **AGRADECIMIENTO**

Muy agradecida a mi Dios, por haberme dado la inteligencia y fortaleza para culminar esta etapa de mi vida.

Gracias a mi familia por apoyarme en cada decisión y proyecto, a mis padres por ser los principales promotores de mis sueños, gracias a ellos por cada día confiar y creer en mí y en mis expectativas.

Gracias a mi esposo por entenderme en todo momento, fue un apoyo incondicional para culminar esta etapa.

Les agradezco, y hago presente mi gran afecto hacia ustedes, mi hermosa familia.

*Fernanda*

## ABREVIATURAS

<b>HTML:</b>	HyperText Markup Language
<b>WWW:</b>	World Wide Web
<b>MVC:</b>	Model View Controller
<b>XAMPP:</b>	Apache Mariadb Php Perl
<b>PHP:</b>	Hypertext Pre-processor
<b>POO:</b>	Programación Orientada a Objetos
<b>HTTP:</b>	Hypertext Transfer Protocol
<b>GUI:</b>	Graphical User Interface
<b>DBMS:</b>	Database management system
<b>URL:</b>	Localizador de Recursos Uniforme
<b>QR:</b>	Quick Response



## GLOSARIO DE TÉRMINOS

**XAMPP:** Es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl.

(<https://www.apachefriends.org/es/index.html>)

**PHP:** Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

(<https://www.php.net/manual/es/intro-what-is.php>)

**APACHE:** Es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es un servidor multiplataforma, gratuito, muy robusto y que destaca por su seguridad y rendimiento.

(<http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>)

**SERVIDOR WEB:** Es un ordenador de gran potencia que se encarga de “prestar el servicio” de transmitir la información pedida por sus clientes.

(<https://rockcontent.com/es/blog/que-es-un-servidor>)

**MYSQL:** Es un sistema de gestión de bases de datos que cuenta con una doble licencia. De código abierto y una versión comercial gestionada por la compañía Oracle.

(<https://openwebinars.net/blog/que-es-mysql/>)

**WEB HOSTING:** Es un servicio en línea que te permite publicar un sitio o aplicación web en Internet

(<https://www.hostinger.es/tutoriales/que-es-un-hosting>)

## ÍNDICE GENERAL

ABREVIATURAS.....	8
GLOSARIO DE TÉRMINOS .....	9
CAPÍTULO I .....	20
MARCO REFERENCIAL .....	20
1.1 ANTECEDENTES.....	21
1.2 DEFINICIÓN DEL PROBLEMA .....	22
1.3 JUSTIFICACIÓN.....	22
1.4 OBJETIVOS .....	24
1.4.1 OBJETIVO GENERAL .....	24
1.4.2 OBJETIVOS ESPECÍFICOS .....	24
CAPÍTULO II.....	25
MARCO TEÓRICO.....	25
2.1 SITIO WEB.....	26
2.1.1 Definición de un Sitio Web .....	26
2.1.2 Ventajas de un Sitio Web .....	26
2.1.3 Ejemplos de un Sitio Web .....	26
2.1.4 Componentes Básicos de un Sitio Web .....	27
2.2 APLICACIÓN WEB .....	29
2.2.1 Definición de un Aplicación Web .....	29
2.2.2 Usos comunes de las aplicaciones Web.....	29
2.2.3 Funcionamiento de una Aplicación Web .....	30
2.3 SERVIDOR WEB .....	33
2.3.1 Definición de un Servidor Web .....	33
2.3.2 Características de un Servidor Web.....	34
2.3.3 Funciones de un Servidor Web.....	34
2.3.4 Servicios.....	35
2.4 LENGUAJE PHP .....	36
2.4.1 Funciones de PHP .....	37
2.4.2 Formularios .....	40

2.4.3	Sesiones.....	41
2.5	PROGRAMACIÓN ORIENTADA A OBJETOS EN PHP.....	42
2.5.1	Características de la Programación Orientada a Objetos .....	43
2.5.2	Elemento de la Programación Orientada a Objetos .....	45
2.6	PATRÓN MVC.....	49
2.6.1	Importancia del MVC.....	50
2.6.2	Definición del Patrón MVC.....	52
2.6.3	Arquitectura de Aplicaciones MVC .....	54
2.6.4	Lógica de Negocio / Lógica de Aplicación.....	55
2.6.5	Ventajas de utilizar MVC .....	56
2.6.6	¿Cómo funciona el MVC?.....	57
2.6.7	¿Cómo implementar el MVC en PHP?.....	58
2.6.8	Ejemplo de la funcionalidad de un programa MVC en PHP y Mysql.....	58
2.6.8.1	Implementación de la base de datos Mysql .....	58
2.6.8.2	Estructura de archivos del Proyecto Web .....	59
2.7	BASE DE DATOS EN MYSQL .....	60
2.7.1	Componentes de los SGBD .....	60
2.7.2	MYSQL.....	64
2.7.3	XAMPP.....	69
2.8	CÓDIGO QR.....	70
2.8.1	Definición de Código QR .....	70
2.8.2	¿Para qué sirve el Código QR?.....	70
2.8.3	¿Qué puede contener un Código QR?.....	71
2.8.4	¿Cómo leer un Código QR?.....	71
2.8.5	¿Cómo generar códigos QR utilizando PHP? .....	71
CAPÍTULO III.....		73
ANÁLISIS Y DISEÑO .....		73
3.1	RECOPIACIÓN DE INFORMACIÓN.....	74
3.2	ANÁLISIS .....	75
3.2.1	ESTUDIO DE FACTIBILIDAD .....	75
3.2.2	ANÁLISIS DE LOS REQUERIMIENTOS .....	77
3.2.3	CASOS DE USO.....	80

3.3	DISEÑO .....	82
3.3.1	Diseño Conceptual .....	82
3.3.2	Modelo Relacional .....	83
3.3.3	Diccionario de Datos .....	83
3.3.4	Diseño de Interfaces .....	89
CAPÍTULO IV.....		93
IMPLEMENTACIÓN DE LA APLICACIÓN WEB .....		93
4.1	CONFIGURACIÓN DE LAS HERRAMIENTAS DE DESARROLLO .....	94
4.2	ARQUITECTURA DEL SISTEMA .....	95
4.3	IMPLEMENTACIÓN DE LA APLICACIÓN.....	95
4.3.1	Definición de Módulos de Implementación.....	95
4.3.2	Desarrollo de la Aplicación .....	96
4.3.3	Implantación del Sistema.....	97
4.4	PRUEBAS AL SISTEMA.....	97
4.5	CAPACITACIÓN AL PERSONAL.....	98
4.6	MANTENIMIENTO .....	99
CAPÍTULO V.....		100
CONCLUSIONES Y RECOMENDACIONES .....		100
5.1	CONCLUSIONES .....	101
5.2	RECOMENDACIONES .....	102
BIBLIOGRAFÍA .....		103
ANEXOS .....		105

## ÍNDICE DE TABLAS

Tabla 1: Definición de tipos de variables en PHP .....	40
Tabla 2: Formato de un Formulario.....	41
Tabla 3: Sesiones .....	42
Tabla 4: Factibilidad Económica.....	77
Tabla 5: Requerimientos Funcionales .....	78
Tabla 6: Requerimientos No Funcionales.....	79
Tabla 7: Tabla acceso .....	84
Tabla 8: Tabla Cliente .....	85
Tabla 9: Tabla equipo.....	85
Tabla 10: Tabla estado .....	86
Tabla 11: Tabla historial .....	86
Tabla 12: Tabla reparacion.....	87
Tabla 13: Tabla tipo_usuario.....	87
Tabla 14: Tabla usuario.....	88

## ÍNDICE DE FIGURAS

Figura 1: Página Web Estática .....	31
Figura 2: Procesamiento de Página Web Estática.....	32
Figura 3: Procesamiento de Página Web Dinámica.....	33
Figura 4: Funcionamiento de PHP .....	37
Figura 5: Etiquetas soportadas por PHP .....	38
Figura 6: PHP embebido en código HTML.....	39
Figura 7: Métodos de una Clase .....	47
Figura 8: Representación de un Objeto .....	47
Figura 9: Objeto como Instancia de un Clase .....	47
Figura 10: Representación de un Método.....	48
Figura 11: Métodos Mágicos.....	48
Figura 12: Propiedades y Atributos.....	48
Figura 13: Herencia de Clases .....	49
Figura 14: Polimorfismo .....	49
Figura 15: Arquitectura MVC desde el punto de vista del Usuario .....	52
Figura 16: Arquitectura MVC .....	54
Figura 17: Esquema funcionamiento del MVC en PHP .....	58
Figura 18: Esctructura del MVC en PHP .....	58
Figura 19: Esquema de la Base de Dattos .....	59
Figura 20: Arquitectura de Myssql.....	65
Figura 21: Funcionamiento de Mysql.....	68
Figura 22: Xampp .....	69
Figura 23: Código QR.....	70
Figura 24: Generar Código QR con PHP .....	72

Figura 25: Resultado del Código QR con PHP.....	72
Figura 26: Organigrama de la Empresa @PC.....	75
Figura 27: Diagrama de Caso de Uso Administrador .....	80
Figura 28: Diagrama de Caso de Uso Secretaria .....	81
Figura 29: Diagrama de Caso de Uso Técnico .....	81
Figura 30: Diagrama de Caso de Uso Cliente.....	82
Figura 31: Diagrama Conceptual.....	82
Figura 32: Diseño Conceptual.....	83
Figura 33: Inicio de Sesión.....	89
Figura 34: Pantalla Principal .....	90
Figura 35: Pantalla de Administración de Usuarios.....	90
Figura 36: Pantalla de Administración de Clientes.....	91
Figura 37: Pantalla de Administración de Equipos.....	91
Figura 38: Pantalla de Reparación de Tickets .....	92
Figura 39: Pantalla de Visual Studio Code.....	94
Figura 40: Pantalla de Inicio de Xampp .....	94
Figura 41: Pantalla de Arquitectura del Sistema.....	95
Figura 42: Pantalla de Módulos de la Aplicación.....	96
Figura 43: Pantalla Principal de la Aplicación .....	97

## INTRODUCCIÓN

En la actualidad, la tecnología e informática son muy importantes, ya que estos medios nos permiten realizar actividades necesarias para la vida cotidiana, igual suerte corre el campo empresarial el cual implementa nuevas formas de enseñanza y facilidad en la realización de trabajos. Hoy en día todo lo existente constituye un proceso tecnológico e informático creado para satisfacer las necesidades de la humanidad. Es por esta razón que la empresa @PC ubicada en ciudad de Riobamba provincia de Chimborazo, dedicada al mantenimiento y reparación de equipos informáticos, actualmente cuenta con un proceso de recepción, reparación y entrega de equipos de manera manual, que son llevados en Kardex , información que corre el riesgo de perderse o de confundirse.

Ante lo expuesto, se pretende desarrollar un aplicativo web para el Centro de Servicio Técnico de la Empresa @PC, tendiente a controlar el registro de los equipos que ingresen a ser reparados, y mediante un ticket que el sistema genera dará seguimiento al mismo, por consiguiente, la entrega será controlada por un código QR que deberá ser verificado antes de entregar al cliente, facilitando la entrega de equipos al encontrarse todo organizado.

Lo que motivó la realización del trabajo de investigación fue evidenciar la desorganización del manejo y entrega de los quipos reparados en la empresa @PC observado durante la ejecución de prácticas, y al poseer los conocimientos técnicos por ser estudiante de la Carrera de Informática Mención Análisis de Sistemas del Instituto Superior Tecnológico “San Gabriel”, se considera oportuno aplicar los mismos y contribuir al desarrollo de la empresa.

El objetivo general de la investigación es analizar, diseñar e implementar un sistema del control de tickets el cual automatice el control de todos los eventos ocurridos en cada uno de los equipos que ingresan al Centro de Servicio Técnico de la Empresa “@PC”.

Finalmente se indica que, el informe de investigación realizado consta de 5 capítulos los cuales se describen a continuación:

El capítulo I constituido por el Marco Referencial, describe la problemática con la cual surge el tema de investigación, los objetivos del proyecto, así como la justificación, los cuales



facilitaron la comprensión de las necesidades, requerimientos de la institución, al mismo tiempo que constituyen el direccionamiento de la investigación.

El capítulo II Marco Teórico, detalla los conceptos investigados que son necesario para la implementación de la aplicación. Entre ellos, qué es un Servidor Web, qué es un Aplicativo Web, el Lenguaje PHP Orientado a Objetos, el gestor de Base de Datos Mysql y el Código QR, herramientas que facilitaron al diseño del sitio web, así como también al almacenamiento de la información, seguimiento y lectura de los códigos QR.

El capítulo III Análisis y Diseño, analiza técnicas, factibilidad, requerimientos funcionales, no funcionales y diseño de la arquitectura de la base de datos, en donde se describe como se recolectaron los datos para su realización, factibilidad, casos de uso, y la delimitación de la interfaz web.

El capítulo IV Implementación de la Aplicación, se detalla todas las herramientas utilizadas para la implementación del sistema, arquitectura del sistema, pruebas, capacitación al personal del centro de servicio técnico de la empresa y mantenimiento del aplicativo, de esta forma la empresa y los usuarios del sistema pueden verificar que el sistema es totalmente funcional y que no posee errores.

Y, el capítulo V constituido por las conclusiones, que son los resultados obtenidos durante el desarrollo de la investigación; las recomendaciones, que direccionan la propuesta de futuros trabajos de investigación, la bibliografía y anexos constituyen también elementos que complementan este capítulo.

## RESUMEN

El trabajo de investigación titulado Diseño e Implementación de un sitio web para el control de tickets del Centro de Servicio Técnico de la Empresa “@PC” de la ciudad de Riobamba, programado en lenguaje de programación PHP Orientado a Objetos, con gestor de base de datos Mysql, se desarrolló durante el periodo 2020. La metodología empleada inicia con el análisis de todos los requerimientos funcionales y requerimientos no funcionales, para lo cual fue necesario la aplicación de técnicas de investigación de campo y bibliográfica, las cuales permitieron identificar las falencias existentes en el centro de servicio técnico de @PC. Seguidamente se diseñan los diagramas entidad relación de la base de datos, los cuales permitieron identificar a los clientes y equipos; posteriormente se determinó el diseño funcional con los gráficos de caso de uso que permitieron entender el funcionamiento del sistema por parte de los programadores. En la fase de programación con la ayuda de IDE Visual Studio Code como editor de código se pudo insertar el lenguaje de programación PHP orientado a objetos, con lo que se pudo generar los tickets del sistema; con el lenguaje HTML y hojas de estilos CSS se pudo obtener una interfaz amigable e interactiva para el usuario final; mientras que el sistema gestor de base de datos MySQL permitió que los datos de clientes y equipos reparados, así como los tickets generados sean guardados de una forma segura. Finalmente se realiza la fase de pruebas e implementación, en donde se evidencio que el sistema funciona en su totalidad, al ser analizado por las pruebas de caja blanca y caja negra sin obtener errores de funcionamiento o visualización. Los resultados que se obtuvieron con el diseño e implementación del sitio web en el Centro de Servicio Técnico de la empresa @PC fueron: la obtención de formularios para la creación de usuarios que puedan acceder al sistema como la recepcionista y el técnico de reparación en donde se almacenarán los datos de los equipos ingresados para su reparación, el ingreso de los datos personales de los propietarios de los equipos, así como también la generación de tickets de reparación en donde consta también los datos del equipo y datos del cliente, a lo que se suma el código QR en donde se muestra la actualización del historial de reparación del equipo que es enviado al correo del cliente, el cual puede ser visualizado y escaneado en algún aplicativo móvil, llegando así la información de manera oportuna y precisa al cliente.

## SUMMARY

The research work entitled Design and Implementation of a website for the control of tickets of the Technical Service Center of the Company "@PC" of the city of Riobamba, programmed in Object Oriented PHP programming language, with database manager Mysql data, was developed during the 2020 period. The methodology used begins with the analysis of all functional requirements and non-functional requirements, for which it was necessary to apply field and bibliographic research techniques, which allowed identifying the existing shortcomings at the @PC technical service center. Next, the entity-relationship diagrams of the database are designed, which allowed the identification of clients and teams; Later, the functional design was determined with the use case graphics that allowed the programmers to understand the operation of the system. In the programming phase, with the help of IDE Visual Studio Code as a code editor, the object-oriented programming language PHP could be inserted, with which system tickets could be generated; With the HTML language and CSS style sheets, a friendly and interactive interface for the end user was obtained; while the MySQL database management system allowed the customer data and repaired equipment, as well as the generated tickets, to be stored in a secure way. Finally, the testing and implementation phase is carried out, where it was evidenced that the system works in its entirety, when analyzed by the white box and black box tests without obtaining operating or visualization errors. The results that were obtained with the design and implementation of the website in the Technical Service Center of the company @PC were: obtaining forms for the creation of users who can access the system such as the receptionist and the repair technician where The data of the equipment entered for repair will be stored, the entry of the personal data of the owners of the equipment, as well as the generation of repair tickets which also contains the equipment data and customer data, to which The QR code is added showing the update of the repair history of the equipment that is sent to the customer's email, which can be viewed and scanned in a mobile application, thus reaching the information in a timely and accurate manner to the customer.

**CAPÍTULO I**  
**MARCO REFERENCIAL**

## 1.1 ANTECEDENTES

La empresa @PC es una empresa Riobambeña que se desempeña en el área tecnológica, inicia en el año 2001 dedicada a la distribución y comercializa de componentes, partes, piezas, equipos de computación, equipos electrónicos y tecnológicos. Además, provee servicios de mantenimiento reparación, diagnóstico de computadoras y asesoría de sistemas.

Actualmente se encuentra ubicada en las calles Rocafuerte entre 10 de Agosto y Primera Constituyente. La empresa cuenta con más de 19 años de vasta experiencia en el área de mercadeo y ventas de productos y servicios mediante planes estratégicos y de negocios bien elaborados. Los mismos que permitirán posicionar a la empresa en el mercado ecuatoriano y así tener mayor participación de mercado. Consideramos que el equipo humano de @PC representa una de las bases fundamentales, debido a la experiencia y conocimiento que poseen de sus productos y de las tendencias del mercado local. Lo que permite generar confianza hacia nuestros clientes actuales y futuros. Preocupándonos siempre por el mejoramiento continuo y capacitación de nuestro equipo.

En la larga trayectoria que tiene la empresa “@PC”, no cuenta con una aplicación web donde se pueda se realicen de forma automática la generación de un ticket para el mantenimiento y reparación de equipos que ingresan a la empresa.

Lo cual dificulta de tal manera el tener historiales de los equipos que han sido reparados, debido a que deben atender un gran número de equipos, no permite llevar un control de todos los sucesos que se presentan en cada uno de los equipos que ingresan a la empresa.

Para resolver este problema se plantea el desarrollo del aplicativo web enfocado principalmente a mostrar información permanente acerca del estado del mantenimiento y reparación de los equipos que ingresan a la empresa, de esta manera brindando un mejor servicio al usuario, además la aplicación podrá ayudar a los clientes de la empresa a mantenerse informados sobre el estado de su equipo, pudiendo ingresar al sistema desde cualquier parte donde tengan acceso a Internet.

## **1.2 DEFINICIÓN DEL PROBLEMA**

La empresa @PC en la actualidad no cuenta con un aplicativo web para el proceso de mantenimiento y reparación de computadoras, que proporcione la información necesaria de los estados de los equipos que se están reparando en el momento que sea conveniente y de forma oportuna.

Actualmente la empresa realiza de forma manual la generación de una orden de mantenimiento y reparación de computadoras, haciendo que exista inconsistencia de información en el momento de realizar el proceso de mantenimiento del equipo, como también falta de información del estado en el que se encuentra su computadora para el usuario.

Luego de un breve análisis de los requerimientos y necesidades relacionadas con ésta situación, se obtuvo lo siguiente:

- Falta de control sobre quién realizó o atendió un soporte técnico.
- Falta de reportes que sirvan como indicadores para los clientes, con la intención de demostrar los esfuerzos realizados por el personal de atención.
- No existe forma de saber si un ticket se encuentra sin atención, es necesario saber si el ticket se encuentra con un estado abierto o cerrado.
- La creación de reportes es manual, lo que ocasiona desorganización y mayor esfuerzo.
- Falta de buena atención a todos los clientes de la empresa.

## **1.3 JUSTIFICACIÓN**

El presente proyecto de investigación pretende realizar la investigación de la programación orientada a objetos en el lenguaje de programación PHP, el cual servirá como aporte académico aplicando los conocimientos adquiridos en el desarrollo de un aplicativo web el cual será implementado en el Departamento Técnico de la Empresa “@PC”.

El desarrollo del proyecto permitirá ayudar en el control a las personas que trabajan en el departamento técnico de la empresa en el proceso de ingreso de los equipos para el mantenimiento y reparación, debido que actualmente se lo realizan manualmente, en la

actualidad no existe un software que ayude a la automatización de dicho proceso, lo cual dificulta de tal manera el tener historiales de los equipos que han sido reparados, debido a que deben atender un gran número de equipos, esto no permite llevar un control de todos los sucesos que se presentan en cada uno de los equipos que ingresan a la empresa.

A consecuencia de estos problemas, el departamento de técnico de la empresa se ve en la necesidad de desarrollar un sistema y poder resolver la problemática planteada. Es precisamente en estos casos cuando se habla del sistema de automatización de procesos, específicamente del sistema de seguimiento de tickets, el cual permitirá establecer los procedimientos para solventar las deficiencias encontradas en el análisis previo.

Este proyecto se puede realizar ya que el gerente propietario de la empresa “@PC” nos ha facilitado toda la información necesaria para el desarrollo del proyecto, así permitiendo sea ejecutado sin contratiempos, además nos ha permitido la toma de datos y análisis preliminares.

Este trabajo se lo realizará con herramientas open source y será un software hecho a la medida para la empresa “@PC” según las necesidades de la misma. Para el desarrollo se va utilizar una arquitectura MVC (Modelo Vista Controlador) con el lenguaje de programación PHP Orientado a Objetos y el gestor de base de datos MySQL.

Las personas beneficiarias directas con la implementación del aplicativo web será la empresa “@PC”, ya que mejorará notablemente el control en el que se encuentre cada equipo cuando ingresa a departamento técnico, mientras que las personas beneficiarias indirectas serán los usuarios ya que tendrán un informe constante en el estado que su equipo se encuentra. Toda la información será obtenida de forma fácil, confiable y rápida.

## **1.4 OBJETIVOS**

### **1.4.1 OBJETIVO GENERAL**

- Analizar, diseñar e implementar un Sistema del Control de Tickets el cual automatice el control de todos los eventos ocurridos en cada uno de los equipos que ingresan en el Departamento Técnico de la empresa @PC.

### **1.4.2 OBJETIVOS ESPECÍFICOS**

- Investigar la programación PHP orientada a objetos y el gestor de base de datos mysql para el desarrollo del aplicativo.
- Automatizar el procedimiento que realiza el departamento de Soporte Técnico para el control de tickets de los equipos que ingresan a reparación.
- Desarrollar una aplicación web robusta, escalable y segura.



**CAPÍTULO II**  
**MARCO TEÓRICO**

## **2.1 SITIO WEB**

### **2.1.1 Definición de un Sitio Web**

Conjunto organizado y coherente de páginas Web que tiene como función ofrecer, informar, publicitar o vender contenidos, productos y servicios al resto del mundo. Para que un sitio Web pueda ser visitado por otras personas es necesario que se encuentre alojado en un servidor. Se trata de una computadora conectada a la World Wide Web con espacio en disco y conectividad suficiente para albergar sitios y servirlos a resto de la comunidad de usuarios de Internet a través de direcciones IP o nombres de dominio (ecured.cu, 2020).

### **2.1.2 Ventajas de un Sitio Web**

Un sitio Web de calidad le permitirá mejorar su imagen profesional, demostrando por encima de todo, el conocimiento y la aceptación de las nuevas tecnologías. Además, cada vez son más los miles de usuarios que acceden a Internet buscando información, un producto o servicio. Por eso, no tener una Web supone una gran desventaja competitiva con respecto a otras empresas que sí la tienen.

Internet se ha convertido en la forma más eficiente de transmitir información. Permite el contacto directo entre su empresa, cliente y proveedor de una manera más rápida y económica. Toda la información que usted incluya en su sitio Web estará disponible las 24 horas del día, 7 días a la semana y 365 días al año (desarrolloweb.com, 2020).

### **2.1.3 Ejemplos de un Sitio Web**

Es importante establecer que en Internet encontramos una gran variedad de tipos de sitios web que suelen diferenciarse fundamentalmente por la clase de contenido que ofrecen o por el servicio que brindan a cualquiera de las personas que se encuentran navegando por la Red.

Así, por ejemplo, tendríamos que destacar los conocidos blogs. Se tratan de una especie de bitácora en la que una persona en particular realiza post relativos a sus gustos, a sus aficiones o a sus conocimientos en una relativa manera. De la misma manera, están aquellos que son utilizados simplemente para contar, en forma de diario, lo que les sucede día a día. Para poder crear a aquellos existen diversos tipos de plataformas gratuitas tales como Blogger o WordPress. De la misma forma, tampoco podemos pasar por alto la existencia de los sitios

web de empresas. Como su propio nombre indica, son aquellos lugares en la Red que son utilizados por negocios de diversa índole para dar a conocer sus productos o sus servicios. No obstante, también los emplean para aumentar su cartera de clientes y para mantener una relación más directa con sus clientes.

Los sitios personales son, por su parte, aquellos que mantienen los individuos en distintas plataformas y que utilizan para mostrar información sobre sí mismos, ya sea a nivel escrito o bien mediante fotografías. Ejemplos de ellos son los sitios que tienen muchos usuarios en redes sociales tales como Facebook, Tuenti o Twitter. Los sitios de subastas, los sitios de noticias o los sitios de descargas son otros de los tipos más conocidos.

Los sitios web incluyen documentos HTML, fotografías, sonidos, vídeos, animaciones Flash y otro tipo de contenidos que pueden compartirse en línea. La URL raíz del sitio web se conoce como portada o homepage. Lo más habitual es que esta portada facilite el acceso a todas las páginas del sitio mediante hipervínculos (también conocidos como enlaces o links).

La mayor parte de los sitios web ofrecen sus contenidos de manera gratuita. Otros, en cambio, cobran una suscripción y exigen una contraseña para su acceso. Ese es el caso de muchos sitios pornográficos, de información bursátil o de juegos online, entre otros servicios.

Cabe destacar que para acceder a un sitio web se necesita contar con un navegador en la computadora. Este es un programa informático que posibilita la visualización de la información contenida en una página web (Merino, 2013).

#### **2.1.4 Componentes Básicos de un Sitio Web**

- **Hosting o Alojamiento:** Es el lugar físico donde van estar tus datos y los sistemas necesarios para que todo funcione. Básicamente es una empresa que tiene miles de servidores (computadoras) con discos rígidos y buena conexión a Internet. Estas empresas se encargan de (B, 2019):
  - Guardar tus datos
  - Protegerlos contra ataques
  - Hacer una copia de respaldo de tus datos
  - Mantener y reparar sistemas y equipos

- Asegurar el funcionamiento constante de los servidores
  - Procurar la accesibilidad a los datos en todo momento
  - Brindar asistencia técnica a sus clientes
  - Administrar tus cuentas de correo electrónico
- **Dominio:** Un dominio de Internet es una red de identificación asociada a un grupo de dispositivos o equipos conectados a la red Internet. El propósito principal de los nombres de dominio en Internet y del sistema de nombres de dominio (DNS), es traducir las direcciones IP de cada nodo activo en la red, a términos memorizables y fáciles de encontrar. Esta abstracción hace posible que cualquier servicio (de red) pueda moverse de un lugar geográfico a otro en la red Internet, aun cuando el cambio implique que tendrá una dirección IP diferente. Sin la ayuda del sistema de nombres de dominio, los usuarios de Internet tendrían que acceder a cada servicio web utilizando la dirección IP del nodo (por ejemplo, sería necesario utilizar `http://192.0.32.10` en vez de `http://ejemplo.com`). Hay algunos entes internacionales que administran los dominios en Internet de primer nivel (terminados en `com`, `edu`, `gov`, `mil`, `org`, `net`, `int`, etc.). Hay dominios de segundo nivel que han sido delegados a cada país para su administración (`ar` para Argentina, `cn` para China, `mx` para México, `co` para Colombia, etc.)
  - **Información:** Es lo que se aloja en el servidor, lo que ve la gente que escribe tu dirección web. Como ocurre en cualquier sistema informático la información debe estar organizada en un formato que el sistema pueda interpretar. Tú no puedes abrir ver un video con un procesador de texto. Lo mismo ocurre con los sitios de Internet. Deben estar realizados con programas que sean compatibles con los navegadores (Chrome, Explorer, Firefox, etc.).  
Hay 2 maneras de hacer un sitio web:  
Con programas o aplicaciones que están destinados a la construcción y edición de sitios y aplicaciones Web basados en estándares, creando archivos de texto en lenguajes HTML, PHP, Asp, Aspx, JavaScript, JSP, Python, Ruby. (MS FrontPage, Adobe Dreamweaver, etc.).

Con sistemas de gestión de contenido que se instalan en el servidor y permiten modificar el sitio de manera sencilla vía Internet (WordPress , Joomla, etc.).

La gran ventaja de utilizar un gestor de contenido es la posibilidad de agregar, quitar y modificar los contenidos fácilmente. Esto le da mucho dinamismo al sitio y permite la actualización del sitio sin necesidad de recurrir a un programador.

Estos programas han sido optimizados para que los buscadores los posicionen en los primeros resultados de las búsquedas.

Con el tiempo se han creado grandes comunidades de colaboradores que han desarrollado muchas aplicaciones que dan potencia y versatilidad a estos sistemas.

Espero que te sirva como introducción al maravilloso mundo de la generación y divulgación de contenido.

- **El contenido es la nueva moneda:** Cuando la gente hace una búsqueda en un buscador como Google en realidad está haciendo una pregunta. Si tu logras averiguar las preguntas que hacen tus compradores potenciales y la respondes de forma de ayudarlos genuinamente a solucionar sus problemas, las ventas vendrán por añadidura (Botas, s.f.).

## **2.2 APLICACIÓN WEB**

### **2.2.1 Definición de un Aplicación Web**

Una aplicación Web es un sitio Web que contiene páginas con contenido sin determinar, parcialmente o en su totalidad. El contenido final de una página se determina sólo cuando el usuario solicita una página del servidor Web. Dado que el contenido final de la página varía de una petición a otra en función de las acciones del visitante, este tipo de página se denomina página dinámica.

Las aplicaciones Web se crean en respuesta a diversas necesidades o problemas. En esta sección se describen los usos más habituales de las aplicaciones Web y se proporciona un ejemplo sencillo (Adobe, 2019).

### **2.2.2 Usos comunes de las aplicaciones Web**

Las aplicaciones Web pueden tener numerosos usos tanto para los visitantes como para los ingenieros de desarrollo, entre otros:

- Permitir a los usuarios localizar información de forma rápida y sencilla en un sitio Web en el que se almacena gran cantidad de contenido. Este tipo de aplicación web ofrece a los visitantes la posibilidad de buscar contenido, organizarlo y navegar por él de la manera que estimen oportuna. Entre los ejemplos se incluyen intranets de las empresas, Microsoft MSDN ([www.msdn.microsoft.com](http://www.msdn.microsoft.com)) y Amazon.com ([www.amazon.com](http://www.amazon.com)).
- Recoger, guardar y analizar datos suministrados por los visitantes de los sitios. En el pasado, los datos introducidos en los formularios HTML se enviaban como mensajes de correo electrónico a los empleados o a aplicaciones CGI para su procesamiento. Una aplicación Web permite guardar datos de formularios directamente en una base de datos, además de extraer datos y crear informes basados en la Web para su análisis. Ejemplos de ello son las páginas de los bancos en línea, las páginas de tiendas en línea, las encuestas y los formularios con datos suministrados por el usuario.
- Actualizar sitios Web cuyo contenido cambia constantemente. Una aplicación Web evita al diseñador Web tener que actualizar continuamente el código HTML del sitio. Los proveedores de contenido, como los editores de noticias, proporcionan el contenido a la aplicación Web y ésta actualiza el sitio automáticamente. Entre los ejemplos, se incluyen The Economist ([www.economist.com](http://www.economist.com)) y la CNN ([www.cnn.com](http://www.cnn.com)) (acens Technologies, S.L, 2018).

### **2.2.3 Funcionamiento de una Aplicación Web**

Una aplicación Web es un conjunto de páginas Web estáticas y dinámicas. Una página Web estática es aquella que no cambia cuando un usuario la solicita: el servidor Web envía la página al navegador Web solicitante sin modificarla. Por el contrario, el servidor modifica las páginas Web dinámicas antes de enviarlas al navegador solicitante. La naturaleza cambiante de este tipo de página es la que le da el nombre de dinámica.

Por ejemplo, podría diseñar una página para que mostrara los resultados del programa de salud y dejara cierta información fuera (como el nombre del empleado y sus resultados) para calcularla cuando la página la solicite un empleado en particular. En las siguientes secciones se describe con mayor detalle el funcionamiento de las aplicaciones Web (Henst, 2001).

## Procesamiento de Páginas Web Estáticas

Un sitio Web estático consta de un conjunto de páginas y de archivos HTML relacionados alojados en un equipo que ejecuta un servidor Web. Un servidor Web es un software que suministra páginas Web en respuesta a las peticiones de los navegadores Web. La petición de una página se genera cuando el usuario hace clic en un vínculo de una página Web, elige un marcador en un navegador o introduce una URL en el cuadro de texto Dirección del navegador (Adobe, 2019).

El contenido final de una página Web estática lo determina el diseñador de la página y no cambia cuando se solicita la página. A continuación, se incluye un ejemplo:

```
<html>
  <head>
    <title>Trio Motors Information Page</title>
  </head>
  <body>
    <h1>About Trio Motors</h1>
    <p>Trio Motors is a leading automobile manufacturer.</p>
  </body>
</html>
```

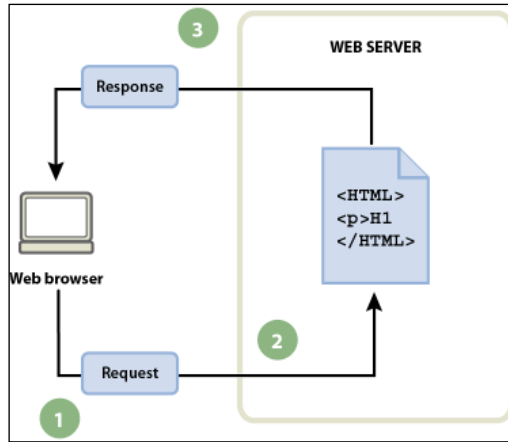
**Figura 1:** Página Web Estática

**Fuente:** <https://helpx.adobe.com/es/dreamweaver/user-guide.html/es/dreamweaver/using/web-applications.ug.html>

El diseñador escribe todas y cada una de las líneas de código HTML de la página antes de colocarla en el servidor. El código HTML no cambia una vez colocado en el servidor y por ello, este tipo de páginas se denomina página estática.

Nota: En sentido estricto, una página “estática” puede no ser estática en absoluto. Por ejemplo, una imagen de sustitución o contenido de Flash (un archivo SWF) puede hacer que una página estática tome vida. No obstante, en esta guía se habla de página estática cuando ésta se envía al navegador sin modificaciones.

Cuando el servidor Web recibe una petición de una página estática, el servidor lee la solicitud, localiza la página y la envía al navegador solicitante, como se muestra en el siguiente ejemplo:



1. El navegador web solicita la página estática.
2. El servidor localiza la página.
3. El servidor Web envía la página al navegador solicitante.

**Figura 2:** Procesamiento de Página Web Estática

**Fuente:** <https://helpx.adobe.com/es/dreamweaver/user-guide.html/es/dreamweaver/using/web-applications.ug.html>

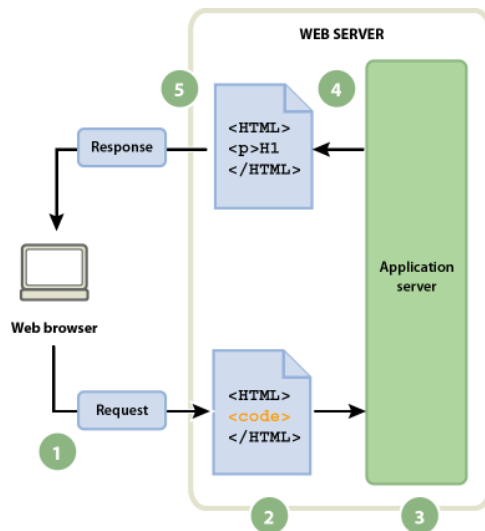
En el caso de las aplicaciones Web, algunas líneas de código no están determinadas cuando el usuario solicita la página. Estas líneas deben determinarse mediante algún mecanismo antes de enviar la página al navegador. En la siguiente sección se describe dicho mecanismo (Adobe, 2019).

### **Procesamiento de Páginas Dinámicas**

Cuando un servidor Web recibe una petición para mostrar una página Web estática, el servidor la envía directamente al navegador que la solicita. Cuando el servidor Web recibe una petición para mostrar una página dinámica, sin embargo, reacciona de distinta forma: transfiere la página a un software especial encargado de finalizar la página. Este software especial se denomina servidor de aplicaciones (Rollet, 2015).

El servidor de aplicaciones lee el código de la página, finaliza la página en función de las instrucciones del código y elimina el código de la página. El resultado es una página estática que el servidor de aplicaciones devuelve al servidor Web, que a su vez la envía al navegador solicitante. Lo único que el navegador recibe cuando llega la página es código HTML puro. A continuación, se incluye una vista de este proceso:





1. El navegador web solicita la página dinámica.
2. El servidor web localiza la página y la envía al servidor de aplicaciones.
3. El servidor de aplicaciones busca instrucciones en la página y la termina.
4. El servidor de aplicaciones pasa la página terminada al servidor web.
5. El servidor web envía la página finalizada al navegador solicitante.

Figura 3: Procesamiento de

Fuente: <https://helpx.adobe.com/es/dreamweaver/user-guide.html/es/dreamweaver/using/web-applications.ug.html>

## 2.3 SERVIDOR WEB

### 2.3.1 Definición de un Servidor Web

Un servidor web o Servidor HTTP es una pieza de software de comunicaciones que **intermedia entre el servidor en el que están alojados los datos solicitados y el computador del cliente**, permitiendo conexiones bidireccionales o unidireccionales, síncronas o asíncronas, con cualquier aplicación del cliente, incluso con los navegadores que traducen un código traducible (renderizable) a una página web determinada. O sea, se trata de programas que median entre el usuario de Internet y el servidor en donde está la información que solicita (Raffino, 2020).

Los servidores web son programas de uso cotidiano en Internet, que **emplean para comunicarse diversos protocolos de datos**, siendo el más común y de alguna manera estándar el HTTP (*HyperText Transfer Protocol*). Sin embargo, es posible también usar el término para referirse al computador en el que están guardados los archivos que componen un sitio web, junto al software necesario para cumplir con la conexión de datos web (Raffino, 2020).

### 2.3.2 Características de un Servidor Web

Un servidor web **opera en un ordenador aguardando las solicitudes de parte del navegador web** de un cliente, brindando los datos solicitados para componer una página web o, en su defecto, un mensaje de error. Los servidores web pueden ser de dos clases: estáticos y dinámicos (Raffino, 2020).

- **Servidores Estáticos.** Consisten en un computador en donde está almacenada la información y un servidor HTTP que responde a los protocolos de pedido. Su nombre proviene del hecho de que los archivos se envían tal y como están almacenados.
- **Servidores Dinámicos.** Son servidores estáticos que contienen software adicional (usualmente aplicaciones y bases de datos) que les permiten actualizar la información solicitada antes de enviarla al cliente.

### 2.3.3 Funciones de un Servidor Web

La función de los servidores web es la de **mediar entre las solicitudes de los usuarios y los archivos** que contienen la información solicitada, en el computador en el que están almacenados. Por ende, se necesita uno para publicar un sitio web, y también para tener acceso a los datos que componen una página web cualquiera. Este proceso ocurre de la siguiente manera (B, 2019):

- El usuario introduce una dirección web (URL) en su navegador y éste envía una solicitud al servidor web.
- El servidor web (software) busca los archivos pertinentes ya sea en el propio servidor (hardware) o en un servicio de hosting en el que están siempre disponibles y en línea.
- Los archivos entonces son procesados según lo solicitado y enviados de acuerdo al protocolo de transferencia, es decir, conforme a un conjunto de reglas que regulan la comunicación entre los computadores.
- El navegador recibe los archivos y ensambla el contenido de la página web que se muestra al usuario.

### 2.3.4 Servicios

- **Internet:** Es una Red de Ordenadores conectados en toda la extensión del Globo Terráqueo, que ofrece diversos servicios a sus usuarios como pueden ser el Correo electrónico, el Chat o la Web. Todos los servicios que ofrece Internet son llevados a cabo por miles de ordenadores que están permanentemente encendidos y conectados a Internet, esperando que los usuarios les soliciten los servicios y sirviéndolos una vez son solicitados.
- **Página Web:** Documento o fuente de información, generalmente en formato HTML y que puede contener hiperenlaces a otras Páginas Web. Dicha página, podrá ser accesible desde un dispositivo físico, una intranet, o Internet.
- **Navegador Web:** Para establecer conexiones con los servidores Web, y obtener la información y los servicios que estos prestan, el usuario necesita tener instalado en su equipo un programa cliente capaz de comunicarse con ellos. Estos programas son los llamados Navegadores Web. Los Navegadores Web, también llamados Visores de Web o Browsers, son aplicaciones que permiten ver en pantalla texto con formato (con palabras en negrita, y con distintas fuentes tipográficas, tamaños y colores) y presentar imágenes en línea. También permiten visualizar secuencias de vídeo y escuchar ficheros de sonido.
- **Servidor:** Un Servidor es un tipo de Software que suministra servicios a los usuarios o terminales que lo solicitan. Por ejemplo, en una típica Arquitectura Cliente-servidor, el cliente podría ser un ordenador que realiza peticiones de información a través de un programa de correo (Outlook Express por ejemplo) y, el servidor le entrega los datos en forma de correos electrónicos en respuesta a su solicitud. Un servidor no es necesariamente una máquina de última generación de grandes proporciones, no es necesariamente un superordenador; un servidor puede ser desde una computadora vieja, hasta una máquina sumamente potente (ej. Servidores web, bases de datos grandes, etc., procesadores especiales y hasta varios gigabytes de memoria) (Henst, 2001).

Todo esto depende del uso que se le dé al servidor. Si usted lo desea, puede convertir al equipo desde el cual se está trabajando en un servidor instalando un programa que trabaje

por la Red y a la que los usuarios de su red ingresen a través de un programa de servidor web como Apache. Por lo cual podemos llegar a la conclusión de que un servidor también puede ser un proceso que entrega información o sirve a otro proceso (ecured.cu, 2020).

El modelo Cliente-Servidor no necesariamente implica tener dos ordenadores, ya que un proceso cliente puede solicitar algo como una impresión a un proceso servidor en un mismo ordenador. Cliente/servidor: Esta Arquitectura consiste básicamente en un programa cliente que realiza peticiones a otro programa el servidor que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un Sistema Operativo multiusuario distribuido a través de una red de computadoras.

En esta Arquitectura la capacidad de proceso está repartida entre los Clientes y los Servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. La separación entre Cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa.

Los tipos específicos de servidores que incluyen son los Servidores Web, los Servidores de archivo, los Servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma. Una disposición muy común son los Sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema (Landa, 2019).

## **2.4 LENGUAJE PHP**

**PHP** (acrónimo recursivo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. PHP es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web (PHP Group, 2020).

El lenguaje PHP tiene la característica de permitir programar con las siguientes metodologías:

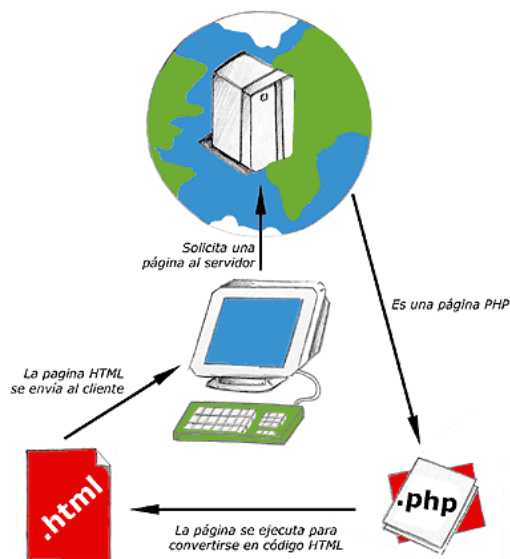
- Programación Lineal: Cuando desarrollamos todo el código disponiendo instrucciones PHP alternado con el HTML de la página.
- Programación Estructurada: Cuando planteamos funciones que agrupan actividades a desarrollar y luego dentro de la página llamamos a dichas funciones que pueden estar dentro del mismo archivo o en una librería separada.
- Programación Orientada a Objetos: Cuando planteamos clases y definimos objetos de las mismas.

PHP es un lenguaje de programación que puede hacer todo tipo de cosas: evaluar datos de formularios enviados desde un navegador, construir contenido web a medida para el navegador, hablar a una base de datos, e incluso enviar y recibir cookies (pequeños paquetes de datos que tu navegador usa para recordar cosas) (codecademy.com, s.f.).

#### 2.4.1 Funciones de PHP

En la siguiente figura se muestra el funcionamiento de PHP para determinada aplicación web, donde se deben cumplir ciertos requisitos básicos, tales como:

- El servidor solicita páginas web de cualquier formato.
- Identifica si es una página con extensión .php.
- PHP reconocible para convertirse en código HTML.
- Página HTML se envía al usuario (PHP cuya sintaxis sea correcta).



**Figura 4:** Funcionamiento de PHP

Fuente: <http://www.desarrolloweb.com/articulos/392.php>

Para interpretar un archivo, PHP simplemente interpreta el texto del archivo hasta que encuentra uno de los caracteres especiales que delimitan el inicio de código PHP. El intérprete ejecuta entonces todo el código que encuentra, hasta que encuentra una etiqueta de fin de código, que le dice al intérprete que siga ignorando el código siguiente. Este mecanismo permite embeber código PHP dentro de HTML: todo lo que está fuera de las etiquetas PHP se deja tal como está, mientras que el resto se interpreta como código.

*<? php: Indica el comienzo del código PHP*  
*?>: Indica el final del código PHP*

- **Sintaxis Básicas**

Hay cuatro conjuntos de etiquetas que pueden ser usadas para denotar bloques de código PHP. De estas cuatro, sólo 2 (`<?php ... ?>` y `<script language="php"> ...</script>`) están siempre disponibles; el resto pueden ser configuradas en el fichero de `php.ini` para ser o no aceptadas por el intérprete. Mientras que el formato corto de etiquetas (short-form tags) y el estilo ASP (ASP-style tags) pueden ser convenientes, no son portables como la versión de formato largo de etiquetas. Además, si se pretende embeber código PHP en XML o XHTML, será obligatorio el uso del formato `<?php ... ?>` para la compatibilidad con XML.

```
Formas de escapar de HTML

<?php
echo("si quieres servir documentos XHTML o XML, haz como aquí<BR>");
?>

<?
echo ("esta es la más simple, una instrucción de procesado SGML <BR>");
?>
<?= expression ?> Esto es una abreviatura de "<? echo expression ?>"<BR>

<script language="php">
echo ("muchos editores (como FrontPage) no aceptan instrucciones de
procesado<BR>");
</script>

<%
echo ("Opcionalmente, puedes usar las etiquetas ASP<BR>");
%>
<%= $variable; # Esto es una abreviatura de "<% echo . . ." %>
```

**Figura 5:** Etiquetas soportadas por PHP  
**Fuente:** <https://n-1.cc/file/download/745373>

## Programa Hola mundo con PHP embebido en código HTML:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <title> Ejemplo básico PHP</title>
  </head>
  <body>
    <?php
      echo 'Hola mundo';
    ?>
  </body>
</html>
```

**Figura 6:** PHP embebido en código HTML

**Fuente:** <https://es.wikipedia.org/wiki/PHP>

- **Variables**

Una variable es una información que se almacena temporalmente en la memoria, es decir, es una zona de la memoria que almacena información en una página PHP y que se destruye automáticamente cuando la página ya no se ejecuta.

Una variable PHP comienza siempre con \$, seguida de una letra y de una secuencia de letras, cifras o del signo \_. **Ejemplo: \$edad**

**Atención:** PHP distingue entre mayúsculas y minúsculas, por lo que *\$nombre* es distinto de *\$Nombre*.

Una variable siempre tiene un nombre y un valor. Por ejemplo, `$edad = 25`, el valor 25 se asigna a la variable `$edad` gracias al signo `=`. No es necesario definir y buscar el tipo de variable. Se hace automáticamente. De este modo, puede escribir:

```
<?php
    $edad = 25; //variable de tipo
    numérico
    //después
    $edad = 'toto'; //variable de
    tipo texto
?>
```

**Tabla 1:** Definición de tipos de variables en PHP

**Fuente:** <http://www.freelibros.org/tag/aprender-a-desarrollar-un-sitio-web-con-php-y-mysql-pdf>

Esto no origina ningún error. El tipo de variable puede cambiar. En el ejemplo anterior cambia del tipo integer al tipo string (Steve, 2016).

## 2.4.2 Formularios

Los formularios se usan en la navegación habitual por páginas web (i.e. Cuando entramos en nuestro correo web, iniciamos sesión en youtube para subir un vídeo, buscamos una página con google, o nos identificamos en facebook para acceder a nuestra red social siempre estamos enviando datos mediante los campos de un formulario (MDN contributors, 2020).

Los formularios de una página web se diseñan como el resto de la página, empleando HTML (Lenguaje de Etiquetas de HiperTexto) (MDN contributors, 2020).

### Elemento <Form> de un formulario Html

Para crear un formulario basta insertar etiquetas <form> de apertura y cierre en tu página WEB. Todos los controles (campos de texto, casillas de verificación, botones de opción o botón de enviar) que añadas entre las etiquetas <form> y </form> formarán parte del formulario que se envía al servidor WEB.

El elemento <form> cuenta con muchos atributos pero basta con utilizar 2 de ellos: action y method.

- **Atributo action:** Indica al servidor a qué página debe desplazarse cuando el usuario pulsa el botón de envío del formulario. Es decir, indica qué página recibirá la información del formulario. Por supuesto, esta página debe existir en el servidor Web.



- **Atributo method:** Controla la forma en que se envía la información al servidor y existen dos métodos, GET y POST.
  - Con GET se envía la información introducida por el usuario adjuntándola a la URL después del signo “?”.
  - Con POST se envían los valores introducidos en el formulario en el cuerpo de la solicitud HTTP.

El código siguiente crea un formulario WEB que aún no contiene ningún campo. Cuando añadamos los campos normales en un formulario, el archivo “respuesta1.php” recibirá mediante el método POST los valores introducidos en el formulario (codecademy.com, s.f.).

```
<form action="respuesta1.php"method="post">
</form>
```

**Tabla 2:** Formato de un Formulario

**Fuente:** [https://tecnoclara.wikispaces.com/file/view/Formulario\\_Web.pdf](https://tecnoclara.wikispaces.com/file/view/Formulario_Web.pdf)

### 2.4.3 Sesiones

Una sesión es un archivo que se almacena en el servidor de cada una de las personas que se conectan en su sitio Web. Se destruye automáticamente cuando la persona abandona el sitio Web. Una sesión en PHP se define con la palabra: \$\_SESSION.

\$\_SESSION es una tabla asociativa que permite almacenar cualquier valor de cada usuario. Esta tabla es válida en todas las páginas PHP del sitio Web y permite pasar variables de una página a otra. Tenga cuidado con no abusar de ella, porque si se conecta mucha gente en su sitio Web puede llenar la memoria del servidor Web (uniwebsidad, 2020).

- **session\_start():** Crea una sesión o reanuda la actual basada en un identificador de sesión pasado mediante una petición GET o POST, o pasado mediante una cookie. Para usar una sesión nominada hay que utilizar la función session\_name antes de llamar a session\_start.
- **session\_destroy():** Sirve para destruir la sesión actual. Tenga cuidado porque esta función no destruye las variables de la sesión asociadas a la sesión actual. Debe utilizar

también la función `unset($_SESSION)`. La sesión se destruirá automáticamente cuando el usuario cierre la ventana de su navegador. Por ejemplo, el siguiente código en la página `prueba.php`:

```
<?php
    session_start();
    $_SESSION['nombre'] = 'Juan';
    echo "El nombre en sesión
    es:". $_SESSION['nombre'];
?>
```

**Tabla 3:** Sesiones

**Fuente:** <http://www.freelibros.org/tag/aprender-a-desarrollar-un-sitio-web-con-php-y-mysql-pdf>

Da como resultado:

**El nombre en sesión es: Juan**

## 2.5 PROGRAMACIÓN ORIENTADA A OBJETOS EN PHP

La POO es un paradigma que tiene sus orígenes desde antes de 1990 (a partir de este año se empieza a popularizar). Por lo tanto no es ninguna excusa (menos como Desarrollador PHP) seguir a la fecha desconociendo cómo trabajar con POO o discutiendo si realmente es útil su adopción. “Los objetos son entidades que combinan estado, comportamiento e identidad”. Fundamental, los beneficios que obtenemos usando este paradigma (Landa, 2019):

- “La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.”
- 

La razón de por qué no es necesario que todos los objetos que creamos tengan un `id` como si fuera una clave primaria de una tabla (con el fin de ubicar un objeto en particular):

- “De esta forma, un objeto contiene toda la información que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases e incluso frente a objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos.”

Diferencias con respecto a la Programación Estructurada versus Programación Orientada a Objetos:

- La primera se pensó como funcionalidad por un lado y datos por otro, es decir, llamar a una función y pasarle constantemente datos para que los procese, mientras que la POO está pensada para tener todo integrado en el mismo objeto.
- “En la programación estructurada sólo se escriben funciones que procesan datos. Los programadores que emplean éste nuevo paradigma, en cambio, primero definen objetos para luego enviarles mensajes solicitándoles que realicen sus métodos por sí mismos.”

Muy importante es tener SIEMPRE en claro los conceptos FUNDAMENTALES, si no se tiene claros cuando programe OO, algo está mal, seguro errará el camino que define el paradigma: Clase, Herencia, Objeto, Método, Evento, Mensaje, Atributo, Estado Interno, Componentes de un objeto y Representación de un objeto. No dude en volver a repasarlos todas las veces que lo necesites, por más experto que se considere, siempre viene bien una relectura de nuestras bases.

Igual que el punto anterior, es fundamental tener claros estos conceptos cada vez que desarrollamos, con principal énfasis en el Principio de Ocultación (que es muy común confundir con Encapsulamiento), lo que explica por qué no deberían existir los atributos públicos ni abusar de los setter/getter (Landa, 2019).

### **2.5.1 Características de la Programación Orientada a Objetos**

Existe un acuerdo acerca de qué características contempla la "orientación a objetos". Las características siguientes son las más importantes (Landa, 2019):

- **Abstracción:** Denota las características esenciales de un objeto, donde se capturan sus comportamientos. Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar cómo se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos, y, cuando lo están, una variedad de técnicas son requeridas para ampliar una abstracción. El proceso de

abstracción permite seleccionar las características relevantes dentro de un conjunto e identificar comportamientos comunes para definir nuevos tipos de entidades en el mundo real. La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se quiere atacar.

- **Encapsulamiento:** Significa reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.
- **Modularidad:** Se denomina modularidad a la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes. Estos módulos se pueden compilar por separado, pero tienen conexiones con otros módulos. Al igual que la encapsulación, los lenguajes soportan la modularidad de diversas formas.
- **Principio de Ocultación:** Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una interfaz a otros objetos que especifica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas; solamente los propios métodos internos del objeto pueden acceder a su estado. Esto asegura que otros objetos no puedan cambiar el estado interno de un objeto de manera inesperada, eliminando efectos secundarios e interacciones inesperadas. Algunos lenguajes relajan esto, permitiendo un acceso directo a los datos internos del objeto de una manera controlada y limitando el grado de abstracción. La aplicación entera se reduce a un agregado o rompecabezas de objetos.
- **Polimorfismo:** Comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre; al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O, dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última

característica se llama asignación tardía o asignación dinámica. Algunos lenguajes proporcionan medios más estáticos (en "tiempo de compilación") de polimorfismo, tales como las plantillas y la sobrecarga de operadores de C++.

- **Herencia:** Las clases no se encuentran aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento, permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo. Esto suele hacerse habitualmente agrupando los objetos en clases y estas en árboles o enrejados que reflejan un comportamiento común. Cuando un objeto hereda de más de una clase se dice que hay herencia múltiple; siendo de alta complejidad técnica por lo cual suele recurrirse a la herencia virtual para evitar la duplicación de datos.
- **Recolección de Basura:** La recolección de basura o garbage collection es la técnica por la cual el entorno de objetos se encarga de destruir automáticamente, y por tanto desvincular la memoria asociada, los objetos que hayan quedado sin ninguna referencia a ellos. Esto significa que el programador no debe preocuparse por la asignación o liberación de memoria, ya que el entorno la asignará al crear un nuevo objeto y la liberará cuando nadie lo esté usando. En la mayoría de los lenguajes híbridos que se extendieron para soportar el Paradigma de Programación Orientada a Objetos como C++ u Object Pascal, esta característica no existe y la memoria debe desasignarse expresamente (Landa, 2019).

### 2.5.2 Elemento de la Programación Orientada a Objetos

- **Clase:** Una clase es un modelo que se utiliza para crear objetos que comparten un mismo comportamiento, estado e identidad (Robles, 2014).

```
class Persona {  
    # Propiedades  
    # Métodos  
}
```

- **Propiedades de una Clase**
  - **Públicas:** Accesibles desde cualquier parte de la aplicación.

```
class Persona {
    public $nombre;
    public $genero;
}
```

- **Protegidas:** Solo accesibles desde la clase que las declara y clases que la heredan.

```
class Persona {
    public $nombre;
    public $genero;
    private $edad;
    protected $pasaporte;
}
```

- **Privadas:** Solo accesibles desde la clase que las declara.

```
class Persona {
    public $nombre;
    public $genero;
    private $edad;
}
```

- **Estáticas:** Aquellas que no pueden ser modificadas. Estas propiedades estáticas pueden tener cualquier visibilidad: pública, privada o protegida.

```
class PersonaAPositivo extends Persona {
    public static $tipo_sangre = 'A+';
}
```

- **Clase Abstracta:** La clase abstracta tiene la misma estructura que una clase normal, solo es necesario añadir la palabra clave `abstract` al inicio de su declaración.

```
abstract class NombreDeMiClaseAbstracta {
    #...
}
```

- **Métodos de una Clase**

- Dinámicos: Aquellos que efectúan cambios a la clase y/o acceden dinámicamente a sus propiedades.
- Estáticos: No efectúan cambios de ningún tipo y su función, en consecuencia, es estática.
- Públicos, Privados o Protegidos: Donde aplica la misma visibilidad descrita en las propiedades.

```

<?php
class Ejemplo {

    public $propiedad = 'valor';
    static $variable = 'valor de variable';

    public function metodo_publico($valor) {
        $this->propiedad = $valor;
        // modifica el valor de la propiedad
    }

    static function metodo_estatico() {
        return 45;
        // siempre retorna el número 45
    }

    private function metodo_privado() {
        // a este método solo se puede acceder desde esta clase
    }

}
?>

```

**Figura 7:** Métodos de una Clase

**Fuente:** [http://vitaminaweb.com/introduccion-a-la-programacion-orientada-a-objetos-en-php\\_1046](http://vitaminaweb.com/introduccion-a-la-programacion-orientada-a-objetos-en-php_1046)

- **Objeto:** Es una entidad provista de métodos o mensajes a los cuales responde (comportamiento); atributos con valores concretos (estado); y propiedades (identidad).

```

$persona = new Persona();
/*
    El objeto, ahora, es $persona,
    que se ha creado siguiendo el modelo de la clase Persona
*/

```

**Figura 8:** Representación de un Objeto

**Fuente:** [https://pablor.files.wordpress.com/2013/12/poo\\_mvc\\_iutoms.pdf](https://pablor.files.wordpress.com/2013/12/poo_mvc_iutoms.pdf)

- **Objetos como Instancias de las Clases:** Las clases son tan solo la definición del objeto, así que para poder trabajar con ellas es necesario instanciarlas mediante la creación de un objeto. Así el objeto es la representación útil de la clase que lo define.

```

# declaro la clase
class Persona {
    #...
}

# creo el objeto instanciando la clase
$persona = new Persona();

```

**Figura 9:** Objeto como Instancia de un Clase

**Fuente:** [https://pablor.files.wordpress.com/2013/12/poo\\_mvc\\_iutoms.pdf](https://pablor.files.wordpress.com/2013/12/poo_mvc_iutoms.pdf)

- **Método:** Es el algoritmo asociado a un objeto que indica la capacidad de lo que éste puede hacer.

```
function caminar() {
    #...
}
```

**Figura 10:** Representación de un Método

Fuente: [https://pablor.files.wordpress.com/2013/12/poo\\_mvc\\_iutoms.pdf](https://pablor.files.wordpress.com/2013/12/poo_mvc_iutoms.pdf)

- **Métodos Mágicos:** Como ya podemos imaginar son métodos que nos sirven para definir el constructor y destructor en una clase.
  - **Constructor (construct):** El constructor de una clase es utilizado para inicializar la misma, así podremos asignar algunos valores a las propiedades que se crean convenientes en el momento de creación de un objeto.
  - **Destructor (destruct):** El destructor sin embargo es ejecutado en el momento en el que el objeto deja de existir. Así aquellas tareas que queramos ejecutar en el momento de liberar un objeto las definiremos en el destructor de la clase.

```
# constructor de la clase
function __construct() {
    $this->set_estado_producto('en uso');
}

# destructor de la clase
function __destruct() {
    $this->set_estado_producto('liberado');
    print 'El objeto ha sido destruido';
}
```

**Figura 11:** Métodos Mágicos

Fuente: [https://pablor.files.wordpress.com/2013/12/poo\\_mvc\\_iutoms.pdf](https://pablor.files.wordpress.com/2013/12/poo_mvc_iutoms.pdf)

- **Evento y Mensaje:** Un evento es un suceso en el sistema mientras que un mensaje es la comunicación del suceso dirigida al objeto.
- **Propiedades y Atributos:** Las propiedades y atributos, son variables que contienen datos asociados a un objeto.

```
$nombre = 'Juan';
$edad = '25 años';
$altura = '1,75 mts';
```

**Figura 12:** Propiedades y Atributos

Fuente: [https://pablor.files.wordpress.com/2013/12/poo\\_mvc\\_iutoms.pdf](https://pablor.files.wordpress.com/2013/12/poo_mvc_iutoms.pdf)

- **Herencia de Clases:** La herencia es uno de los pilares fundamentales dentro de la programación orientada a objetos. La herencia consiste en la definición de una clase a



partir de otra existente, así esta nueva clase se le llama clase extendida o clase derivada. La importancia de la herencia reside en que cuando creamos una clase extendida, esta a su vez hereda todos los métodos y propiedades de su clase base o ancestral, pudiendo a su vez implementar otros métodos o propiedades exclusivos de nuestra clase extendida (codigo-facil.com, 2019).

```
class NombreDeMiClaseMadre {
    #...
}

class NombreDeMiClaseHija extends NombreDeMiClaseMadre {
    /* esta clase hereda todos los métodos y propiedades de
    la clase madre NombreDeMiClaseMadre
    */
}
```

**Figura 13:** Herencia de Clases

**Fuente:** [https://pablor.files.wordpress.com/2013/12/poo\\_mvc\\_iutoms.pdf](https://pablor.files.wordpress.com/2013/12/poo_mvc_iutoms.pdf)

- **Encapsulamiento:** El uso de la característica `final` restringe en la programación orientada a objetos. Publica, protege y personaliza los modificadores de acceso a métodos y propiedades incorporados a partir de PHP5.
- **Modificador Público:** Este modificador se le otorga permisos de acceso al método o propiedad de acuerdo desde cualquier clase para de nuestro programa. Considerar que en el modificador `public` por defecto en caso de que no se especifica lo contrario.
- **Modificador Privado:** Este modificador establece el nivel más restrictivo, puesto que las propiedades o métodos que declaramos como `private` solo serán accesibles desde el interior de la clase.
- **Modificador Protegido:** Este modificador establece un nivel de restricción similar, pero que permite que las propiedades o métodos declarados como `protected` solo serán accesibles desde la clase `hija` o las clases `hijas` que hereden de la clase `base`.
- **Polimorfismo:** El polimorfismo, como su nombre indica, requiere múltiples formas. En programación orientada a objetos, el polimorfismo se refiere a la capacidad de acceder a múltiples funciones o métodos del mismo nombre. Es decir que se crean identificadores, o función puede tener diferentes comportamientos en función del contexto en el que sea ejecutada. Su implementación varía en función del lenguaje de programación. En algunos casos para establecer una relación polimórfica se necesitan que cada uno de los objetos implemente completamente una misma interfaz, siendo entonces necesario establecer una jerarquía de clases. Tanto en el caso de los lenguajes de programación fuertemente tipados como Java. Para el caso que son lenguajes no tipados o polimorfismo en PHP estableciendo una jerarquía de clases, ya que son métodos que `funcionan` en métodos de clases, ya que son métodos que `funcionan` en métodos de clases, ya que son métodos que `funcionan` en métodos de clases.

Declaro la función:

```
function estacionar( Vehiculo ) {}
```

Invoco la función: (soporto polimorfismo)

```
estacionar( Coche );
estacionar( Moto );
estacionar( Bus );
```

**Figura 14:** Polimorfismo

**Fuente:** <http://www.desarrolloweb.com/articulos/polimorfismo-programacion-orientada-objetos-concepto.html>

## 2.6 PATRÓN MVC

En líneas generales, MVC (Modelo-Vista- Controlador) es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos (codigo-facil.com, 2019).

Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores, o lo que es lo mismo, Model, Views & Controllers, si se prefiere en inglés.

MVC es un "invento" que ya tiene varias décadas y fue presentado incluso antes de la aparición de la Web. No obstante, en los últimos años ha ganado mucha fuerza y seguidores gracias a la aparición de numerosos frameworks de desarrollo web que utilizan el patrón MVC como modelo para la arquitectura de las aplicaciones web.

### **2.6.1 Importancia del MVC**

La rama de la ingeniería del software se preocupa por crear procesos que aseguren calidad en los programas que se realizan y esa calidad atiende a diversos parámetros que son deseables para todo desarrollo, como la estructuración de los programas o reutilización del código, lo que debe influir positivamente en la facilidad de desarrollo y el mantenimiento.

Los ingenieros del software se dedican a estudiar de qué manera se pueden mejorar los procesos de creación de software y una de las soluciones a las que han llegado es la arquitectura basada en capas que separan el código en función de sus responsabilidades o conceptos. Por tanto, cuando estudiamos MVC lo primero que tenemos que saber es que está ahí para ayudarnos a crear aplicaciones con mayor calidad (desarrolloweb.com, 2020).

Quizás, para que a todos nos queden claras las ventajas del MVC podamos echar mano de unos cuantos ejemplos:

- Aunque no tenga nada que ver, comencemos con algo tan sencillo como son el HTML y las CSS. Al principio, en el HTML se mezclaba tanto el contenido como la presentación. Es decir, en el propio HTML tenemos etiquetas como "font" que sirven para definir las características de una fuente, o atributos como "bgcolor" que definen el color de un fondo. El resultado es que tanto el contenido como la presentación estaban juntos y si algún día pretendíamos cambiar la forma con la que se mostraba una página, estábamos obligados a cambiar cada uno de los archivos HTML que componen una web, tocando todas y cada una de las etiquetas que hay en el documento. Con el tiempo se observó que eso no era

práctico y se creó el lenguaje CSS, en el que se separó la responsabilidad de aplicar el formato de una web.

- Al escribir programas en lenguajes como PHP, cualquiera de nosotros comienza mezclando tanto el código PHP como el código HTML (e incluso el Javascript) en el mismo archivo. Esto produce lo que se denomina el "Código Espaguetti". Si algún día pretendemos cambiar el modo en cómo queremos que se muestre el contenido, estamos obligados a repasar todas y cada una de las páginas que tiene nuestro proyecto. Sería mucho más útil que el HTML estuviera separado del PHP.
- Si queremos que en un equipo intervengan perfiles distintos de profesionales y trabajen de manera autónoma, como diseñadores o programadores, ambos tienen que tocar los mismos archivos y el diseñador se tiene necesariamente que relacionar con mucho código en un lenguaje de programación que puede no serle familiar, siendo que a éste quizás solo le interesan los bloques donde hay HTML. De nuevo, sería mucho más fácil la separación del código.
- Durante la manipulación de datos en una aplicación es posible que estemos accediendo a los mismos datos en lugares distintos. Por ejemplo, podemos acceder a los datos de un artículo desde la página donde se muestra éste, la página donde se listan los artículos de un manual o la página de backend donde se administran los artículos de un sitio web. Si un día cambiamos los datos de los artículos (alteramos la tabla para añadir nuevos campos o cambiar los existentes porque las necesidades de nuestros artículos varían), estamos obligados a cambiar, página a página, todos los lugares donde se consumían datos de los artículos. Además, si tenemos el código de acceso a datos disperso por decenas de lugares, es posible que estemos repitiendo las mismas sentencias de acceso a esos datos y por tanto no estamos reutilizando código.

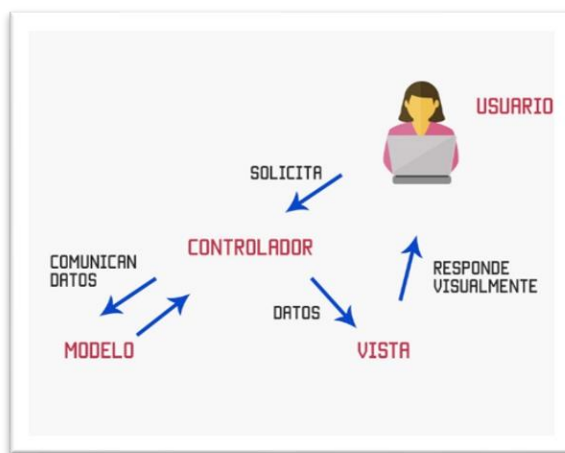
Son solo son simples ejemplos, habiendo decenas de casos similares en los que resultaría útil aplicar una arquitectura como el MVC, con la que nos obliguemos a separar nuestro código atendiendo a sus responsabilidades.

Ahora ya se puede tener una idea de las ventajas que nos puede aportar el MVC, analicemos las diversas partes o conceptos en los que debemos separar el código de nuestras aplicaciones.

## 2.6.2 Definición del Patrón MVC

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo (Universidad de Alicante, 2020).

El MVC o Modelo-Vista-Controlador es un patrón de arquitectura de software que, utilizando 3 componentes (Vistas, Models y Controladores) separa la lógica de la aplicación de la lógica de la vista en una aplicación. Es una arquitectura importante puesto que se utiliza tanto en componentes gráficos básicos hasta sistemas empresariales; la mayoría de los frameworks modernos utilizan MVC (o alguna adaptación del MVC) para la arquitectura, entre ellos podemos mencionar a Ruby on Rails, Django, AngularJS y muchos otros más (Hernandez, 2016).



**Figura 15:** Arquitectura MVC desde el punto de vista del Usuario

**Fuente:** <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>

- **Modelo**

El modelo representa la parte de la aplicación que implementa la lógica de negocio. Esto significa que es responsable de la recuperación de datos convirtiéndolos en conceptos significativos para la aplicación, así como su procesamiento, validación, asociación y cualquier otra tarea relativa a la manipulación de dichos datos. A primera vista los objetos

del modelo pueden ser considerado como la primera capa de la interacción con cualquier base de datos que podría estar utilizando tu aplicación. Pero en general representan los principales conceptos en torno a los cuales se desea implementar un programa (Cake PHP, 2020).

En el caso de una red social, la capa de modelo se haría cargo de tareas tales como guardar datos del usuario, el almacenamiento de asociaciones con amigos, el almacenamiento y la recuperación de fotos de los usuarios, encontrar sugerencias de nuevos amigos, etc. Mientras que los objetos del modelo pueden ser considerados como “Amigo”, “Usuario”, “Comentario” y “Foto”.

- Contiene el núcleo de la funcionalidad (dominio) de la aplicación.
- Encapsula el estado de la aplicación.
- No sabe nada / independiente del Controlador y la Vista.

- **Vista**

La vista hace una presentación de los datos del modelo estando separada de los objetos del modelo. Es responsable del uso de la información de la cual dispone para producir cualquier interfaz de presentación de cualquier petición que se presente.

Por ejemplo, como la capa de modelo devuelve un conjunto de datos, la vista los usaría para hacer una página HTML que los contenga. O un resultado con formato XML para que otras aplicaciones puedan consumir.

La capa de la Vista no se limita únicamente a HTML o texto que represente los datos, sino que puede ser utilizada para ofrecer una amplia variedad de formatos en función de sus necesidades tales como videos, música, documentos y cualquier otro formato que puedas imaginar (codigo-facil.com, 2019).

- Es la presentación del Modelo.
- Puede acceder al Modelo pero nunca cambiar su estado.
- Puede ser notificada cuando hay un cambio de estado en el Modelo

- **Controlador**

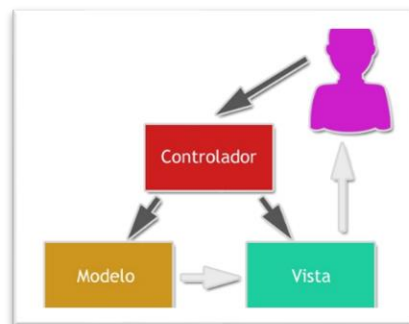
La capa del controlador gestiona las peticiones de los usuarios. Es responsable de responder la información solicitada con la ayuda tanto del modelo como de la vista.

Los controladores pueden ser vistos como administradores cuidando de que todos los recursos necesarios para completar una tarea se deleguen a los trabajadores más adecuados. Espera peticiones de los clientes, comprueba su validez de acuerdo a las normas de autenticación o autorización, delega la búsqueda de datos al modelo y selecciona el tipo de respuesta más adecuado según las preferencias del cliente. Finalmente delega este proceso de presentación a la capa de la Vista.

- Reacciona a la petición del Cliente, ejecutando la acción adecuada y creando el modelo pertinente (Hernandez, 2016).

### 2.6.3 Arquitectura de Aplicaciones MVC

El gráfico que se presenta a continuación servirá para entender un poco mejor cómo colaboran las distintas capas que componen la arquitectura de desarrollo de software en el patrón MVC (desarrolloweb.com, 2020).



**Figura 16:** Arquitectura MVC

Fuente: <http://www.desarrolloweb.com/articulos/que-es-mvc.html>

En esta imagen se representada con flechas los modos de colaboración entre los distintos elementos que formarían una aplicación MVC, junto con el usuario. Como se puede ver, los controladores, con su lógica de negocio, hacen de puente entre los modelos y las vistas. Pero además en algunos casos los modelos pueden enviar datos a las vistas. A continuación se muestra paso a paso cómo sería el flujo de trabajo característico en un esquema MVC.

- El usuario realiza una solicitud al sitio web. Generalmente estará desencadenada por acceder a una página del sitio. Esa solicitud llega al controlador.
- El controlador comunica tanto con modelos como con vistas. A los modelos les solicita datos o les manda realizar actualizaciones de los datos. A las vistas les solicita la salida correspondiente, una vez se hayan realizado las operaciones pertinentes según la lógica del negocio.
- Para producir la salida, en ocasiones las vistas pueden solicitar más información a los modelos. En ocasiones, el controlador será el responsable de solicitar todos los datos a los modelos y de enviarlos a las vistas, haciendo de puente entre unos y otros. Sería corriente tanto una cosa como la otra, todo depende de nuestra implementación; por eso esa flecha coloreado de otro color.
- Las vistas envían al usuario la salida. Aunque en ocasiones esa salida puede ir de vuelta al controlador y sería éste el que hace el envío al cliente, por eso la flecha en otro color (codecademy.com, s.f.).

#### **2.6.4 Lógica de Negocio / Lógica de Aplicación**

Hay un concepto que se usa mucho cuando se explica el MVC que es la "lógica de negocio". Es un conjunto de reglas que se siguen en el software para reaccionar ante distintas situaciones. En una aplicación el usuario se comunica con el sistema por medio de una interfaz, pero cuando acciona esa interfaz para realizar acciones con el programa, se ejecutan una serie de procesos que se conocen como la lógica del negocio. Este es un concepto de desarrollo de software en general (Botas, s.f.).

La lógica del negocio, aparte de marcar un comportamiento cuando ocurren cosas dentro de un software, también tiene normas sobre lo que se puede hacer y lo que no se puede hacer. Eso también se conoce como reglas del negocio. Bien, pues en el MVC la lógica del negocio queda del lado de los modelos. Ellos son los que deben saber cómo operar en diversas situaciones y las cosas que pueden permitir que ocurran en el proceso de ejecución de una aplicación.

Por ejemplo, pensemos en un sistema que implementa usuarios. Los usuarios pueden realizar comentarios. Pues si en un modelo nos piden eliminar un usuario nosotros debemos borrar

todos los comentarios que ha realizado ese usuario también. Eso es una responsabilidad del modelo y forma parte de lo que se llama la lógica del negocio.

Incluso, en nuestra aplicación podría haber usuarios especiales, por ejemplo "administradores" y que no está permitido borrar, hasta que no les quitemos el rol de administrador. Eso también lo deberían controlar los modelos, realizando las comprobaciones necesarias antes de borrar efectivamente el usuario.

Sin embargo existe otro concepto que se usa en la terminología del MVC que es la "lógica de aplicación", que es algo que pertenece a los controladores. Por ejemplo, cuando me piden ver el resumen de datos de un usuario. Esa acción le llega al controlador, que tendrá que acceder al modelo del usuario para pedir sus datos. Luego llamará a la vista apropiada para poder mostrar esos datos del usuario. Si en el resumen del usuario queremos mostrar los artículos que ha publicado dentro de la aplicación, quizás el controlador tendrá que llamar al modelo de artículos, pedirle todos los publicados por ese usuario y con ese listado de artículos invocar a la vista correspondiente para mostrarlos. Todo ese conjunto de acciones que se realizan invocando métodos de los modelos y mandando datos a las vistas forman parte de la lógica de la aplicación.

Otro ejemplo, tenemos un sistema para borrar productos. Cuando se hace una solicitud a una página para borrar un producto de la base de datos, se pone en marcha un controlador que recibe el identificador del producto que se tiene que borrar. Entonces le pide al modelo que lo borre y a continuación se comprueba si el modelo nos responde que se ha podido borrar o no. En caso que se haya borrado queremos mostrar una vista y en caso que no se haya borrado queremos mostrar otra. Este proceso también está en los controladores y lo podemos denominar como lógica de la aplicación (Hernandez, 2016).

### **2.6.5 Ventajas de utilizar MVC**

Una separación total entre lógica de negocio y presentación. A esto se le pueden aplicar opciones como el multilinguaje, distintos diseños de presentación, etc. sin alterar la lógica de negocio. La separación de capas como presentación, lógica de negocio, acceso a datos es fundamental para el desarrollo de arquitecturas consistentes, reutilizables y más fácilmente mantenibles, lo que al final resulta en un ahorro de tiempo en desarrollo en posteriores



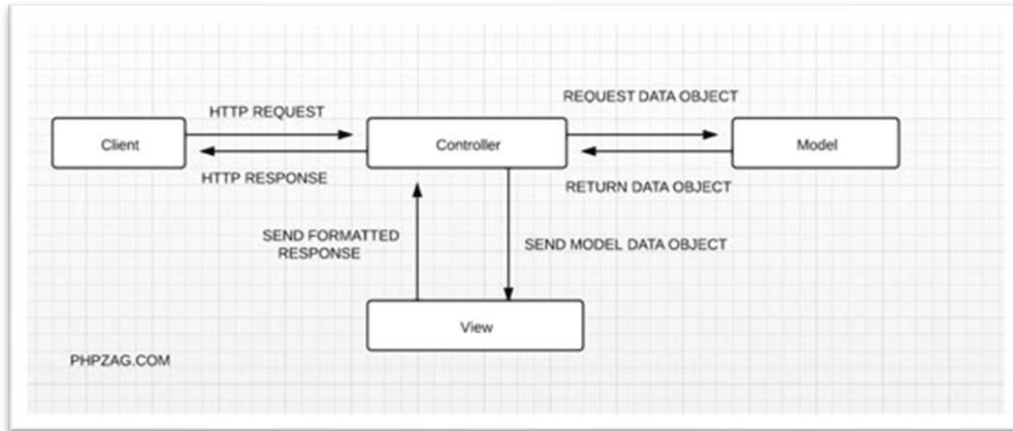
proyectos. Al existir la separación de vistas, controladores y modelos es más sencillo realizar labores de mejora como:

- Agregar nuevas vistas.
- Agregar nuevas formas de recolectar las órdenes del usuario (interpretar sus modelos mentales).
- Modificar los objetos de negocios bien sea para mejorar el performance o para migrar a otra tecnología.
- Las labores de mantenimiento también se simplifican y se reduce el tiempo necesario para ellas. Las correcciones solo se deben hacer en un solo lugar y no en varios como sucedería si tuviésemos una mezcla de presentación e implementación de la lógica del negocio.
- Las vistas también son susceptibles de modificación sin necesidad de provocar que todo el sistema se paralice. Adicionalmente el patrón MVC propende a la especialización de cada rol del equipo, por tanto, en cada liberación de una nueva versión se verán los resultados (Creativos Online, s.f.).

### **2.6.6 ¿Cómo funciona el MVC?**

El funcionamiento básico del patrón MVC, puede resumirse en (Robles, 2014):

1. El usuario realiza una petición.
2. El controlador captura la petición.
3. Hace la llamada al modelo correspondiente.
4. El modelo sera el encargado de interactuar con la base de datos.
5. El controlador recibe la información y la envía a la vista.
6. La vista muestra la información.

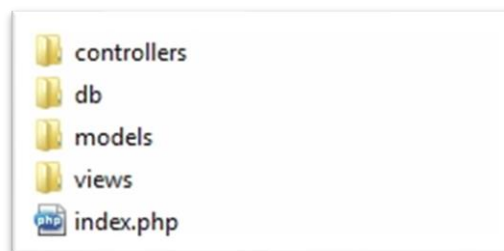


**Figura 17:** Esquema funcionamiento del MVC en PHP

Fuente: <https://victorroblesweb.es/2013/11/18/tutorial-mvc-en-php-nativo/>

### 2.6.7 ¿Cómo implementar el MVC en PHP?

Para implementar el MVC es imprescindible crear una estructura de ficheros parecida a esta (Robles, 2014):



**Figura 18:** Estructura del MVC en PHP

Fuente: <https://victorroblesweb.es/2013/11/18/tutorial-mvc-en-php-nativo/>

### 2.6.8 Ejemplo de la funcionalidad de un programa MVC en PHP y Mysql

#### 2.6.8.1 Implementación de la base de datos Mysql

- Levantar el servidor local Xampp,
- Luego procedemos a crear la base de datos MYSQL en la dirección <http://localhost/phpmyadmin> con el nombre de "mvc"
- Dentro de la BD crearemos una tabla llamada "productos" con tres campos:
  - id: Con clave primaria y autoincrementable
  - nombre: De tipo String varchar de 50

- precio: Campo con notación decimal
- Ejecutar en la ventana de Sql De Xampp

```
CREATE TABLE `productos` (
  `id` int(11) NOT NULL,
  `nombre` varchar(50) NOT NULL,
  `precio` decimal(7,2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `productos`
  ADD PRIMARY KEY (`id`);
ALTER TABLE `productos`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=1;
```

**Figura 19:** Esquema de la Base de Dattos  
 Fuente: <https://codea.app/php/mvc-en-php>

### 2.6.8.2 Estructura de archivos del Proyecto Web

Este proyecto tendrá la siguiente estructura de archivos distribuidos de la siguiente manera (codea.app, 2018):

- **controlador**
  - index.php
- **modelo**
  - index.php
- **vista**
  - css
  - app.css
  - layout
  - header.php
  - footer.php
  - editar.php
  - actualizar.php
  - borrar.php
  - index.php
- index.php
- config.php

## **2.7 BASE DE DATOS EN MYSQL**

- **DBMS**

Un Sistema Gestor de Bases de Datos o SGBD, también llamado DBMS (Data Base Management System) como una colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos. La colección de esos datos se denomina Base de Datos o BD, (DB Data Base) (McGraw-Hill, 2020).

Antes de aparecer los SGBD (década de los setenta), la información se trataba y se gestionaba utilizando los típicos sistemas de gestión de archivos que iban soportados sobre un sistema operativo. Éstos consistían en un conjunto de programas que definían y trabajaban sus propios datos. Los datos se almacenan en archivos y los programas manejan esos archivos para obtener la información. Si la estructura de los datos de los archivos cambia, todos los programas que los manejan se deben modificar; por ejemplo, un programa trabaja con un archivo de datos de alumnos, con una estructura o registro ya definido; si se incorporan elementos o campos a la estructura del archivo, los programas que utilizan ese archivo se tienen que modificar para tratar esos nuevos elementos. En estos sistemas de gestión de archivos, la definición de los datos se encuentra codificada dentro de los programas de aplicación en lugar de almacenarse de forma independiente, y además el control del acceso y la manipulación de los datos vienen impuesto por los programas de aplicación. Esto supone un gran inconveniente a la hora de tratar grandes volúmenes de información.

Surge así la idea de separar los datos contenidos en los archivos de los programas que los manipulan, es decir, que se pueda modificar la estructura de los datos de los archivos sin que por ello se tengan que modificar los programas con los que trabajan. Se trata de estructurar y organizar los datos de forma que se pueda acceder a ellos con independencia de los programas que los gestionan (McGraw-Hill, 2020).

### **2.7.1 Componentes de los SGBD**

Los SGBD son paquetes de software muy complejos que deben proporcionar una serie de servicios que van a permitir almacenar y explotar los datos de forma eficiente. Los componentes principales son los siguientes (acens Technologies, S.L, 2018):

- **Lenguajes de los SGBD**

Todos los SGBD ofrecen lenguajes e interfaces apropiadas para cada tipo de usuario: administradores, diseñadores, programadores de aplicaciones y usuarios finales. Los lenguajes van a permitir al administrador de la BD especificar los datos que componen la BD, su estructura, las relaciones que existen entre ellos, las reglas de integridad, los controles de acceso, las características de tipo físico y las vistas externas de los usuarios.

Los lenguajes del SGBD se clasifican en:

- **Lenguaje de Definición de Datos (LDD o DDL):** Se utiliza para especificar el esquema de la BD, las vistas de los usuarios y las estructuras de almacenamiento. Es el que define el esquema conceptual y el esquema interno. Lo utilizan los diseñadores y los administradores de la BD.
- **Lenguaje de Manipulación de Datos (LMD o DML):** se utilizan para leer y actualizar los datos de la BD. Es el utilizado por los usuarios para realizar consultas, inserciones, eliminaciones y modificaciones. Los hay procedurales, en los que el usuario será normalmente un programador y especifica las operaciones de acceso a los datos llamando a los procedimientos necesarios. Estos lenguajes acceden a un registro y lo procesan. Las sentencias de un LMD procedural están embebidas en un lenguaje de alto nivel llamado anfitrión. Las BD jerárquicas y en red utilizan estos LMD procedurales. No procedurales son los lenguajes declarativos. En muchos SGBD se pueden introducir interactivamente instrucciones del LMD desde un terminal, también pueden ir embebidas en un lenguaje de programación de alto nivel. Estos lenguajes permiten especificar los datos a obtener en una consulta, o los datos a modificar, mediante sentencias sencillas. Las BD relacionales utilizan lenguajes no procedurales como SQL (Structured Query Language) o QBE (Query By Example).
- La mayoría de los SGBD comerciales incluyen lenguajes de cuarta generación (4GL) que permiten al usuario desarrollar aplicaciones de forma fácil y rápida, también se les llama herramientas de desarrollo.

- **Diccionario de Datos**

El diccionario de datos es el lugar donde se deposita información acerca de todos los datos que forman la BD. Es una guía en la que se describe la BD y los objetos que la forman. El diccionario contiene las características lógicas de los sitios donde se almacenan los datos del sistema, incluyendo nombre, descripción, alias, contenido y organización. Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información (Creativos Online, s.f.).

En una BD relacional, el diccionario de datos proporciona información acerca de:

- La estructura lógica y física de la BD.
- Las definiciones de todos los objetos de la BD: tablas, vistas, índices, disparadores, procedimientos, funciones, etcétera.
- El espacio asignado y utilizado por los objetos.
- Los valores por defecto de las columnas de las tablas.
- Información acerca de las restricciones de integridad.
- Los privilegios y roles otorgados a los usuarios.
- Auditoría de información, como los accesos a los objetos.

Un diccionario de datos debe cumplir las siguientes características:

- Debe soportar las descripciones de los modelos conceptual, lógico, interno y externo de la BD.
- Debe estar integrado dentro del SGBD.
- Debe apoyar la transferencia eficiente de información al SGDB. La conexión entre los modelos interno y externo debe ser realizada en tiempo de ejecución.
- Debe comenzar con la reorganización de versiones de producción de la BD. Además debe reflejar los cambios en la descripción de la BD. Cualquier cambio a la descripción de programas ha de ser reflejado automáticamente en la librería de descripción de programas con la ayuda del diccionario de datos.
- Debe estar almacenado en un medio de almacenamiento con acceso directo para la fácil recuperación de información.

- **Seguridad e Integridad de Datos**

Un SGBD proporciona los siguientes mecanismos para garantizar la seguridad e integridad de los datos (McGraw-Hill, 2020):

- Debe garantizar la protección de los datos contra accesos no autorizados, tanto intencionados como accidentales. Debe controlar que sólo los usuarios autorizados accedan a la BD.
- Los SGBD ofrecen mecanismos para implantar restricciones de integridad en la BD. Estas restricciones van a proteger la BD contra daños accidentales. Los valores de los datos que se almacenan deben satisfacer ciertos tipos de restricciones de consistencia y reglas de integridad, que especificará el administrador de la BD. El SGBD puede determinar si se produce una violación de la restricción.
- Proporciona herramientas y mecanismos para la planificación y realización de copias de seguridad y restauración.
- Debe ser capaz de recuperar la BD llevándola a un estado consistente en caso de ocurrir algún suceso que la dañe.
- Debe asegurar el acceso concurrente y ofrecer mecanismos para conservar la consistencia de los datos en el caso de que varios usuarios actualicen la BD de forma concurrente.

- **Administrador de la Base de Datos**

En los sistemas de gestión de BBDD actuales existen diferentes categorías de usuarios. Estas categorías se caracterizan porque cada una de ellas tiene una serie de privilegios o permisos sobre los objetos que forman la Base de Datos

El DBA (Database Administrator,) tiene una gran responsabilidad ya que posee el máximo nivel de privilegios. Será el encargado de crear los usuarios que se conectarán a la BD. En la administración de una BD siempre hay que procurar que haya el menor número de administradores, a ser posible una sola persona (Rollet, 2015).

## 2.7.2 MYSQL

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento.

Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo (Rollet, 2015).

MySQL está disponible para múltiples plataformas, la seleccionada para los ejemplos de este libro es GNU/Linux. Sin embargo, las diferencias con cualquier otra plataforma son prácticamente nulas, ya que la herramienta utilizada en este caso es el cliente `mysql-client`, que permite interactuar con un servidor MySQL (local o remoto) en modo texto. De este modo es posible realizar todos los ejercicios sobre un servidor instalado localmente o, a través de Internet, sobre un servidor remoto.

Para la realización de todas las actividades, es imprescindible que dispongamos de los datos de acceso del usuario administrador de la base de datos.

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL pertenece a la compañía sueca MySQL AB, a la que le pertenece casi todos los derechos del código fuente. La compañía desarrolla y mantiene el sistema, vendiendo soporte y servicios, como también las licencias para usar MySQL.

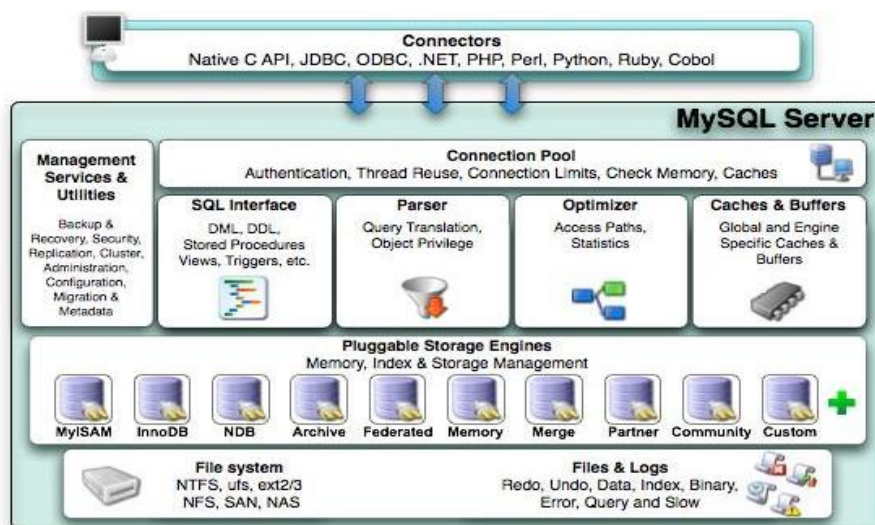
Actualmente son muchas las razones para escoger a MySQL como una solución de misión crítica para la administración de datos (Rodríguez, 2015):

- **Costo:** MySQL es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.
- **Asistencia:** MySQL AB ofrece contratos de asistencia a precios razonables y existe una nutrida y activa comunidad MySQL.
- **Velocidad:** Es mucho más rápido que la mayoría de sus rivales.



- **Potencia:** SQL es un lenguaje muy potente para consulta de bases de datos; usar un motor ahorra una enorme cantidad de trabajo.
- **Funcionalidad:** MySQL dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL ANSI, volcados online, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación.
- **Portabilidad:** Se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad. SQL es también un lenguaje estandarizado, de modo que las consultas hechas usando SQL son fácilmente portables a otros sistemas y plataformas. Esto, unido al uso de C/C++ proporciona una portabilidad enorme.
- **Facilidad de Uso:** Resulta fácil de utilizar y de administrar. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.
- **Seguridad:** En forma de permisos y privilegios, determinados usuarios tendrán permiso para consulta o modificación de determinadas tablas. Esto permite compartir datos sin que peligre la integridad de la base de datos o protegiendo determinados contenidos.
- **Accesibilidad:** Acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones.

## ARQUITECTURA DE MYSQL



**Figura 20:** Arquitectura de Myssql

Fuente: <http://www.gedlc.ulpgc.es/docencia/abd/Recursos/MySQL-Intro-features-benefits-SPANISH.pdf>

## **Subsistemas Primario**

- Motor de Consultas
  - Analizador de la sintaxis
  - Optimizador de Consultas
  - Cache de Consultas
  - Ejecución del componente
- Gestión de Transacciones
- Gestión de Buffer
- Utilidades de Gestión y Servicios
- Motor de Almacenamiento

## **CARACTERÍSTICAS DE MYSQL**

La siguiente lista describe algunas de las características más importantes del software de base de datos MySQL (uniwebsidad, 2020).

- Velocidad
  - Arquitectura Multi-hilos: Múltiples clientes tienen acceso concurrente
  - Cache de Consultas: Cachea los resultados de las consultas comunes
- Fiabilidad
- Facilidad de uso
  - Interfaz de línea de comandos
  - Herramientas gráficas-Escritorio/ basadas en Web
- Soporte Multi-Usuario
  - Múltiples clientes tienen acceso concurrente a una (o más) bases de datos simultáneamente
  - Sistema de privilegios de usuarios potente y flexible
  - Esquemas de autenticación basados en usuario-máquina
- Escalabilidad
  - Bases de datos conteniendo ¡50 millones de registros!Empresas-Yahoo!, NASA, Google.com
- Portabilidad
  - Unix/ No-Unix: Linux, Solaris, Windows.

- Intel x86, Alpha, SPARC, PowerPC
- Cumplimiento de estándares
 

MySQL opera en diferentes modos que se ajustan a los diferentes estándares SQL

Modos: los modos de servidor SQL define a MySQL que sintaxis debe soportar y qué tipo de controles de validación de datos se deberá realizar.
- Internacionalización
  - Soporte para varios lenguajes y juegos de caracteres
- Amplio soporte de aplicaciones
  - Base de datos MySQL de aplicaciones para el Escritorio y la Web
  - APIs para: C/C++, Java, PHP, Perl, Ruby, Python.
- Programas Almacenados
  - Procedimientos y funciones almacenados
  - Disparadores (Triggers)
- Vistas
- Planificador de Eventos

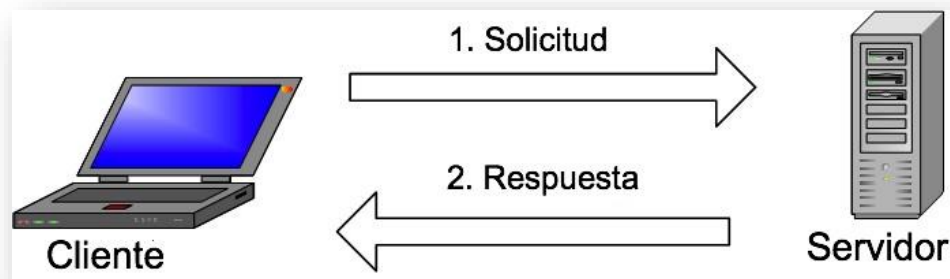
## **VENTAJAS DE MYSQL**

- MySQL software es Open Source.
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet.
- El software MySQL usa la licencia GPL

## DESVENTAJAS DE MYSQL

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS).

### ¿Cómo funciona MySQL?



**Figura 21:** Funcionamiento de Mysql

**Fuente:** <https://www.hostinger.es/tutoriales/que-es-mysql/>

La imagen explica la estructura básica cliente-servidor. Uno o más dispositivos (clientes) se conectan a un servidor a través de una red específica. Cada cliente puede realizar una solicitud desde la interfaz gráfica de usuario (GUI) en sus pantallas, y el servidor producirá el output deseado, siempre que ambas partes entiendan la instrucción. Sin meternos demasiado a fondo en temas técnicos, los procesos principales que tienen lugar en un entorno MySQL son los mismos, y son (B, 2019):

MySQL crea una base de datos para almacenar y manipular datos, definiendo la relación de cada tabla. Los clientes pueden realizar solicitudes escribiendo instrucciones SQL específicas en MySQL. La aplicación del servidor responderá con la información solicitada y esta aparecerá frente a los clientes.

Y eso es prácticamente todo. Desde el lado de los clientes, generalmente enfatizan qué GUI de MySQL usar. Cuanto más ligera y fácil de usar sea la GUI, más rápidas y fáciles serán sus actividades de administración de datos. Algunas de las GUI de MySQL más populares son MySQL WorkBench, SequelPro, DBVisualizer y Navicat DB Admin Tool. Algunas de ellas son gratuitas, mientras que otras son comerciales, otras son exclusivamente para macOS y otras son compatibles con los principales sistemas operativos. Los clientes deben elegir la

GUI en función de sus necesidades. Para la administración de bases de datos web, incluido un sitio de WordPress, la opción más obvia es phpMyAdmin.

### 2.7.3 XAMPP

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. Desde la versión "5.6.15", XAMPP cambió la base de datos de MySQL A MariaDB. El cual es un fork de MySQL con licencia GPL (apachefriends.org, 2020).



**Figura 22:** Xampp

Fuente: <https://www.apachefriends.org/es/index.html/>

El programa se distribuye bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.

XAMPP solamente requiere descargar y ejecutar un archivo ZIP, tar, exe o fkl, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará. XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin. Para instalar XAMPP se requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar los programas por separado. Puede encontrarse tanto en versión completa, así como en una versión más ligera que es portátil (apachefriends.org, 2020).

## 2.8 CÓDIGO QR

### 2.8.1 Definición de Código QR

Los códigos QR, ( en inglés QR Code) son un tipo de códigos de barras bidimensionales. A diferencia de un código de barras convencional ( por ejemplo EAN-13, Código 3 de 9, UPC), la información está codificada dentro de un cuadrado, permitiendo almacenar gran cantidad de información alfanumérica. Los códigos QR son fácilmente identificables por su forma cuadrada y por los tres cuadros ubicados en las esquinas superiores e inferior izquierda (codigos-qr.com, 2015).



**Figura 23:** Código QR

**Fuente:** <https://www.codigos-qr.com/>

### 2.8.2 ¿Para qué sirve el Código QR?

Los códigos QR están ganando popularidad debido a su multiplicidad. Puedes usarlos para obtener feedback con el objetivo de mejorar tus productos y servicios, aumentar la participación de los clientes con imágenes o videos o incluso promocionar tu negocio con eventos y cupones. ¡Todo esto con un solo escaneo!

Aunque el desarrollo inicial de los Códigos QR tenía como objetivo principal su utilización en la industria de la automoción, hoy por hoy la posibilidad de leer códigos QR desde teléfonos y dispositivos móviles permite el uso de Qr Codes en un sinnúmero de aplicaciones completamente diferentes de las que originales como pueden ser:

- Publicidad
- Campañas de marketing
- Merchandising

- Diseño Gráfico
- Papelería corporativa ( tarjetas de visita, catálogos)
- Internet, Webs, blogs

### **2.8.3 ¿Qué puede contener un Código QR?**

Un Código QR es una manera fácil y sencilla de interactuar con un dispositivo móvil y permitir realizar acciones automáticamente con el terminal como, por ejemplo (acens Technologies, S.L, 2018):

- Abrir la URL de una página Web o perfil social
- Leer un Texto
- Enviar un email
- Enviar un SMS
- Realizar un llamada telefónica
- Guardar un evento en la agenda
- Ubicar una posición geográfica en un google maps

### **2.8.4 ¿Cómo leer un Código QR?**

Ya sólo te falta la herramienta que te permita leer los códigos QR desde tu teléfono móvil o dispositivo. Existen múltiples lectores QR gratuitos para la mayoría de móviles y marcas ( iPhone, Samsung, Nokia, BlackBerry, LG etc..) encuentra el lector apropiado para tu terminal o sistema operativo y empieza desde ya a descubrir lo que esconde cada QR Code,

### **2.8.5 ¿Cómo generar códigos QR utilizando PHP?**

Para generar este código, nos apoyaremos en el uso de la librería “**PHP QR Code**” que nos proporciona todas las funcionalidades necesarias para su creación. Además de esta librería, el proyecto estará formado por tres archivos:

- index.php: Será donde se pinte el formulario con las opciones que podemos elegir para generar nuestro código.
- generate\_code.php: Se encargará de mostrar el código generado.

- `ajax_generate_code.js`: Aquí estará el código encargado de realizar la llamada al archivo `generate_code.php` utilizando Ajax y mostrando en pantalla el resultado final.

## Generar códigos QR con PHP

Información :

Nivel del código (ECC) :

Tamaño :

**Figura 24:** Generar Código QR con PHP

**Fuente:** <https://blog.acens.com/white-papers/como-generar-codigos-qr-utilizando-php/>

Como hemos dicho anteriormente, este archivo contendrá todo el código HTML que se encargará de mostrar al usuario el formulario con las diferentes opciones para generar el código QR. (acens Technologies, S.L, 2018).


El resultado final sería algo similar a lo que se muestra en la siguiente imagen.

## Generar códigos QR con PHP

Información :

Nivel del código (ECC) :

Tamaño :



**Figura 25:** Resultado del Código QR con PHP

**Fuente:** <https://blog.acens.com/white-papers/como-generar-codigos-qr-utilizando-php/>



**CAPÍTULO III**  
**ANÁLISIS Y DISEÑO**

### **3.1 RECOPIACIÓN DE INFORMACIÓN**

#### **3.1.1 DESCRIPCIÓN DE LA EMPRESA**

La empresa @PC es una empresa Riobambeña que se desempeña en el área tecnológica, inicia en el año 2001 con la dirección del Ing. Vicente Cujano como Gerente Propietario de la empresa. Dedicada a la distribución y comercializa de componentes, partes, piezas, equipos de computación, equipos electrónicos y tecnológicos. Además, provee servicios de mantenimiento reparación, diagnóstico de computadoras y asesoría de sistemas.

Actualmente se encuentra ubicada en las calles Rocafuerte entre 10 de Agosto y Primera Constituyente. La empresa cuenta con más de 19 años de vasta experiencia en el área de mercadeo y ventas de productos y servicios mediante planes estratégicos y de negocios bien elaborados. Los mismos que permitirán posicionar a la empresa en el mercado ecuatoriano y así tener mayor participación de mercado. La empresa actualmente no cuenta con una aplicación web donde se pueda registrar y llevar el seguimiento de los equipos que ingresan al área de reparación y mantenimiento.

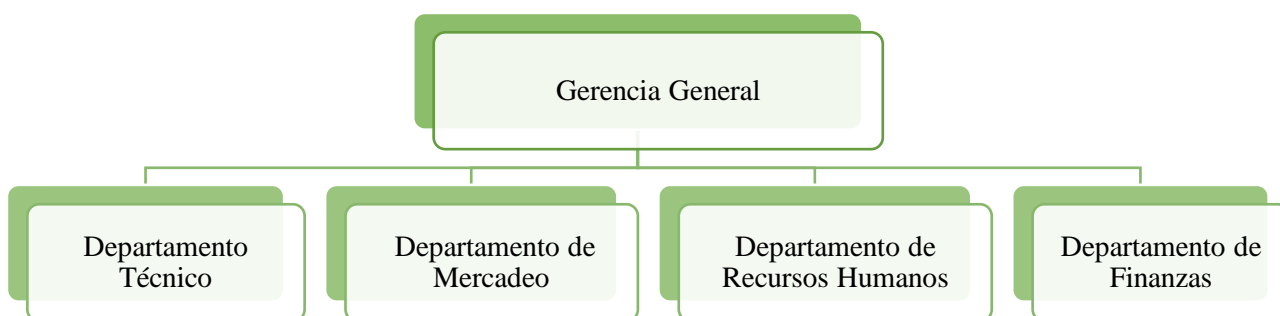
#### **3.1.2 MISIÓN**

Somos una empresa de Servicios Técnicos especializados en equipos de cómputo. Nuestra misión es siempre prestar a nuestros clientes servicios de confianza, respaldados con calidad, responsabilidad y cumplimiento, teniendo como primera prioridad la satisfacción de nuestros clientes por nuestros servicios.

#### **3.1.3 VISIÓN**

En nuestra visión queremos ser una empresa de referencia, que camina con el cambio de la tecnología y la sociedad, dando a conocer las posibilidades de los estándares y tecnologías libres. Esta labor se debe desempeñar de forma ética y satisfactoria para nosotros, nuestros clientes y el resto de la sociedad.

### 3.1.4 ORGANIGRAMA DE LA INSTITUCIÓN



**Figura 26:** Organigrama de la Empresa @PC  
**Fuente:** Empresa @PC

### 3.1.5 TIPOS DE INVESTIGACIÓN

- **Investigación Bibliográfica:** Se aplica este tipo de investigación puesto se utiliza preliminarmente en el proceso de elaboración del marco teórico de la investigación, ya que por medio de ella se logran reunir los más importantes estudios, investigaciones, datos e información sobre el problema formulado. Para la presente investigación se recopiló información de libros, revistas y artículos científicos publicados en internet.
- **Investigación de Campo:** Implica una combinación del método de observación de participante, entrevistas y análisis. Para la presente investigación se realizó una entrevista al personal del @PC, donde se pudo conocer la situación actual del departamento, así como los procesos que involucran las actividades que se realizan.

## 3.2 ANÁLISIS

### 3.2.1 ESTUDIO DE FACTIBILIDAD

El estudio de factibilidad es un instrumento que sirve para orientar la toma de decisiones de un proyecto. Ayudará a determinar si el proyecto está técnico, operacional, económico y legalmente apto para su desarrollo. A continuación, se detalla cada una de las factibilidades:

- **FACTIBILIDAD TÉCNICA**

El presente sistema de “Tickets” es factible desarrollar en la empresa “@PC”, ya que cuenta con todos los recursos materiales y software necesarios para su implementación, lo que permitirá un óptimo desarrollo e implantación del mismo. La desarrolladora posee los conocimientos necesarios y suficientes acerca de: aplicaciones web, herramientas de desarrollo a utilizar y el gestor de bases de datos, para realizar el sistema de “Tickets” en la empresa “@PC” que permitirá brindar una mejor atención al cliente y una óptima gestión de las reparaciones que la empresa atiende diariamente.

- **FACTIBILIDAD OPERATIVA**

La aplicación web de “Ticket” es factible, ya que al realizar la verificación en la empresa se requiere automatizar el proceso de control y mantenimientos de los equipos de cómputo del departamento de la empresa “@PC”, debido a que en la actualidad se la viene realizando de forma manual. La aplicación será desarrollada con una interfaz interactiva, de rápida respuesta y amigable para el usuario, permitiendo navegar por la aplicación sin ningún tipo de inconveniente. También permitirá la óptima gestión de las reparaciones brindando un rápido acceso a la información mediante reportes autogenerados. Para facilidad de uso posterior a la capacitación del personal el sistema contar con su respectivo manual de operaciones.

- **FACTIBILIDAD ECONÓMICA**

El costo con referente a la creación del sistema no será muy elevado debido que se manejará software libre.

<b>SOFTWARE</b>	<b>VALOR</b>
<b>VSCode</b>	\$ 0.00
<b>Mysql</b>	\$ 0.00
<b>Hosting</b>	\$ 70.00
<b>Dominio</b>	\$ 40.00
<b>TOTAL</b>	\$ 110.00

**Tabla 4:** Factibilidad Económica

**Fuente:** Autor

- **FACTIBILIDAD LEGAL**

La aplicación propuesta está desarrollada con herramientas Open Sources, por lo tanto, no tendrá inconvenientes legales a futuro. Al ser un sistema informático de Control de Reparación de Equipos nuevo, todos los derechos de autoría y propiedad intelectual, por lo que es factible en su implementación.

La factibilidad legal nos permite determina los derechos que tienen los autores sobre la documentación realizada por estos en este proyecto, la cual es exclusividad de los desarrolladores de este sistema, por tal motivo queda prohibida la distribución y reproducción de este documento, tales como la publicación impresa o su grabación.

### **3.2.2 ANÁLISIS DE LOS REQUERIMIENTOS**

En el análisis de requerimientos, gracias al conjunto de técnicas y procedimientos nos permiten conocer los elementos necesarios para definir el proyecto software. Es una tarea de ingeniería del software que permite especificar las características operacionales del software, indicar la interfaz del software con otros elementos del sistema y establecer las restricciones que debe cumplir el software.

- **Requerimientos Funcionales (RF)**

Los requerimientos funcionales ayudarán a definir qué hará el sistema, cuándo lo hará, y restricciones que tendrá. A continuación, se detalla cada uno de los requerimientos:

<b>CÓDIGO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>	<b>PRIORIDAD</b>
<b>RF-1</b>	Iniciar de Sesión	La aplicación debe permitir el acceso únicamente a usuarios que esten registrado para gestionar la aplicación	ALTA
<b>RF-3</b>	Gestionar Usuarios	La aplicación web debe permitir registrar, modificar y eliminar los usuarios que gestionen el sistema.	ALTA
<b>RF-4</b>	Gestionar Clientes	La aplicación web debe permitir registrar, modificar y eliminar los clientes.	ALTA
<b>RF-5</b>	Gestionar Equipos	La aplicación web debe permitir registrar, modificar y eliminar los equipos que ingresan al departamento técnico para su reparación.	ALTA
<b>RF-6</b>	Administrar Tickets de Reparaciones	La aplicación debe permitir gestionar los tickets de reparaciones y por los que estados que cada uno de ellos vaya pasando.	ALTA
<b>RF-7</b>	Generar Reportes	La aplicación web debe permitir generar reportes de los tickets generados.	MEDIA
<b>RF-8</b>	Visualizar ticket enviado al correo electrónico	La aplicación web debe permitir visualizar un ticket de reparación de un cliente por medio de un link enviado al correo electrónico, codificado en código QR	ALTA

**Tabla 5:** Requerimientos Funcionales

**Fuente:** Autor

- **Requerimientos No Funcionales(RFN)**

<b>CÓDIGO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>	<b>PRIORIDAD</b>
<b>RNF-1</b>	Seguridad de Acceso	La aplicación web debe permitir la creación de claves de acceso para el ingreso a la aplicación según el tipo de usuario registrado.	ALTA
<b>RNF-2</b>	Seguridad lógica y de datos	Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.	ALTA
<b>RNF-3</b>	Usabilidad	El sistema debe contar con manuales de usuario estructurados adecuadamente.  El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.	ALTA
<b>RNF-4</b>	Eficiencia	Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.	ALTA
<b>RNF-5</b>	Responsivo	La aplicación web debe poseer un diseño “Responsive” a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes.	ALTA
<b>RNF-6</b>	Escalabilidad	Debe estar en la capacidad de permitir en un futuro el desarrollo de nuevas funcionalidades, modificar o eliminar.	ALTA

**Tabla 6:** Requerimientos No Funcionales

**Fuente:** Autor

### 3.2.3 CASOS DE USO

Los casos de uso describen los intercambios entre el sistema que se está describiendo y las personas que interactúan. Los Actores que interactúan con el sistema se muestra a continuación:

- **Administrador:** Este actor administrador realiza la gestión de Clientes, Usuarios, Equipos, Gestión de Tickets Reparados, Generación de Reportes.

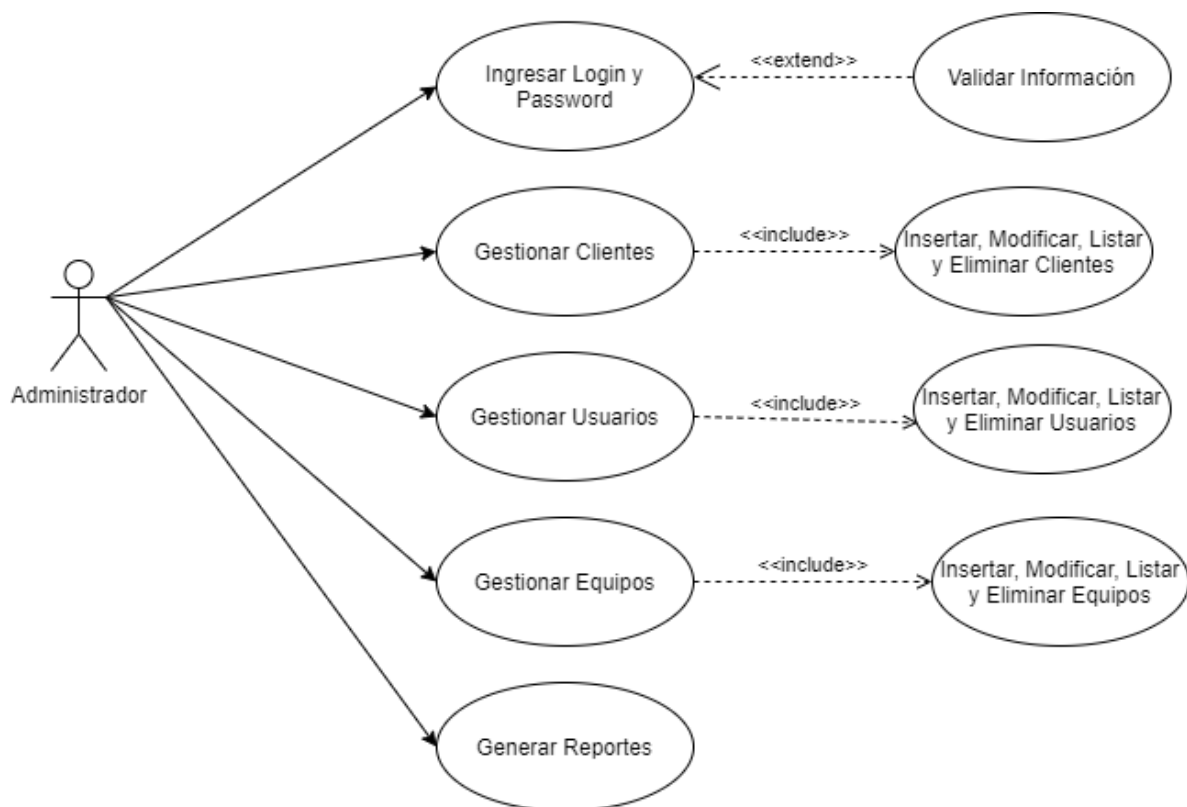
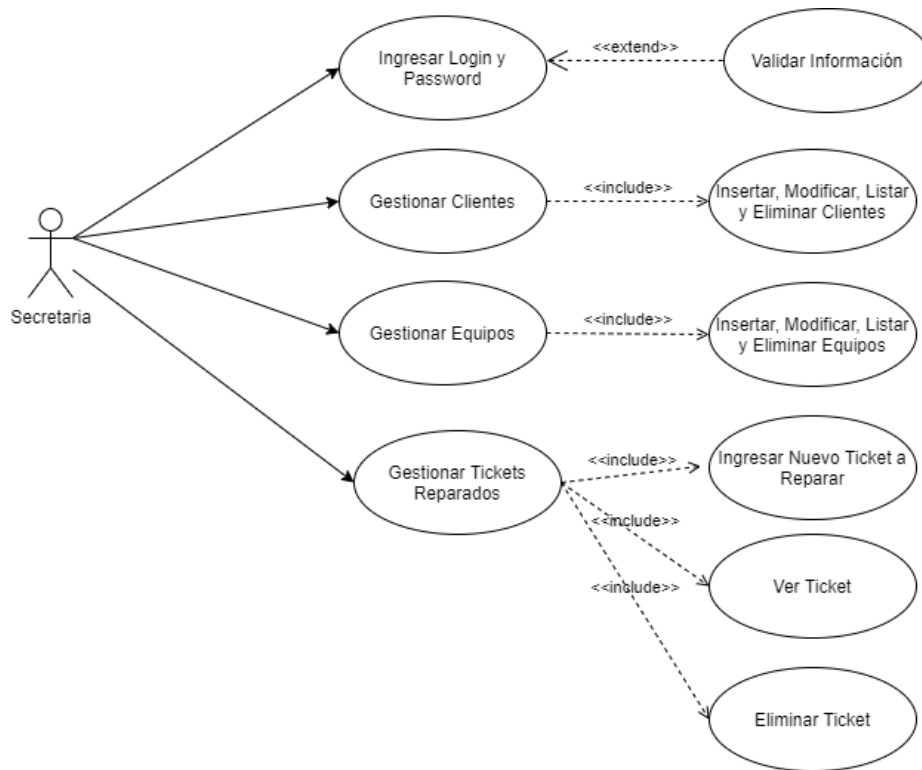


Figura 27: Diagrama de Caso de Uso Administrador

Fuente: Autor

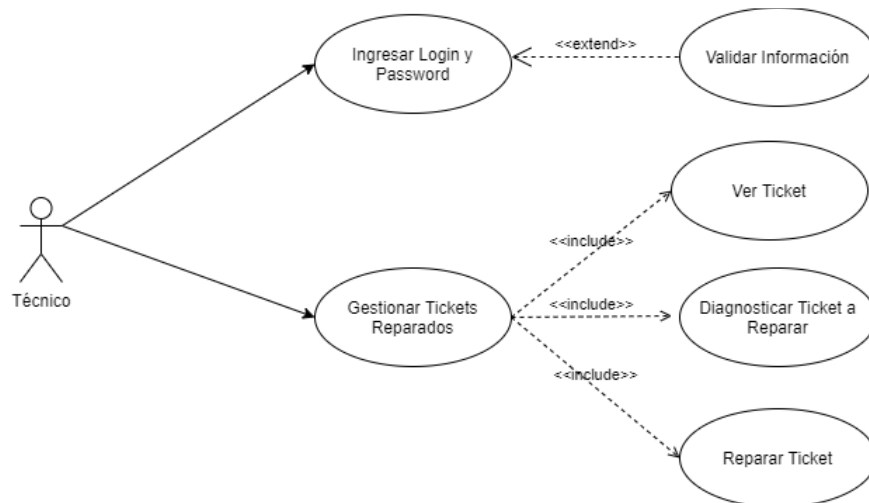
- **Secretaria:** Este actor realiza la Gestión de Equipos, Clientes y Gestión de Tickets Reparados.





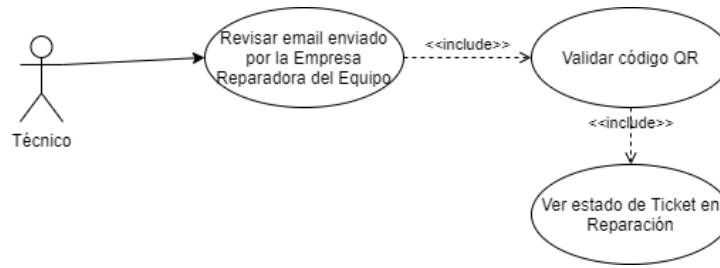
**Figura 28:** Diagrama de Caso de Uso Secretaria  
**Fuente:** Autor

- **Técnico:** Es el actor que solo puede realizar la Gestión de Tickets Reparados tickets que tenga asignados.



**Figura 29:** Diagrama de Caso de Uso Técnico  
**Fuente:** Autor

- **Cliente:** Es el actor que solo puede realizar revisar la información del ticket del equipo que se encuentre en reparación.

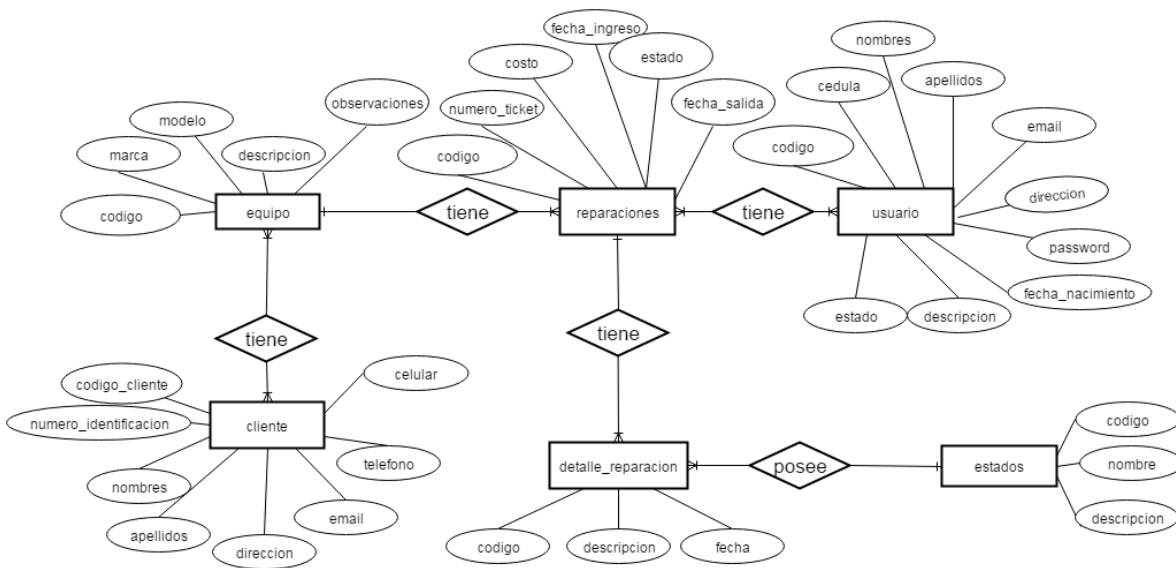


**Figura 30:** Diagrama de Caso de Uso Cliente  
Fuente: Autor

### 3.3 DISEÑO

El diseño de base de datos es un proceso que se guía por varios principios bien definidos, partiendo de un dominio del cual se obtendrá un modelo conceptual, seguidamente un modelo lógico y finalmente obtener un modelo físico y poder implementarlo.

#### 3.3.1 Diseño Conceptual



**Figura 31:** Diagrama Conceptual  
Fuente: Autor

### 3.3.2 Modelo Relacional



Figura 32: Diseño Conceptual  
Fuente: Autor

### 3.3.3 Diccionario de Datos

En el diccionario de datos se presenta la descripción de cada una de las tablas que se utiliza en el sistema.

#### Significado de las Columnas de cada Tablas de la Base de Datos

- Column Name (Nombre de la Columna)
- DataType (Tipo de Dato)
- PK(Primary Key, Llave Primaria)
- NN (No Nulo, Not Nulo)
- AI (Campo Autoincremental)
- Description (Descripción)

## Descripción de las Tablas de la Base de Datos

- acceso

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
<b>Id</b>	INT(11)	✓	✓					✓		
<b>tipo_usuario_id</b>	INT(11)		✓							
<b>url</b>	VARCHAR(1024)		✓							

**Tabla 7:** Tabla acceso

Fuente: Autor

- cliente

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
<b>id</b>	INT(11)	✓	✓					✓		
<b>nombres</b>	VARCHAR(64)		✓							
<b>apellidos</b>	VARCHAR(64)		✓							
<b>identificacion</b>	VARCHAR(13)		✓							
<b>direccion</b>	VARCHAR(512)		✓							
<b>telefono</b>	VARCHAR(10)		✓							
<b>celular</b>	VARCHAR(10)		✓							

<b>email</b>	VARCHAR(128)		✓							
<b>eliminado</b>	INT(11)		✓						0	

**Tabla 8:** Tabla Cliente

Fuente: Autor

- equipo**

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
<b>id</b>	INT(11)	✓	✓					✓		
<b>nombre</b>	VARCHAR(128)		✓							
<b>marca</b>	VARCHAR(64)		✓							
<b>modelo</b>	VARCHAR(64)								NULL	
<b>numero_serie</b>	VARCHAR(64)		✓							
<b>observaciones</b>	VARCHAR(2048)								NULL	
<b>eliminado</b>	VARCHAR(45)		✓						'0'	

**Tabla 9:** Tabla equipo

Fuente: Autor

- estado**

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
<b>id</b>	INT(11)	✓	✓					✓		

<b>nombre</b>	VARCHAR(45)		✓							
<b>descripcion</b>	VARCHAR(512)		✓							

**Tabla 10:** Tabla estado

**Fuente:** Autor

- **historial**

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
<b>id</b>	INT(11)	✓	✓					✓		
<b>reparacion_id</b>	INT(11)		✓							
<b>estado_id</b>	INT(11)		✓							
<b>id_usuario</b>	INT(11)		✓							
<b>fecha_registro</b>	DATE		✓							
<b>observaciones</b>	VARCHAR(2048)								NULL	
<b>activo</b>	INT(11)		✓						0	

**Tabla 11:** Tabla historial

**Fuente:** Autor

- **reparación**

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
<b>id</b>	INT(11)	✓	✓					✓		
<b>cliente_id</b>	INT(11)		✓							

<b>equipo_id</b>	INT(11)		✓							
<b>tecnico_id</b>	INT(11)		✓							
<b>id_usuario</b>	INT(11)		✓							
<b>fecha_registro</b>	DATE		✓							
<b>fecha_ingreso</b>	DATE		✓							
<b>observacion</b>	VARCHAR(2048)		✓							
<b>eliminado</b>	INT(11)		✓						0	

**Tabla 12:** Tabla reparacion

Fuente: Autor

- **tipo\_usuario**

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
<b>id</b>	INT(11)	✓	✓					✓		
<b>nombre</b>	VARCHAR(45)		✓							
<b>descripcion</b>	VARCHAR(512)		✓							

**Tabla 13:** Tabla tipo\_usuario

Fuente: Autor

- usuario

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
<b>id</b>	INT(11)	✓	✓					✓		
<b>tipo_usuario_id</b>	INT(11)		✓							
<b>nombres</b>	VARCHAR(64)		✓							
<b>apellidos</b>	VARCHAR(64)		✓							
<b>identificacion</b>	VARCHAR(13)		✓							
<b>direccion</b>	VARCHAR(512)		✓							
<b>telefono</b>	VARCHAR(10)		✓							
<b>celular</b>	VARCHAR(10)		✓							
<b>email</b>	VARCHAR(128)		✓							
<b>usuario</b>	VARCHAR(64)		✓							
<b>password</b>	VARCHAR(128)		✓							
<b>eliminado</b>	INT(11)		✓						0	
<b>genero</b>	VARCHAR(2)		✓							

**Tabla 14:** Tabla usuario

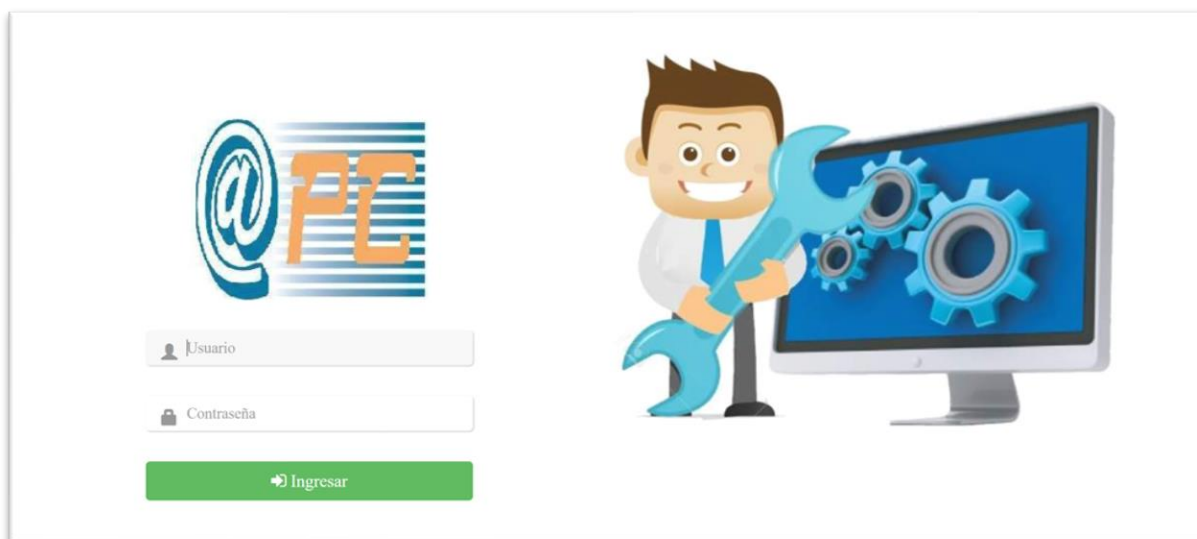
**Fuente:** Autor



### 3.3.4 Diseño de Interfaces

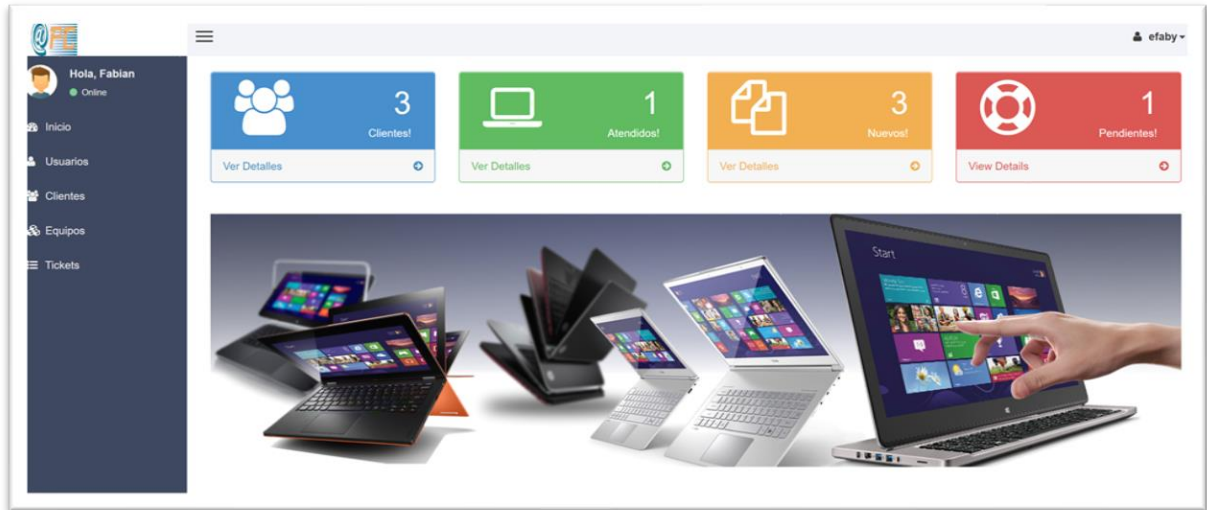
El diseño de la interfaz de usuario crea un medio eficaz de comunicación entre los seres humanos y la computadora. Siguiendo un conjunto de principios de diseño de la interfaz, el diseño identifica los objetos y acciones de ésta y luego crea una pantalla que constituye la base del prototipo de la interfaz de usuario. A continuación se muestra las principales interfaces de la aplicación:

- **Inicio de Sesión:** Al iniciar la sesión desde el computador se conectan directamente a la aplicación, son una medida de seguridad diseñada para evitar el acceso no autorizado a información confidencial de la aplicación.



**Figura 33:** Inicio de Sesión  
**Fuente:** Autor

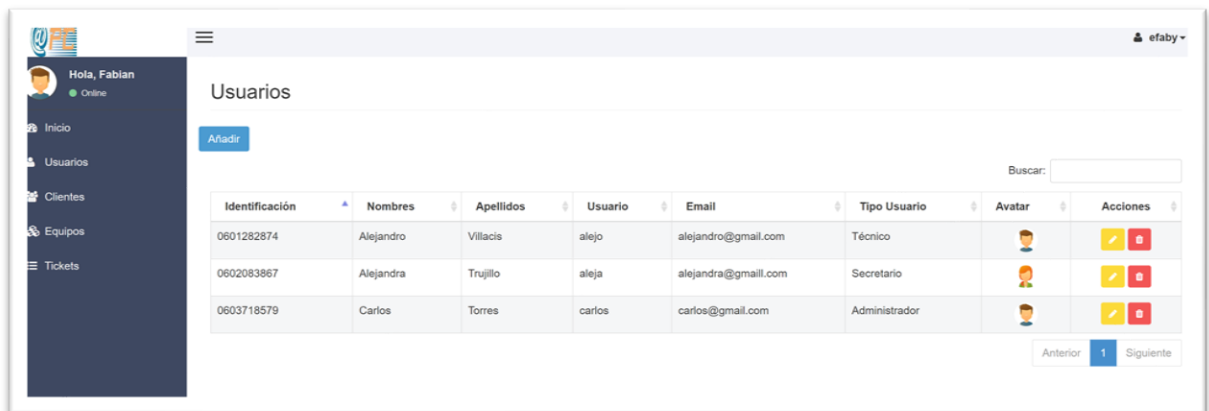
- **Pantalla Principal:** Luego de iniciar sesión, se presentará la pantalla principal de la aplicación con el menú de opciones.



**Figura 34:** Pantalla Principal

**Fuente:** Autor

- **Administración de Usuarios:** En esta pantalla se gestionará los usuarios.



**Figura 35:** Pantalla de Administración de Usuarios







**Fuente:** Autor

- **Administración de Clientes:** En esta pantalla se gestionará los clientes.

Clientes

Añadir

Buscar:

Identificación	Nombres	Apellidos	Email	Teléfono	Acciones
0603020462	Cristina	Veloz	crisrina@gmail.com	032456788	 
0603108799	Maria	Palacios	maria@gmail.com	032828628	 
0603745342	Jose	Tierra	jose@gmail.com	22222222	 

Anterior 1 Siguiente

**Figura 36:** Pantalla de Administración de Clientes







**Fuente:** Autor

- **Administración de Equipos:** En esta pantalla se gestionará los equipos.

Equipos

Añadir

Buscar:

Serie	Nombre Equipo	Marca	Modelo	Acciones
123	impresora	epson	lx33333	 
343434	laptop	hp	ererer	 
abc-123	laptop toshiba	toshiba	lc-2012	 

Anterior 1 Siguiente

**Figura 37:** Pantalla de Administración de Equipos

**Fuente:** Autor

- **Reparación de Tickets:** En esta pantalla se gestionará la reparación de tickets.

The screenshot displays the 'Tickets Reparaciones' management interface. On the left is a dark sidebar with navigation options: Inicio, Usuarios, Clientes, Equipos, and Tickets. The main content area has a header with a user profile 'Hola, Fabian' and 'efaby' in the top right. Below the header is the title 'Tickets Reparaciones' and an 'Añadir' button. A search bar labeled 'Buscar:' is positioned above a table. The table contains four rows of ticket data. At the bottom right, there are pagination controls showing 'Anterior', '1', and 'Siguiente'.

Ticket	Cliente	Equipo	Motivo de Ingreso	Técnico	Fecha Ingreso	Estado	Acciones
TR-0001	Cristina Veloz	laptop	no vale	Alejandro Villacis	2020-10-28	Entregado	[Iconos de acciones]
TR-0002	Cristina Veloz	laptop	rttrtrtrt	Alejandro Villacis	2020-11-23	Ingreso	[Iconos de acciones]
TR-0006	Cristina Veloz	pc de escritorio	drfdidf	Alejandro Villacis	2020-11-24	Ingreso	[Iconos de acciones]
TR-0008	Maria Palacios	ppp	ssss	Alejandro Villacis	2020-11-24	Reparación	[Iconos de acciones]

**Figura 38:** Pantalla de Reparación de Tickets

**Fuente:** Autor

**CAPÍTULO IV**  
**IMPLEMENTACIÓN DE LA APLICACIÓN WEB**

## 4.1 CONFIGURACIÓN DE LAS HERRAMIENTAS DE DESARROLLO

Para el desarrollo de la aplicación fueron instaladas dos herramientas, la una fue el IDE llamado Visual Studio Code para el desarrollo de la aplicación y la otra fue un servidor independiente llamado Xampp.

- **Visual Studio Code**

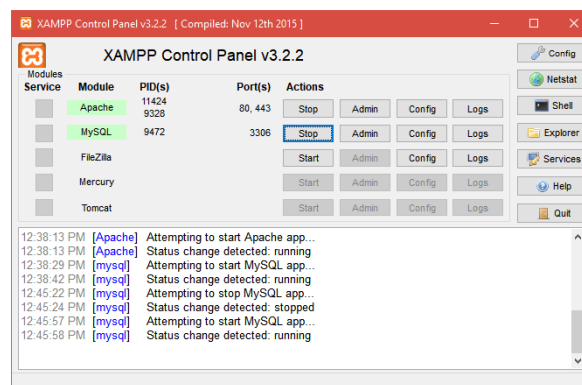
Visual Studio Code es un editor de código optimizado con soporte para operaciones de desarrollo como depuración, ejecución de tareas y control de versiones. Su objetivo es proporcionar solo las herramientas que un desarrollador necesita para un ciclo rápido de código, compilación y depuración.



**Figura 39:** Pantalla de Visual Studio Code  
**Fuente:** Autor

- **Xampp**

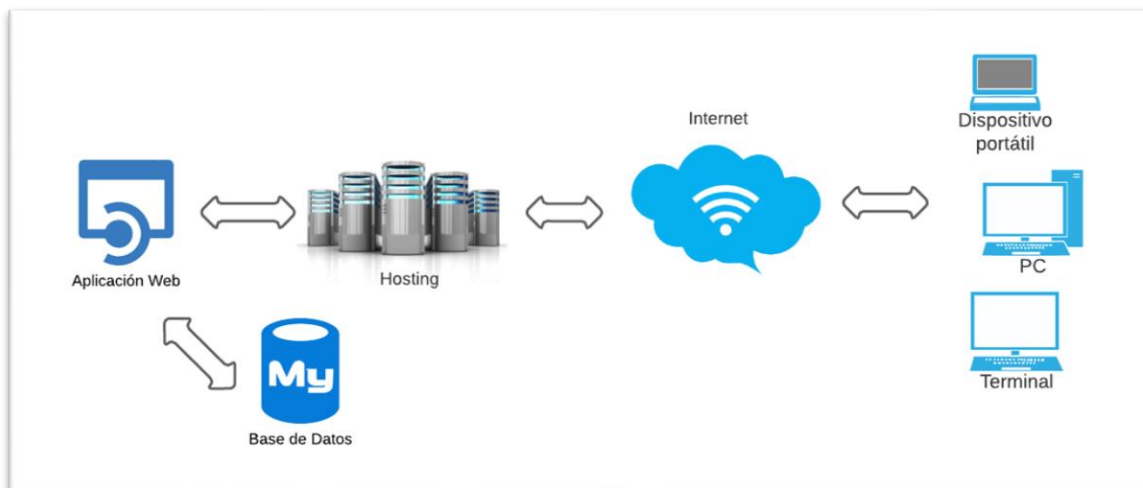
Es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.



**Figura 40:** Pantalla de Inicio de Xampp  
**Fuente:** Autor

## 4.2 ARQUITECTURA DEL SISTEMA

La arquitectura de aplicación describe los patrones y las técnicas que se utilizan para diseñar y desarrollar aplicaciones. La arquitectura proporciona un plan y las prácticas recomendadas que debe seguir al momento de diseñar la aplicación, de modo que obtenga una aplicación bien estructurada. Los recursos estructurales necesarios para la implementación de nuestro sistema se detallan en el siguiente diagrama.



**Figura 41:** Pantalla de Arquitectura del Sistema

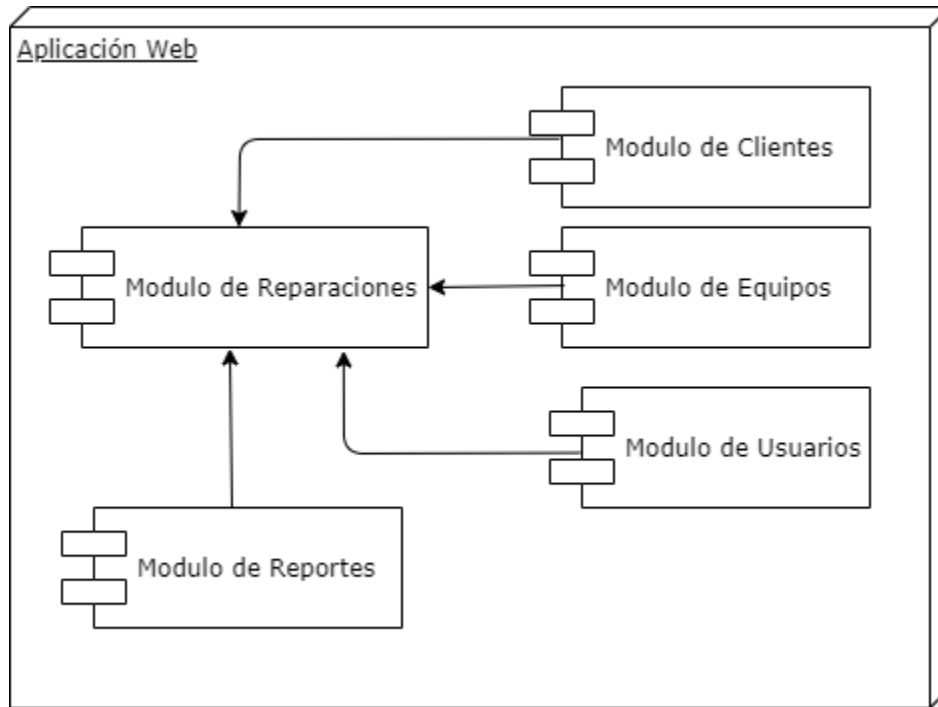
**Fuente:** Autor

## 4.3 IMPLEMENTACIÓN DE LA APLICACIÓN

Para la implementación del sistema, implica el previo conocimiento de las herramientas con las cuales se va a desarrollar el sistema, lenguaje de programación para su desarrollo, conexión del cliente con el servidor y viceversa.

### 4.3.1 Definición de Módulos de Implementación

Los módulos de implementación representan la abstracción de los requerimientos principales que fueron planteados en este documento para su desarrollo e implementación.



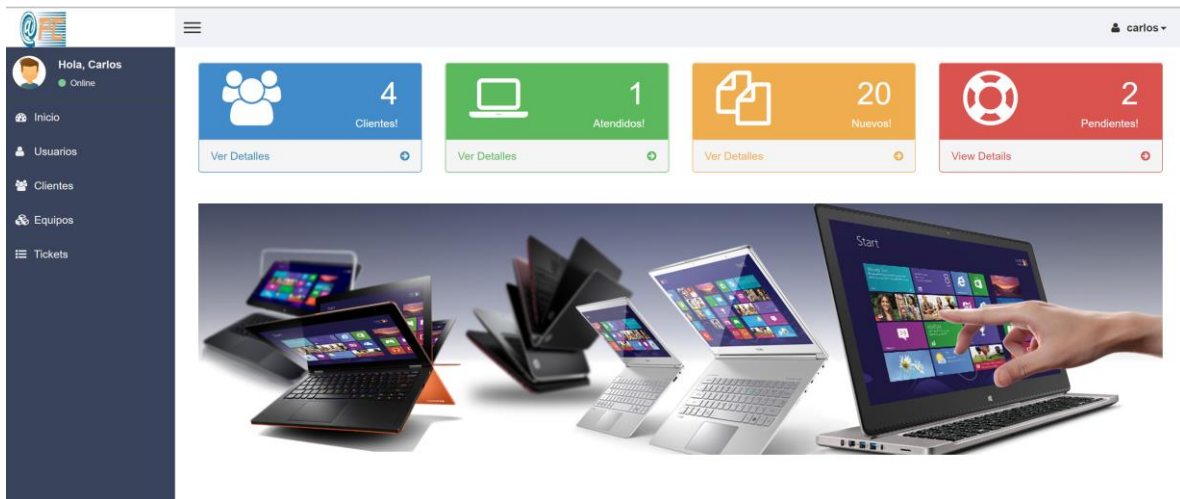
**Figura 42:** Pantalla de Módulos de la Aplicación  
**Fuente:** Autor

### 4.3.2 Desarrollo de la Aplicación

Luego de haber realizado el análisis y diseño del sistema, se desarrolla el sistema de acuerdo a los requerimientos funcionales y no funcionales que se han descrito anteriormente.

La aplicación web se desarrolló de manera que los usuarios acceden a través de internet, utilizando un navegador. Todos los datos con los que trabaja están almacenados en la web y son procesados en la misma. La aplicación tiene un inicio de sesión para el ingreso de los usuarios, una vez que se ha ingresado al aplicativo se mostrará una pantalla principal, donde se encuentra el menú de opción en función de los permisos que el usuario posea.





**Figura 43:** Pantalla Principal de la Aplicación  
**Fuente:** Autor

### 4.3.3 Implantación del Sistema

La implantación del aplicativo se la realizó en un Hosting Web que fue adquirido para el aplicativo. El dominio con el cual está vinculada es <http://www.arrobapcriobamba.com.ec>.

El proceso de publicación del aplicativo en el hosting se detalla en el Manual Técnico, que consta como (ANEXO 2).

## 4.4 PRUEBAS AL SISTEMA

Las pruebas del sistema forman parte del ciclo del desarrollo del sistema y permiten verificar el buen funcionamiento y calidad de la aplicación. Existen herramientas que están orientadas a la prueba, tanto de correcto funcionamiento como de carga y rendimiento de las aplicaciones web. A continuación, se presenta el listado de pruebas que se realizó a la aplicación:

- Pruebas de Funcionalidad
  - Comprobación de la concordancia con los requisitos del cliente
  - Validación de Formularios
  - Validación de Resultados Procesados

- Pruebas de Estándares e Interfaces
  - Validación HTML y CSS
  - Validación de un correcto despliegue
  - Validación de presentación óptima de componentes
- Pruebas de Contenidos
  - Verificación de ortografía y redacción
  - Verificación de enlaces principales
  - Verificación de imágenes en páginas
- Pruebas de Carga
  - Tiempo de acceso de los usuarios al sistema y procesamientos de los datos.
- Pruebas de Navegabilidad
  - Comprobación de sintaxis y semántica de navegación.
  - Comprobación de todo el flujo.

#### **4.5 CAPACITACIÓN AL PERSONAL**

Después de la implantación de un aplicativo web corresponde capacitar a los usuarios para que sepan utilizar de manera adecuada el nuevo recurso. Esta fase correspondiente al proceso de implantación de software se realiza habitualmente de manera presencial, aunque dada esta época de pandemia se lo recomienda realizar de manera virtual y el respaldo de manual de usuario y manual técnico. El nivel de preparación y los conocimientos previos que tengan los usuarios y las funciones que desarrollan al interior de la empresa, determina el número de horas necesarias para desarrollar la capacitación.

La capacitación está dirigida para todo el personal del departamento técnico, gerencia y secretaria de la empresa “@PC”. Las personas tendrán sus roles definidos y en función a eso podrán navegar en la aplicación.

Todas las personas antes mencionadas serán capacitadas en:

- El uso básico de un computador
- Navegabilidad en internet
- El uso del software a implementar en la empresa.

#### **4.6 MANTENIMIENTO**

La aplicación web es efectiva siempre y cuando se mantenga actualizada tanto tecnológica como funcionalmente, proporcionando al usuario lo que necesita en cada momento.

Todo proyecto de desarrollo de software va ligado a un mantenimiento de aplicaciones que posibilitan la evolución tecnológica de las empresas y evitan que las aplicaciones caigan en desuso y queden obsoletas, provocando pérdidas económicas y de tiempo.

Los servicios de mantenimiento que la empresa “@PC” tendrá será durante 1 año, donde la desarrolladora del aplicativo web se compromete a dar el soporte técnico y humano a los usuarios tanto en el desarrollo de elementos correctivos como en evolutivos que doten de mayor funcionalidad a la aplicación web implementada.

**CAPÍTULO V**

**CONCLUSIONES Y RECOMENDACIONES**

## 5.1 CONCLUSIONES

- El estudio de la programación Orientada a Objetos en el lenguaje PHP permitió ahondar los conocimientos y automatizar la programación del sitio web, obteniendo un código limpio y escalable,
- La automatización del control de tickets de reparación de equipos permitió obtener un ahorro en el tiempo de gestión del procedimiento en el departamento técnico, permitiendo además que el cliente haga un seguimiento de su equipo en reparación.
- En el desarrollo del sitio web se lo realizó utilizando una plantilla pre definida haciendo que éste proceso sea rápido y fácil de implementar. Además, se utilizaron librerías que permitieron optimizar tiempo en el desarrollo en los módulos administrativos.
- El uso del código QR facilita el acceso directo al ticket correspondiente al cliente en la aplicación mediante el uso de un dispositivo lector.

## 5.2 RECOMENDACIONES

- El óptimo levantamiento de requerimientos enfocado en las necesidades existentes en una empresa permite solucionar de una manera eficiente los problemas al momento de la implementación de cualquier aplicativo.
- Una vez levantados los requerimientos, es necesario realizar un análisis de los mismos para realizar una implementación del aplicativo que se propone realizar, definiendo adecuadamente las fases del desarrollo mediante un cronograma.
- Es importante realizar un estudio previo de las herramientas disponibles a utilizar en el desarrollo de cualquier sistema, ya que el uso correcto de las mismas permite agilizar el desarrollo, puesto que las librerías traen implementadas algunas funcionalidades útiles y reutilizables.
- La capacitación al personal que va utilizará el sistema es de vital importancia, ya que permitirá obtener el mayor provecho de la solución implementada, y así tener una mejor gestión de los tickets de reparación de la empresa.

## BIBLIOGRAFÍA

- German Silvia Bustos Forero. (3 de Marzo de 2010). *Red Gráfica Latinoamérica*. Obtenido de <http://redgrafica.com/El-lenguaje-de-programacion-PHP>
- acens Technologies, S.L. (23 de Noviembre de 2018). *acens Technologies, S.L.* Obtenido de <https://blog.acens.com/white-papers/como-generar-codigos-qr-utilizando-php/>
- Adobe. (2019). *adobe.com*. Obtenido de <https://helpx.adobe.com/es/dreamweaver/user-guide.html/es/dreamweaver/using/web-applications.ug.html>
- apachefriends.org. (2020). *apachefriends.org*. Obtenido de <https://www.apachefriends.org/es/index.html>
- B, G. (Noviembre de 2019). *Hostinger Tutoriales*. Obtenido de <https://www.hostinger.es/tutoriales/que-es-mysql/>
- Botas, S. (s.f.). *gestionempresarialrentable.com*. Obtenido de <http://gestionempresarialrentable.com/blog/componentes-basicos-para-construir-el-sitio-web-de-tu-empresa.html>
- Cake PHP. (30 de Noviembre de 2020). *Cake PHP*. Obtenido de <https://book.cakephp.org/2/es/cakephp-overview/understanding-model-view-controller.html>
- codea.app. (2018). *codea.app*. Obtenido de <https://codea.app/php/mvc-en-php>
- codecademy.com. (s.f.). *codecademy.com*. Obtenido de 2019: [https://www.codecademy.com/es/courses/web-beginner-en-aqtuv/0/1?curriculum\\_id=53a32baefed2a8232e000001](https://www.codecademy.com/es/courses/web-beginner-en-aqtuv/0/1?curriculum_id=53a32baefed2a8232e000001)
- codigo-facil.com. (2019). *codigo-facil.com*. Obtenido de <http://www.codigo-facil.com/es/poo-php-herencia.php>
- codigos-qr.com. (2015). *codigos-qr.com*. Obtenido de <https://www.codigos-qr.com/>
- Creativos Online. (s.f.). *Creativos Online*. Obtenido de <https://www.creativosonline.org/blog/93.html>
- desarrolloweb.com. (28 de Julio de 2020). *desarrolloweb.com*. Obtenido de <https://desarrolloweb.com/articulos/que-es-mvc.html>
- ecured.cu. (2020). *ecured.cu*. Obtenido de [https://www.ecured.cu/Sitio\\_Web](https://www.ecured.cu/Sitio_Web)

- Henst, C. V. (23 de mayo de 2001). *Maestros del Web*. Obtenido de <http://www.maestrosdelweb.com/phpintro/>
- Hernandez, U. (22 de Febrero de 2016). *codigofacilito.com*. Obtenido de <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>
- Landa, G. U. (2019). *Programación Orientada a Objetos en PHP (Spanish Edition) Kindle Edition*.
- McGraw-Hill. (2020). *McGraw-Hill*. Obtenido de <https://www.mheducation.es/bcv/guide/capitulo/8448148797.pdf>
- MDN contributors. (15 de Septiembre de 2020). *MDN Web Docs*. Obtenido de [https://developer.mozilla.org/es/docs/Learn/HTML/Forms/Your\\_first\\_HTML\\_form](https://developer.mozilla.org/es/docs/Learn/HTML/Forms/Your_first_HTML_form)
- Merino, J. P. (2013). *definicion.de*. Obtenido de <https://definicion.de/sitio-web/>
- PHP Group. (2020). *php.net*. Obtenido de <https://www.php.net/manual/es/language.operators.assignment.php>
- PHP Group. (2020). *Php.net*. Obtenido de <https://www.php.net/manual/es/language.operators.assignment.php>
- Raffino, M. E. (2020). *Concepto.de*. Obtenido de <https://concepto.de/servidor-web/#ixzz6eNayExcY>
- Robles, V. (2014). *victorroblesweb.es*. Obtenido de <https://victorroblesweb.es/2013/11/18/tutorial-mvc-en-php-nativo/>
- Rodriguez, N. (2015). *exa.unne.edu.ar*. Obtenido de [http://exa.unne.edu.ar/informatica/SO/Adcripcion\\_Nelson\\_DAD.pdf](http://exa.unne.edu.ar/informatica/SO/Adcripcion_Nelson_DAD.pdf)
- Rollet, O. (2015). *Aprender a desarrollar un sitio web con Php y Mysql*. Barcelona: ENI.
- Steve, P. (2016). *Learn PHP 7*. Apress.
- Universidad de Alicante. (2020). *Universidad de Alicante*. Obtenido de <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
- uniwebsidad. (2020). *uniwebsidad*. Obtenido de <https://uniwebsidad.com/libros/jobeeet-1-4/capitulo-4/la-arquitectura-mvc?from=librosweb>



## **ANEXOS**

# Manual Técnico

## ÍNDICE DE CONTENIDOS

INTRODUCCIÓN .....	109
1. PUBLICACIÓN DEL SITIO WEB.....	110
2. RESTAURACIÓN DE LA BASE DE DATOS .....	112

## ÍNDICE DE FIGURAS

<b>Figura 1:</b> Ingreso al CPanel.....	110
<b>Figura 2:</b> Página Principal de CPanel.....	111
<b>Figura 3:</b> Pantalla de Administrador de Archivos .....	111
<b>Figura 4:</b> Pantalla de Subida de Archivos .....	112
<b>Figura 5:</b> Pantalla de Administración de la Base de Datos .....	113
<b>Figura 6:</b> Pantalla de Administración de phpMyAdmin.....	113
<b>Figura 7:</b> Pantalla de Importación de la Base de Datos.....	114
<b>Figura 8:</b> Pantalla de Administración de phpMyAdmin .....	114
<b>Figura 9:</b> Pantalla de Administración de Bases de Datos.....	115

## **INTRODUCCIÓN**

El presente documento tiene como objetivo describir la forma como se realizó la publicación en el Ambiente de Producción del sistema “Gestión de Tickets de Reparación” en un Hosting Web.

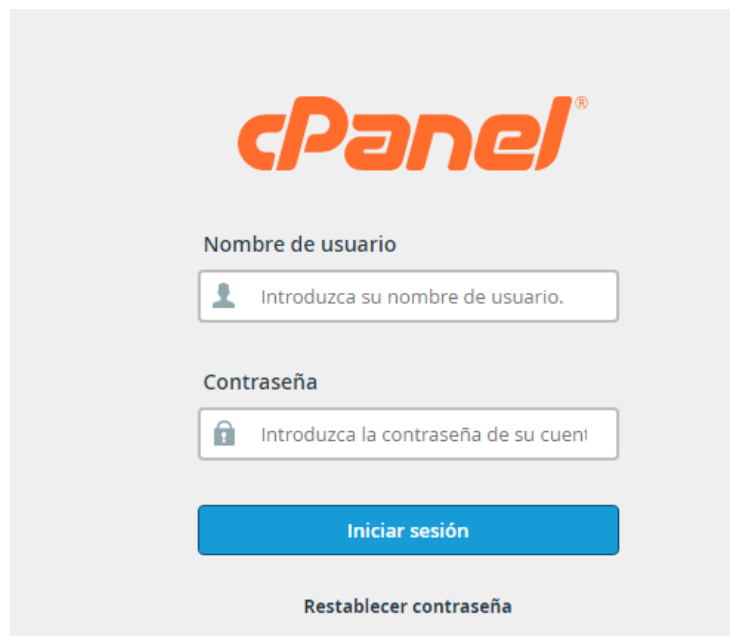
El sitio web que fue implementado y probado localmente pero tienen la necesidad de ser publicado en la web a través de un dominio y hosting junto con todos los archivos relacionados como pueden ser imágenes, archivos html, archivos php, etc. La publicación del sitio queda disponible para todas las personas que lo necesiten, el único requerimiento es tener una conexión a internet.

## PUBLICACIÓN DEL SITIO WEB

Para realizar la publicación del aplicativo web en ambiente de producción se debe ingresar a la siguiente dirección:

<https://www.arrobapcriobamba.com/cpanel>

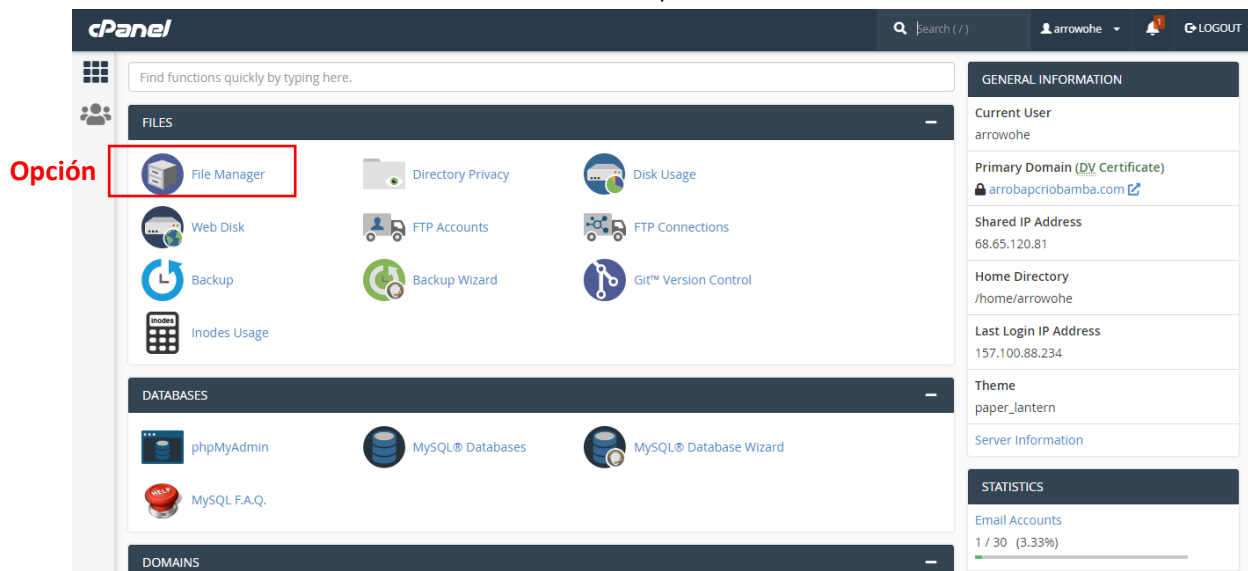
Una vez que se ingresa a la url antes mencionada, se muestra una pantalla con un login donde se debe ingresar el Nombre de Usuario y Contraseña del Hosting (*Figura 44*).



**Figura 44: Ingreso al CPanel**

Fuente: Autor

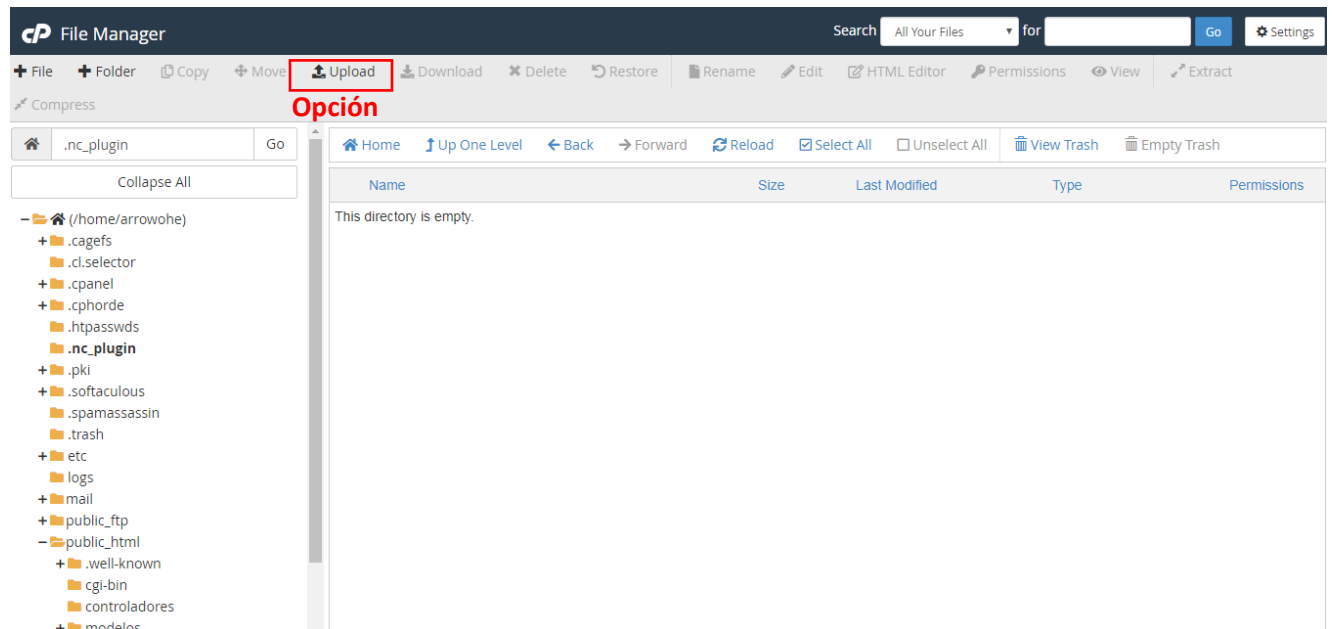
Cuando se ha ingresado correctamente los datos se presenta una pantalla de inicio, donde es el menú principal (*Figura 45*).



**Figura 45: Página Principal de CPanel**

Fuente: Autor

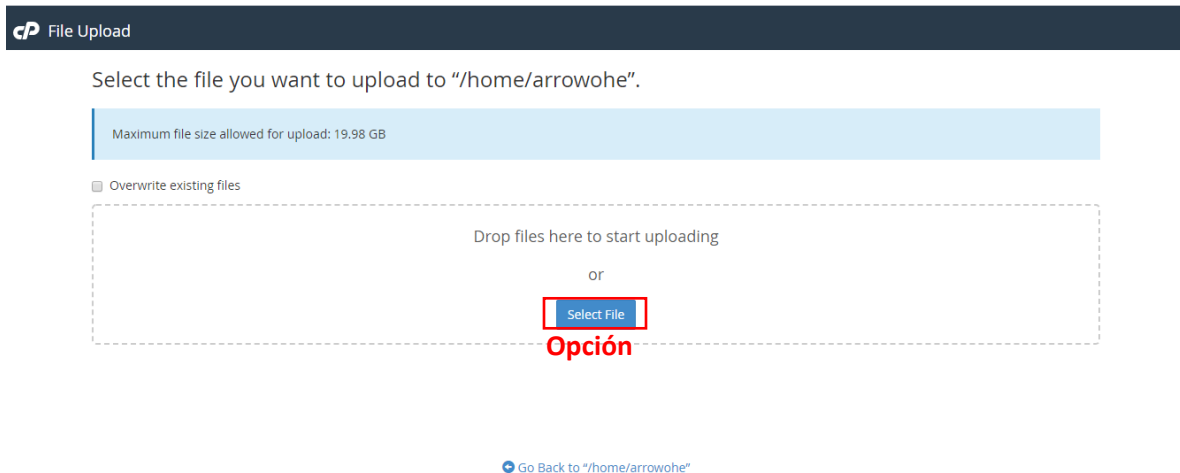
A continuación de la pestaña “Files”, se escoge la opción “File Manager” donde aparece la siguiente pantalla (Figura 46). En esta pantalla se puede subir o bajar todos los archivos que la aplicación web que se encuentren publicados.



**Figura 46: Pantalla de Administrador de Archivos**

Fuente: Autor

Con la opción “*Upload*” se puede subir los archivos que se necesiten. Al presionar sobre la opción “*Upload*” se muestra la siguiente pantalla (*Figura 47*).



**Figura 47: Pantalla de Subida de Archivos**

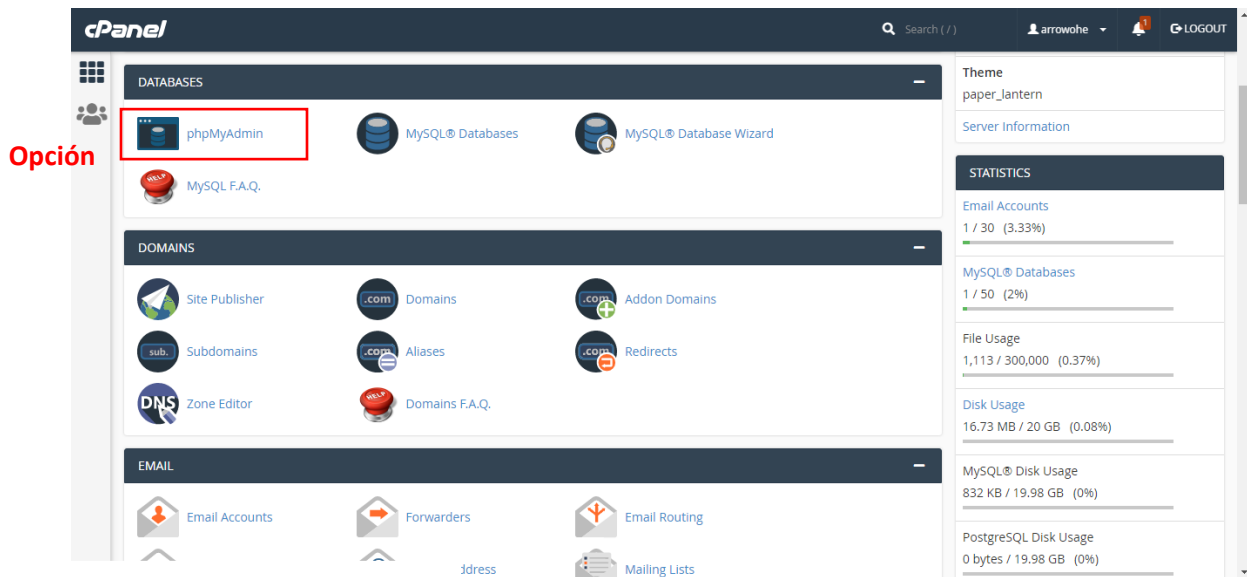
Fuente: Autor

Una vez que se haya ingresado a esta página debe presionar sobre la opción “*Select File*”, donde se presentará una nueva pantalla donde permitirá subir todos los archivos referentes al sitio.

## RESTAURACIÓN DE LA BASE DE DATOS

Para la restauración de la base de datos del sitio web, se debe escoger de la pestaña “*Databases*” y escoger la opción “*phpMyAdmin*” (*Figura 48*).

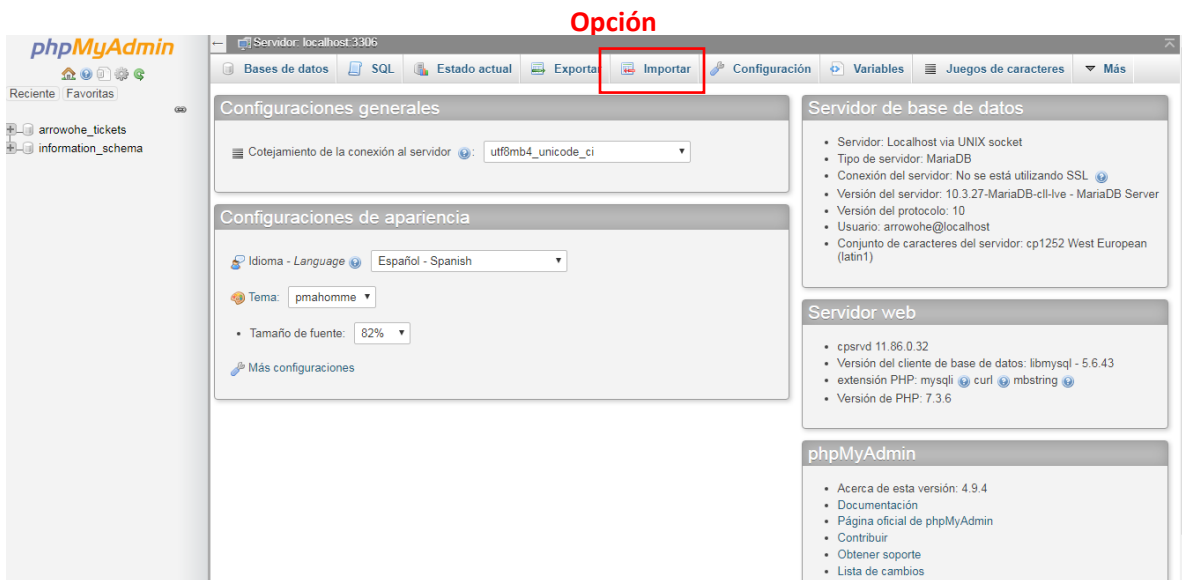




**Figura 48: Pantalla de Administración de la Base de Datos**

Fuente: Autor

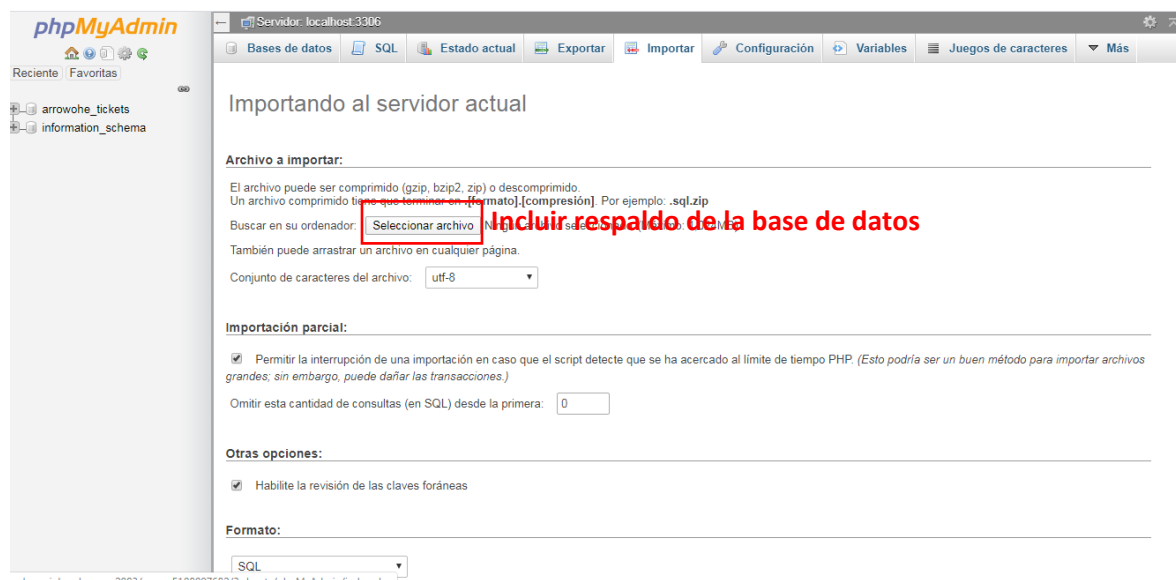
Al presionar sobre la opción “phpMyAdmin” se presentará una pantalla como se muestra a continuación (Figura 49).



**Figura 49: Pantalla de Administración de phpMyAdmin**

Fuente: Autor

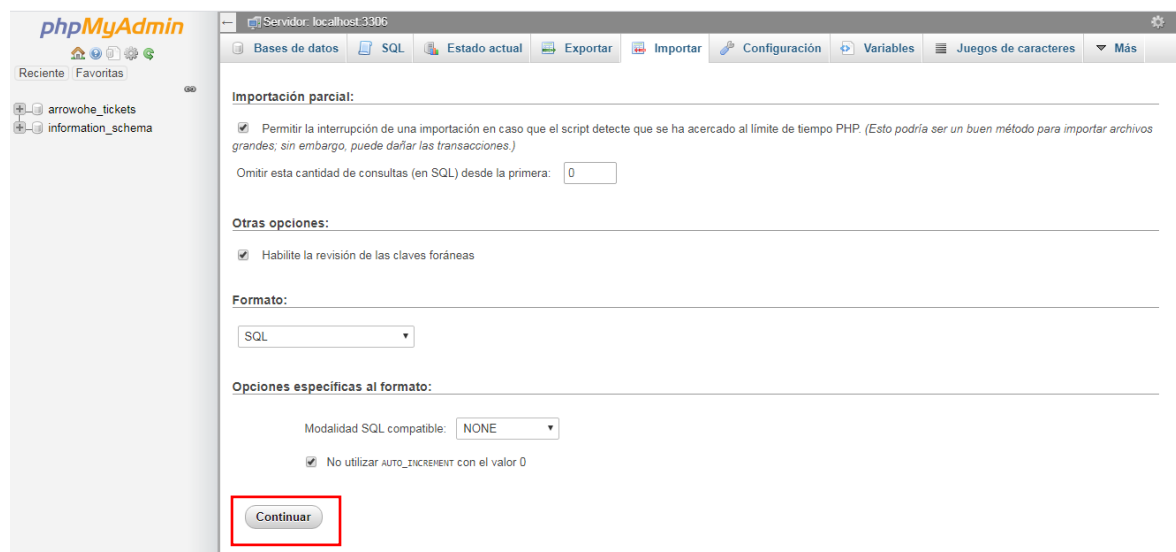
Se escoge la opción **“Importar”** como se muestra la pantalla (**Figura 50**), donde se presiona la opción **“Seleccionar archivo”** y aparece la ventana para cargar el archivo de la base de datos a restaurar.



**Figura 50: Pantalla de Importación de la Base de Datos**

Fuente: Autor

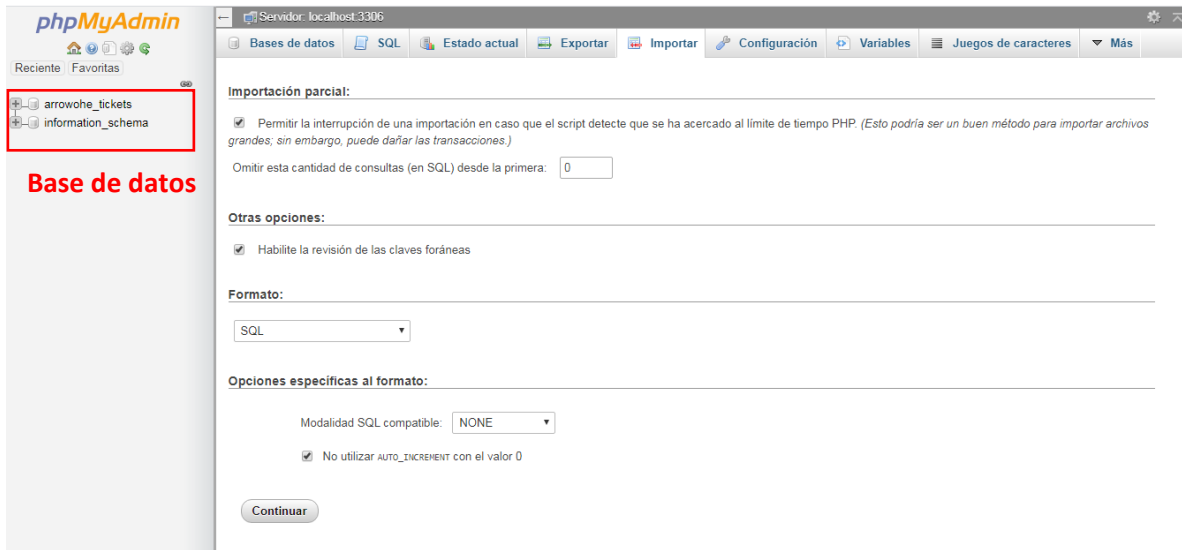
Una vez que el archivo sea el correcto, se debe presionar sobre el botón **“Continuar”** que se encuentra en la parte inferior de la pantalla (**Figura 51**).



**Figura 51: Pantalla de Administración de phpMyAdmin**

Fuente: Autor

Una vez hecho este proceso se habrá realizado la restauración de la base de datos de forma correcta, se podrá visualizarse la misma en la parte izquierda del formulario de phpMyAdmin (*Figura 52*).



**Figura 52: Pantalla de Administración de Bases de Datos**

Fuente: Autor

# Manual de Usuario

## ÍNDICE GENERAL

INTRODUCCIÓN	121
1. APLICACIÓN WEB	121
2. TIPOS DE USUARIOS	122
3. GESTIÓN DE TICKETS DE REPARACION	123
4. ROL TÉCNICO	126
4.1. Mis Tickets	126
5. ROL RECEPCIONISTA	130
5.1. Clientes	130
5.2. Equipos	134
5.3. Ingreso de ticket de reparación	138
5.4. Reportes	144
6. ROL ADMINISTRADOR	146
6.1. Usuarios	146

## INDICE DE FIGURAS

<b>Figura 1:</b> Página de Inicio	122
<b>Figura 2:</b> Inicio de Sesión	123
<b>Figura 3:</b> Pantalla Bienvenida	124
<b>Figura 4:</b> Cambio Contraseña	124
<b>Figura 5:</b> Formulario Cambiar Contraseña	125
<b>Figura 6:</b> Cerrar de sesión	125
<b>Figura 7:</b> Menú Técnico	126
<b>Figura 8:</b> Listado de Tickets de Reparación - Rol Técnico	126
<b>Figura 9:</b> Diagnosticar Ticket - Rol Técnico	127
<b>Figura 10:</b> Formato del Email de Diagnóstico de Equipo - Rol Técnico	128
<b>Figura 11:</b> Reparación Ticket - Rol Técnico	128
<b>Figura 12:</b> Detalle de ticket solo lectura - Rol Técnico	129
<b>Figura 13:</b> Pantalla Bienvenida - Rol Recepcionista	130
<b>Figura 14:</b> Listado de Clientes - Rol Recepcionista	130
<b>Figura 15:</b> Opción Nuevo Cliente - Rol Recepcionista	131
<b>Figura 16:</b> Nuevo Cliente - Rol Recepcionista	131
<b>Figura 17:</b> Opción Editar Cliente - Rol Recepcionista	132
<b>Figura 18:</b> Editar Cliente - Rol Recepcionista	133
<b>Figura 19:</b> Opción Eliminar Cliente - Rol Recepcionista	134
<b>Figura 20:</b> Mensaje de Confirmación de Eliminación de Clientes - Rol Recepcionista	134

<b>Figura 21:</b> Listado de Equipos - Rol Recepcionista	135
<b>Figura 22:</b> Opción Nuevo Equipo - Rol Recepcionista	135
<b>Figura 23:</b> Nuevo Equipo - Rol Recepcionista	135
<b>Figura 24:</b> Opción Editar Equipo - Rol Recepcionista	136
<b>Figura 25:</b> Editar Equipo - Rol Recepcionista	137
<b>Figura 26:</b> Opción Eliminar Equipo - Rol Recepcionista	137
<b>Figura 27:</b> Mensaje de Confirmación de Eliminación de Equipos - Rol Recepcionista	138
<b>Figura 28:</b> Listado de tickets de reparación - Rol Recepcionista	138
<b>Figura 29:</b> Opción Nuevo Ticket Reparación- Rol Recepcionista	139
<b>Figura 30:</b> Nuevo Ticket de Reparación- Rol Recepcionista	139
<b>Figura 31:</b> Nuevo Ticket de Reparación Cliente registrado - Rol Recepcionista	139
<b>Figura 32:</b> Nuevo Ticket de Reparación Cliente no Registrado- Rol Recepcionista	140
<b>Figura 33:</b> Nuevo Ticket de Reparación Equipo no Registrado- Rol Recepcionista	140
<b>Figura 34:</b> Nuevo Ticket de Reparación Completo- Rol Recepcionista	141
<b>Figura 35:</b> Formato del Email de confirmación enviado al Cliente	141
<b>Figura 36:</b> Vista del Cliente del Ticket	142
<b>Figura 37:</b> Impresión de nuevo Ticket - Rol Recepcionista	142
<b>Figura 38:</b> Opción Entrega de Equipo - Rol Recepcionista	143
<b>Figura 39:</b> Mensaje de Confirmación de Eliminar Ticket - Rol Recepcionista	144
<b>Figura 40:</b> Reporte de Tickets de Reparación - Rol Recepcionista	144
<b>Figura 41:</b> Reporte de Tickets de Reparación Vista de impresión - Rol Recepcionista	145
<b>Figura 42:</b> Pantalla de Bienvenida – Rol Administrador	146

<b>Figura 43:</b> Listado de Usuarios - Rol Administrador	146
<b>Figura 44:</b> Opción Nuevo Usuario - Rol Administrador	147
<b>Figura 45:</b> Nuevo Usuario - Rol Administrador	147
<b>Figura 46:</b> Opción Editar Usuario - Rol Administrador	148
<b>Figura 47:</b> Editar Usuario - Rol Administrador	149
<b>Figura 48:</b> Opción Eliminar Usuario - Rol Administrador	150
<b>Figura 49:</b> Mensaje de Confirmación de Eliminación de Usuarios - Rol Administrador	150



# INTRODUCCIÓN

La aplicación Web de Control de Tickets de Reparación permite llevar un registro de las reparaciones que la empresa @PC gestiona, mediante lo cual tendrá un mejor control de las reparaciones que atiende de sus clientes en un periodo determinado. La aplicación ha sido desarrollada utilizando PHP orientado a Objetos usando un patrón de diseño MVC acorde con los requerimientos presentado por la empresa.

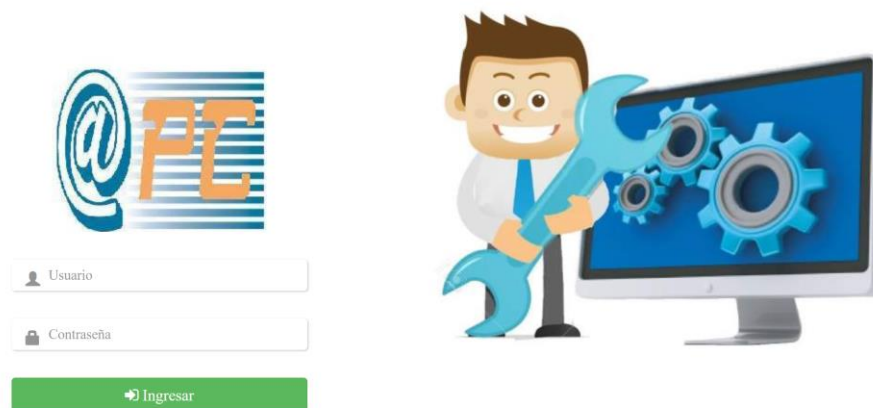
El presente documento tiene como finalidad detallar de manera clara y concisa el funcionamiento del aplicativo Web y el proceso de generación de tickets de reparación, de acuerdo al rol de cada usuario.

Se puede acceder a la aplicación a través del siguiente enlace web:

<https://arrobapcriobamba.com/>

## 1. APLICACIÓN WEB

La aplicación web implementada permite gestionar los tickets de reparación de equipos de cómputo que ingresan a la institución @PC. Al momento de ingresar a la dirección <https://arrobapcriobamba.com/> presenta la pantalla principal (*Figura 53*).



**Figura 53: Página de Inicio**

Fuente: Autor

## 2. TIPOS DE USUARIOS

- **Administrador:** Es la persona que se encarga de administrar el aplicativo Web, así como el manejo de los usuarios, gestión de los clientes, sus equipos, la administración de todas los tickets de reparación y la generación de reportes.
- **Recepcionista:** Es la persona que se encarga de registrar los clientes y los equipos a reparar y además generar los tickets de reparación, y por ultimo hacer la entrega del equipo una vez reparado.
- **Técnico:** Son las personas que atienden los tickets de reparación generados por el usuario recepcionista, son los encargados de diagnosticar y posterior a la confirmación del cliente realizar la reparación.

### 3. GESTIÓN DE TICKETS DE REPARACION

La funcionalidad de gestión de tickets de reparación es un módulo que permite gestionar las reparaciones que ingresan a soporte técnico de la empresa @PC, en la siguiente ilustración (*Figura 54*) se presenta la pantalla de acceso al sistema para ingresar las credenciales de acceso.

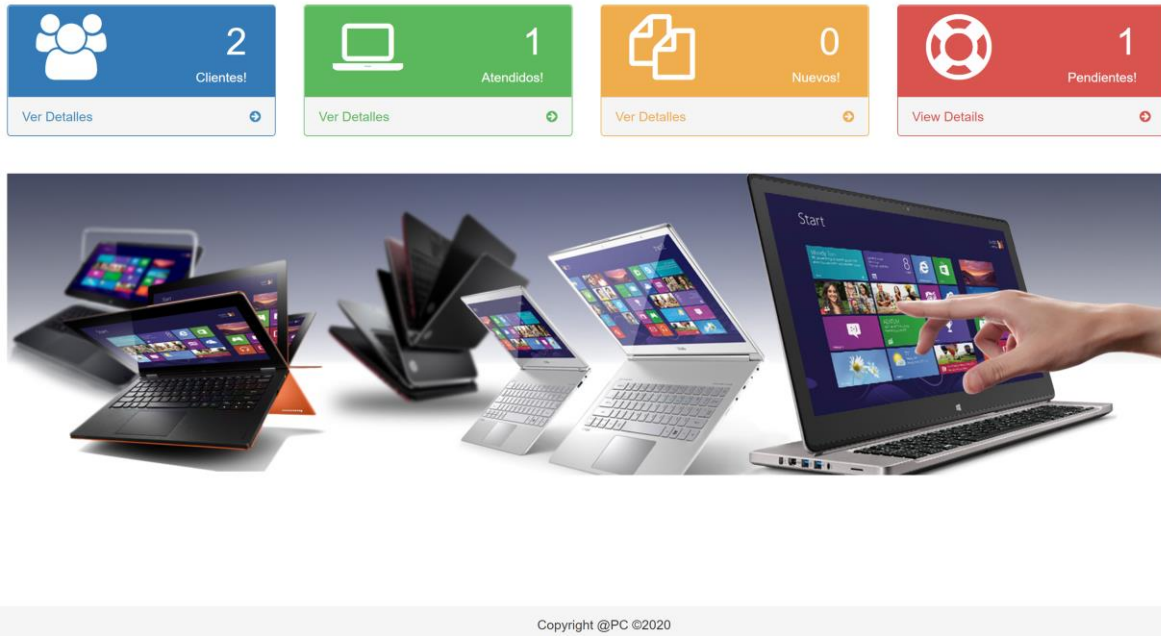


**Figura 54:** Inicio de Sesión

**Fuente:** Autor

El sistema validará los datos ingresados y permitirá el ingreso a la aplicación en dependencia del Tipo de Usuario y permitirá ejecutar las acciones que le corresponde a cada usuario; en caso de existir algún error en los datos digitados, se mostrará los respectivos mensajes para su respectiva corrección.

Al ingresar al sistema presentara la pantalla de bienvenida con las siguientes opciones (*Figura 55*).



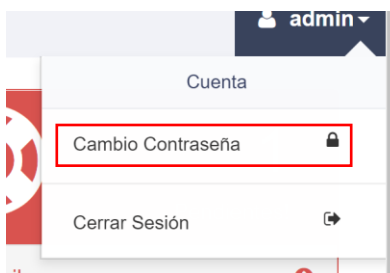
**Figura 55:** Pantalla Bienvenida

**Fuente:** Autor

Opciones de pantalla.

- Clientes
- Tickets Atendidos
- Tickets Nuevos
- Tickets Pendientes

El usuario puede tener la opción de cambio de su contraseña para los cual se debe presionar sobre el enlace “Cambio contraseña” que se encuentra en menú superior derecho. (*Figura 56*)



**Figura 56:** Cambio Contraseña

**Fuente:** Autor

El sistema mostrará un formulario donde el usuario debe ingresar la contraseña anterior y la nueva contraseña, en caso de existir algún error de digitación el sistema mostrará al usuario el respectivo mensaje de error. (*Figura 57*)

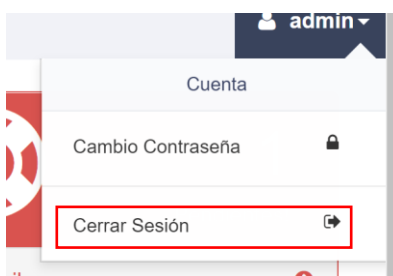
Cambiar Contraseña

Contraseña Actual	Nueva Contraseña	Repetir Contraseña
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Cambiar Contraseña"/>		

**Figura 57:** Formulario Cambiar Contraseña

**Fuente:** Autor

Para salir de la aplicación se debe presionar sobre el enlace “Cerrar Sesión” que se encuentra en menú superior derecho. (*Figura 58*)

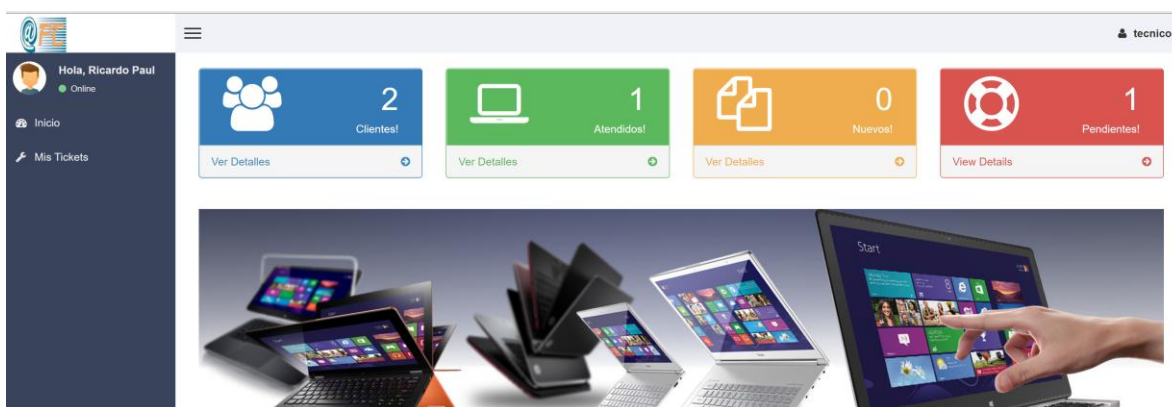


**Figura 58:** Cerrar de sesión

**Fuente:** Autor

## 4. ROL TÉCNICO

Cuando el usuario con rol Técnico ingresa al sistema se presenta el siguiente menú principal en la pantalla de bienvenida (*Figura 59*) donde el técnico puede en cada opción hacer las siguientes opciones:



**Figura 59:** Menú Técnico







Fuente: Autor

### 4.1. Mis Tickets

En esta opción el usuario con rol Técnico puede visualizar el listado de tickets a él asignados y registrados en el sistema y puede hacer una búsqueda ingresando un texto de cualquier columna. (*Figura 60*)

Tickets Reparaciones


**Búsqueda**

Ticket	Cliente	Equipo	Motivo de Ingreso	Fecha Ingreso	Estado	Acciones
TR-0001	William Adriano	PORTATIL	NO SE PRENDE Y DESPUES DE TANTO INTENTAR REFLEJA CON PANTALLASO AZUL	2020-12-12	Entregado	  
TR-0002	Ana Cervantes	PORTATIL	Se reinicia a cada momento	2020-12-16	Diagnóstico	  

Anterior **1** Siguiente

**Figura 60:** Listado de Tickets de Reparación - Rol Técnico

Fuente: Autor

Diagnosticar Ticket En esta opción  el usuario con rol técnico puede visualizar el detalle del ticket y le permite diagnosticar el problema que tiene el equipo a reparar, el usuario puede hacer un comentario de diagnóstico y además indicar el costo de la reparación. **(Figura 61)**

### Reparación de Equipo x

		<b>No. Ticket:</b>	<b>TR-0002</b>
Cliente:	Ana Cervantes		
Equipo:	PORTATIL	Serie:	123456789
Marca:	DELL	Modelo:	INSPIRON
Observación Equipo:	PORTATIL COLOR AZUL CON LOGO MARCA DELL		
Técnico	Ricardo Paul Manzano Peñafiel	Fecha Ingreso:	2020-12-16
Motivo de Ingreso:	Se reinicia a cada momento		
Costo Reparacion:	15.00 USD		

**Historial de Reparación**

Estado	Fecha	Técnico	Observaciones
Ingreso	2020-12-17	Fernanda Manzano	Se reinicia a cada momento

**Diagnostico del Equipo**

El equipo necesita formateo

**Costo de la reparacion**

20

Guardar

**Figura 61:** Diagnosticar Ticket - Rol Técnico

**Fuente:** Autor

Una vez que el usuario técnico guarde el diagnostico se envía un email al cliente donde le comentan que el cliente puede revisar el diagnostico de su equipo y que debe contactarse para confirmar si desea proceder con la reparación. Se envía además un código QR para que el cliente pueda ver los detalles del ticket en la aplicación.

127

Hola Ana Cervantes

Su ticket ha sido Diagnosticado en nuestro sistema. Por favor comuníquese con nosotros para proceder con la Reparación de su Equipo. Si desea saber el estado de su ticket por favor escanee el siguiente código.




Atentamente



Figura 62: Formato del Email de Diagnóstico de Equipo - Rol Técnico

Fuente: Autor

**Reparar Ticket** En esta opción  el usuario con rol técnico puede visualizar el detalle del ticket y le permite realizar la reparación del equipo, el usuario puede hacer un comentario de reparación. (Figura 63)

### Reparación de Equipo

		No. Ticket:	TR-0002
Cliente:	Ana Cervantes		
Equipo:	PORTATIL	Serie:	123456789
Marca:	DELL	Modelo:	INSPIRON
Observación Equipo:	PORTATIL COLOR AZUL CON LOGO MARCA DELL		
Técnico	Ricardo Paul Manzano Peñafiel	Fecha Ingreso:	2020-12-16
Motivo de Ingreso:	Se reinicia a cada momento		
Costo Reparación:	20.00 USD		

#### Historial de Reparación

Estado	Fecha	Técnico	Observaciones
Ingreso	2020-12-17	Fernanda Manzano	Se reinicia a cada momento
Diagnóstico	2020-12-20	Ricardo Paul Manzano Peñafiel	El equipo necesita formateo

#### Reparación del Equipo

El equipo se ha foramteado se han instaldo programas utilitarios


**Guardar**

Figura 63: Reparación Ticket - Rol Técnico

Fuente: Autor

Una vez que el usuario técnico guarde la reparación el ticket está listo para la entrega al cliente por parte del usuario Recepcionista.



**Ver Ticket** En esta opción  el usuario con rol Técnico puede visualizar el Ticket en solo lectura y le permitirá también poder reimprimirlo, el ticket contiene el historial de las etapas en las que el ticket ha pasado. (Figura 64)

Ticket de Reparación		No. Ticket:	TR-0001
Cliente:	William Adriano		
Equipo:	PORTATIL	Serie:	123456789
Marca:	DELL	Modelo:	INSPIRON
Observación Equipo:	PORTATIL COLOR AZUL CON LOGO MARCA DELL		
Técnico a Cargo	Ricardo Paul Manzano Peñafiel	Fecha Ingreso:	2020-12-12
Motivo de Ingreso:	NO SE PRENDE Y DESPUES DE TANTO INTENTAR REFLEJA CON PANTALLASO AZUL		
Costo Reparacion:	0.00 USD		
Usuario Entrega	Ricardo Paul Manzano Peñafiel	Fecha Entrega:	2020-12-12
Solución:	SE PROCEDE A ENTREGAR AL SR. WILLIAM ADRIANO PROPIETARIO DE LA LAPTON DELL		

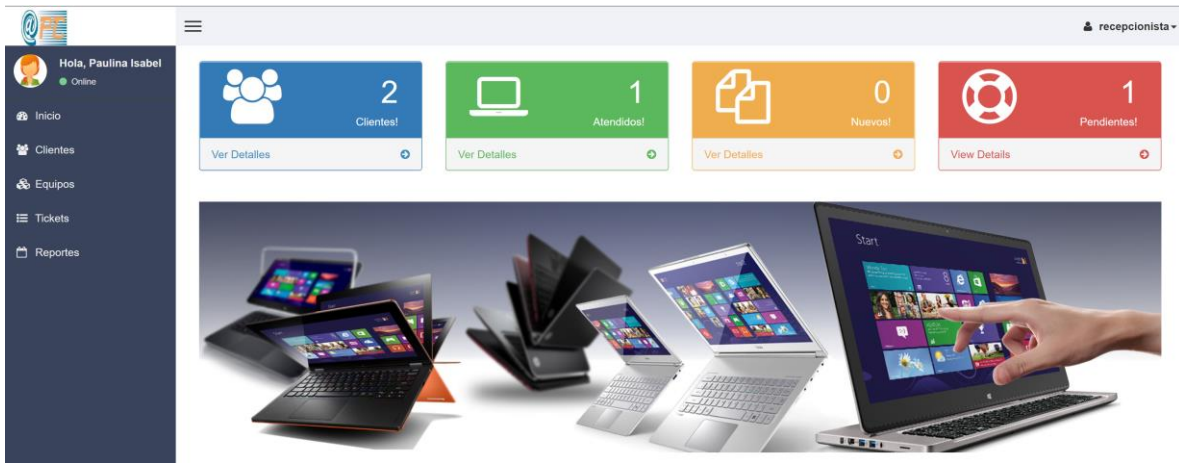
Historial de Reparación			
Estado	Fecha	Técnico	Observaciones
Ingreso	2020-12-12	Paulina Isabel Manzano Calderon	NO SE PRENDE Y DESPUES DE TANTO INTENTAR REFLEJA CON PANTALLASO AZUL
Diagnóstico	2020-12-12	Ricardo Paul Manzano Peñafiel	LA MAQUINA DEBE SER REEMPLAZADA LA RAM
Reparación	2020-12-12	Ricardo Paul Manzano Peñafiel	EL EQUIPO SE ENCUNTRA REPARADO EN SU TOTALIDAD FUNCIONANDO LISTO PARA SER ENTREGADO AL CLIENTE
Entregado	2020-	Paulina Isabel	SE PROCEDE A ENTREGAR AL SR. WILLIAM ADRIANO PROPIETARIO DE LA

**Figura 64:** Detalle de ticket solo lectura - Rol Técnico

**Fuente:** Autor

## 5. ROL RECEPCIONISTA

Cuando el usuario con rol Recepcionista ingresa al sistema se muestra el siguiente menú principal en la pantalla de bienvenida (*Figura 65*) donde el usuarios recepcionista puede en cada opción hacer lo siguiente:



**Figura 65:** Pantalla Bienvenida - Rol Recepcionista

Fuente: Autor

### 5.1. Clientes





En esta opción el usuario con rol Recepcionista puede visualizar el listado de clientes registrados en el sistema y puede hacer una búsqueda ingresando cualquier texto de las columnas. (*Figura 66*)

#### Clientes

Añadir

**Búsqueda**

Buscar:

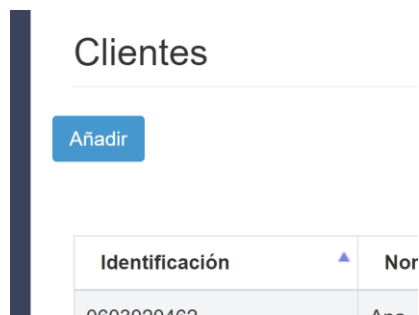
Identificación	Nombres	Apellidos	Email	Teléfonos	Acciones
0603020462	Ana	Cervantes	mf.manzano17@gmail.com	9999999999 -	 
0604796581	William	Adriano	gwiliam.adriano@gmail.com	0999275406 - 032969676	 

Anterior 1 Siguiente

**Figura 66:** Listado de Clientes - Rol Recepcionista

Fuente: Autor

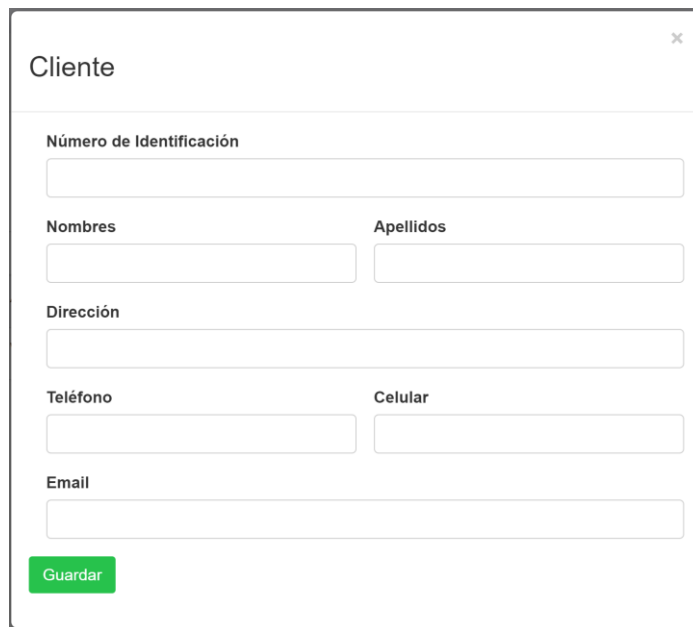
Para crear un nuevo cliente se debe presionar sobre el botón “Añadir”, ubicado en la parte superior izquierda (*Figura 67*)



**Figura 67:** Opción Nuevo Cliente - Rol Recepcionista

**Fuente:** Autor

El Sistema presenta una pantalla con un formulario que tiene los siguientes campos de información (*Figura 68*):

The image shows a modal window titled 'Cliente'. It contains a form with the following fields: 'Número de Identificación', 'Nombres', 'Apellidos', 'Dirección', 'Teléfono', 'Celular', and 'Email'. At the bottom left of the form is a green button labeled 'Guardar'.

**Figura 68:** Nuevo Cliente - Rol Recepcionista

**Fuente:** Autor

Los campos presentados anteriormente deben ser ingresados de forma obligatoria, caso contrario se presentarán los mensajes de alerta. Cada uno de los campos debe ser ingresado como se detalla a continuación.

- Numero de Identificación: Se debe Ingresar la Cedula del Cliente.
- Nombres: Se debe ingresar el Nombre del Cliente.
- Apellidos: se debe ingresar el Apellido del Cliente.
- Dirección: Se debe ingresar la Dirección del Cliente.
- Teléfono: Se ingresa el teléfono del Cliente.
- Celular: Se debe ingresar el Celular del Cliente
- Email: Se debe ingresar el Email del Cliente

Una vez que haya sido ingresada toda la información se debe presionar sobre el botón “**Guardar**”, si desea almacenar los datos del Cliente caso contrario presionar el botón “**X**” para salir sin guardar nada ubicado en la parte superior derecha de la pantalla.

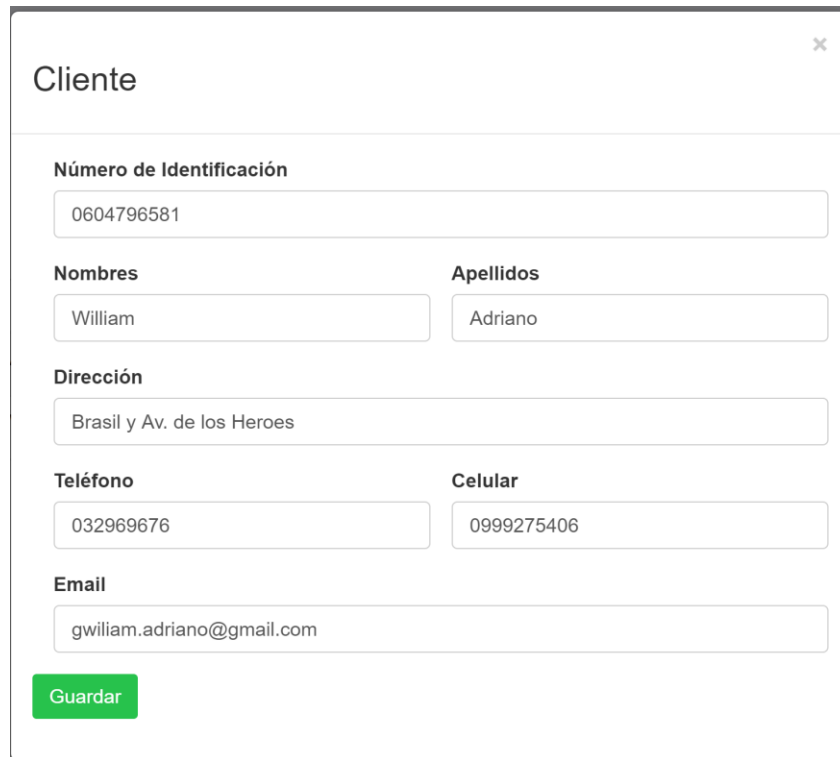
**Para editar los datos de un Cliente** se debe presionar sobre el botón editar, ubicado en cada una de los clientes del listado en la columna de Acciones (*Figura 69*).



**Figura 69:** Opción Editar Cliente - Rol Recepcionista

**Fuente:** Autor

Al presionar sobre la opción “**Editar**”, el sistema presentará una página con un formulario con todos los datos que podrían ser editados (*Figura 70*):



Cliente

Número de Identificación  
0604796581

Nombres: William      Apellidos: Adriano

Dirección  
Brasil y Av. de los Heroes

Teléfono: 032969676      Celular: 0999275406

Email  
gwilliam.adriano@gmail.com

Guardar

**Figura 70:** Editar Cliente - Rol Recepcionista

**Fuente:** Autor

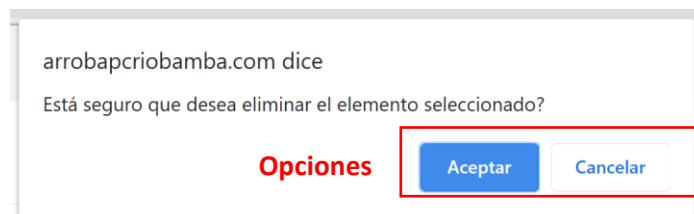
Una vez que los datos hayan sido modificados se debe presionar sobre el botón “**Guardar**” y automáticamente si están correctamente ingresados se guardarán los datos, caso contrario se emitirán mensajes de error los cuales deberían ser corregidos para continuar con la modificación de los datos. En el caso que no se requiera realizar ningún cambio solo se debe presionar sobre el botón “**X**” y se regresará a la pantalla de listado de Clientes.

**Para eliminar un Cliente** se debe presionar sobre el botón “Eliminar”, ubicado en cada una de los clientes que se encuentran en el listado en la columna Acciones (**Figura 71**).



**Figura 71:** Opción Eliminar Cliente - Rol Recepcionista  
**Fuente:** Autor

Una vez que se haya presionado sobre el botón el eliminar saldrá un mensaje de confirmación si desea realizar la eliminación del Cliente. Al presionar sobre el botón “*Aceptar*” automáticamente se borrará el cliente seleccionado (*Figura 72*).



**Figura 72:** Mensaje de Confirmación de Eliminación de Clientes - Rol Recepcionista  
**Fuente:** Autor

## 5.2. Equipos

En esta opción el usuario con rol Recepcionista puede visualizar el listado de los equipos a reparar registrados en el sistema y puede hacer una búsqueda por cualquier texto de las columnas. (*Figura 73*)

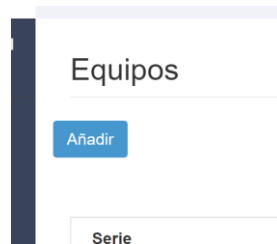
## Equipos

Serie	Nombre Equipo	Marca	Modelo	Acciones
123456789	PORTATIL	DELL	INSPIRON	

**Figura 73:** Listado de Equipos - Rol Recepcionista

**Fuente:** Autor

**Para crear un nuevo equipo** se debe presionar sobre el botón “Añadir”, ubicado en la parte superior izquierda (*Figura 74*)



**Figura 74:** Opción Nuevo Equipo - Rol Recepcionista

**Fuente:** Autor

El Sistema presenta una página con un formulario que tiene los siguientes campos de información (*Figura 75*):

Equipo

Número de Serie

Nombre Equipo

Marca

Modelo

Observación del Equipo

Guardar

**Figura 75:** Nuevo Equipo - Rol Recepcionista

**Fuente:** Autor

Los campos mostrados anteriormente deben ser ingresados de forma obligatoria, caso contrario se presentarán los mensajes de alerta. Cada uno de los campos debe ser ingresado como se detalla a continuación.

- Número de Serie: Se debe ingresar el número de serie del equipo.
- Nombre Equipo: Se debe ingresar el Nombre del Equipo.
- Marca: Se debe ingresar la Marca del Equipo.
- Modelo: Se debe ingresar el Modelo del Equipo.
- Observación: Se debe ingresar una observación del Equipo.

Una vez que haya sido ingresada toda la información se debe presionar sobre el botón “**Guardar**” si desea almacenar los datos del Equipo caso contrario presionar el botón “**X**” de la parte superior derecha de la pantalla para salir sin guardar nada.

**Para editar los datos de un Equipo** se debe presionar sobre el botón editar, ubicado en cada una de los equipos del listado en la columna de Acciones (*Figura 76*).

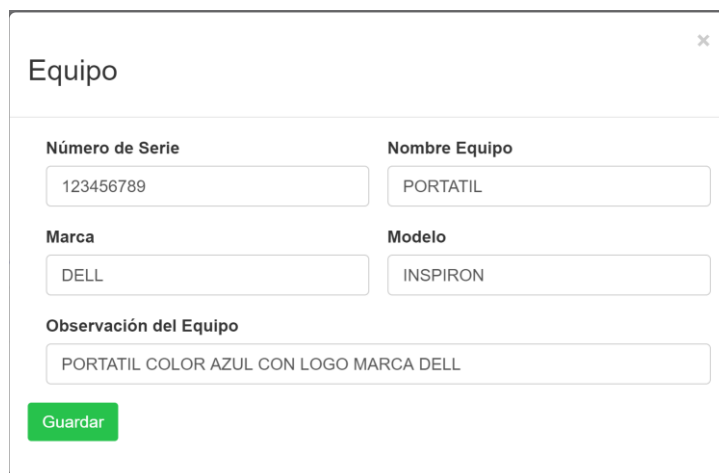


**Figura 76:** Opción Editar Equipo - Rol Recepcionista

**Fuente:** Autor

Al presionar sobre la opción “*Editar*”, el sistema presentará una pantalla con un formulario con todos los datos que podrían ser editados (*Figura 77*):



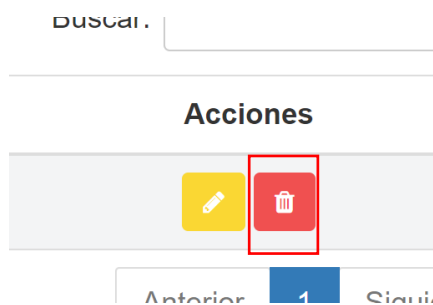
El formulario muestra un campo de texto para el número de serie con el valor '123456789', un campo para el nombre del equipo con el valor 'PORTATIL', un campo para la marca con el valor 'DELL' y un campo para el modelo con el valor 'INSPIRON'. Debajo de estos campos hay un campo de observación con el texto 'PORTATIL COLOR AZUL CON LOGO MARCA DELL'. En la parte inferior izquierda del formulario hay un botón verde con el texto 'Guardar'.

**Figura 77:** Editar Equipo - Rol Recepcionista

**Fuente:** Autor

Una vez que los datos hayan sido modificados se debe presionar sobre el botón “**Guardar**” y automáticamente si están correctamente ingresados se guardarán los datos, caso contrario se emitirán mensajes de error los cuales deberían ser corregidos para continuar con la modificación de los datos. En el caso que no se requiera realizar ningún cambio solo se debe presionar sobre el botón “X” y se regresará a la pantalla de listado de Equipos.

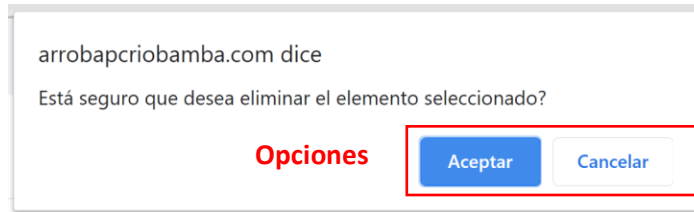
**Para eliminar un Equipo** se debe presionar sobre el botón “Eliminar”, ubicado en cada una de los turnos que se encuentran en el listado en la columna Acciones (**Figura 78**).



**Figura 78:** Opción Eliminar Equipo - Rol Recepcionista

**Fuente:** Autor

Una vez que se haya presionado sobre el botón el eliminar saldrá un mensaje de confirmación si desea realizar la eliminación del Equipo. Al presionar sobre el botón “**Aceptar**” automáticamente se borrará el equipo seleccionado (**Figura 79**).



**Figura 79:** Mensaje de Confirmación de Eliminación de Equipos - Rol Recepcionista

Fuente: Autor

### 5.3. Ingreso de ticket de reparación






En esta opción el usuario con rol Recepcionista puede visualizar el listado de los tickets de reparación que los clientes han solicitado, el usuario Recepcionista puede hacer una búsqueda ingresando un texto de cualquier columna. (*Figura 80*)

Tickets Reparaciones

Añadir

Búsqueda

Buscar:

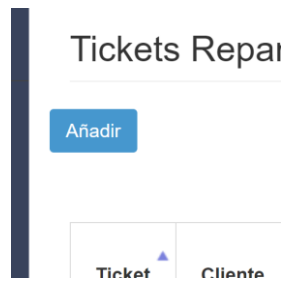
Ticket	Cliente	Equipo	Motivo de Ingreso	Técnico	Fecha Ingreso	Estado	Acciones
TR-0001	William Adriano	PORTATIL	NO SE PRENDE Y DESPUES DE TANTO INTENTAR REFLEJA CON PANTALLASO AZUL	Ricardo Paul Manzano Peñafiel	2020-12-12	Entregado	  
TR-0002	Ana Cervantes	PORTATIL	Se reinicia a cada momento	Ricardo Paul Manzano Peñafiel	2020-12-16	Reparación	  

Anterior 1 Siguiente

**Figura 80:** Listado de tickets de reparación - Rol Recepcionista

Fuente: Autor

**Para crear un nuevo ticket de reservación** se debe presionar sobre el botón “Añadir”, ubicado en la parte superior izquierda (*Figura 81*)



**Figura 81:** Opción Nuevo Ticket Reparación- Rol Recepcionista

**Fuente:** Autor

El sistema presenta una página con un formulario que tiene los siguientes campos de información (**Figura 82**):

## Ingreso Ticket Reparación

The image shows a web form titled 'Ingreso Ticket Reparación'. The form has a section labeled 'Identificación Cliente:' followed by a single-line text input field.

**Figura 82:** Nuevo Ticket de Reparación- Rol Recepcionista

**Fuente:** Autor

El campo solicitado es el Identificación del cliente con lo cual el sistema hará una búsqueda interna mostrando la información almacenada del cliente Acciones (**Figura 83**).

The image shows a web form titled 'Nuevo Ticket de Reparación Cliente registrado'. The form has two main sections. The first section is labeled 'Identificación Cliente:' and contains a text input field with the value '0604796581'. Below this is a section labeled 'Nombre Cliente:' with the value 'William Adriano'. The second section is labeled 'Detalles Equipo a Reparar' and contains a text input field for 'Número de Serie' and a yellow button labeled 'Buscar Equipo'.

**Figura 83:** Nuevo Ticket de Reparación Cliente registrado - Rol Recepcionista

Fuente: Autor

En caso de no encontrar la información del cliente en el sistema se visualizará un botón para ingresar al nuevo cliente Acciones (*Figura 84*).

Ingreso Ticket Reparación



Identificación Cliente: 0603108770

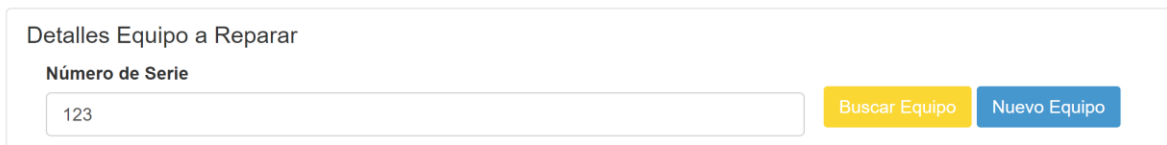
El Cliente No existe! Por favor ingrese un nuevo Cliente

Nuevo Cliente

**Figura 84:** Nuevo Ticket de Reparación Cliente no Registrado- Rol Recepcionista

Fuente: Autor

Una vez cargada la información del cliente en la sección Detalles de equipo solicitará el ingreso del número de serie del equipo a reparar de igual manera permitirá cargar los datos del equipo registrado caso contrario parecer un botón para el nuevo registro del equipo (*Figura 85*).



Detalles Equipo a Reparar

Número de Serie

123

Buscar Equipo

Nuevo Equipo

**Figura 85:** Nuevo Ticket de Reparación Equipo no Registrado- Rol Recepcionista

Fuente: Autor

Posterior a la carga de los datos del equipo el sistema solicitará el ingreso del motivo de la reparación, fecha de ingrese y asignar el ticket a un técnico (*Figura 86*).

Brasil y Av. de los Heroes  
**Teléfonos:**  
032969676 - 0999275406  
**Email:**  
gwilliam.adriano@gmail.com

**Detalles Equipo a Reparar**

**Número de Serie**

**Nombre Equipo**  **Marca**  **Modelo**

**Observación del Equipo**

**Problema a Reparar**

**Fecha Ingreso**  **Técnico Asignado**

Copyright @PC ©2020

**Figura 86:** Nuevo Ticket de Reparación Completo- Rol Recepcionista

**Fuente:** Autor

Una vez que haya sido ingresada toda la información se debe presionar sobre el botón “**Guardar**” para generar el ticket de reparación y empezar el proceso en la aplicación.

Posterior a la generación del ticket de reparación, el sistema envía un email (**Figura 87**) al cliente con un código QR que una vez leído, direcciona al cliente a la aplicación para que pueda ver el estado del ticket (**Figura 88**).

**Hola Willian Adriano**

Un nuevo ticket ha sido registrado en nuestro sistema. Por favor escanee el siguiente código para ver el detalle de su ticket.



Atentamente



**Figura 87:** Formato del Email de confirmación enviado al Cliente

**Fuente:** Autor



[Imprimir](#)

<b>Ticket de Reparación</b>	<b>No. Ticket:</b>	<b>TR-0002</b>
<b>Cliente:</b>		
William Adriano		
<b>Equipo:</b>		
PORTATIL		
<b>Serie:</b>		
123456789		
<b>Marca:</b>		
DELL		
<b>Modelo:</b>		

**Figura 88:** Vista del Cliente del Ticket


**Fuente:** Autor

Al generar el ticket la aplicación genera una pantalla que permite la impresión del ticket generado (*Figura 89*).

https://arrobapcriobamba.com/Reparacion/imprimir/3 - Google Chrome

arrobapcriobamba.com/Reparacion/imprimir/3


[Imprimir](#) [Cerrar](#)




<b>Ticket de Reparación</b>	<b>No. Ticket:</b>	<b>TR-0003</b>
<b>Cliente:</b>	William Adriano	
<b>Equipo:</b>	PORTATIL	<b>Serie:</b> 123456789
<b>Marca:</b>	DELL	<b>Modelo:</b> INSPIRON
<b>Observación Equipo:</b>	PORTATIL COLOR AZUL CON LOGO MARCA DELL	
<b>Técnico a Cargo</b>	Ricardo Paul Manzano Peñafiel	<b>Fecha Ingreso:</b> 2020-12-20
<b>Motivo de Ingreso:</b>	Portatil no enciende	
<b>Costo Reparacion:</b>	0.00 USD	

**Figura 89:** Impresión de nuevo Ticket - Rol Recepcionista

Fuente: Autor

**Ver Ticket** esta opción  se presenta de manera similar a la del usuario técnico y permite visualizar toda la información del ticket junto con la opción imprimir (*Figura 64*).

**Entregar Equipo** esta opción  permite realizar el proceso de entrega del equipo una vez terminada la reparación ahí permite registrar la observación en la entrega del equipo al su propietario (*Figura 90*).

Reparación de Equipo

		No. Ticket:	TR-0002
Cliente:	Ana Cervantes		
Equipo:	PORTATIL	Serie:	123456789
Marca:	DELL	Modelo:	INSPIRON
Observación Equipo:	PORTATIL COLOR AZUL CON LOGO MARCA DELL		
Técnico:	Ricardo Paul Manzano Peñafiel	Fecha Ingreso:	2020-12-16
Motivo de Ingreso:	Se reinicia a cada momento		
Costo Reparación:	20.00 USD		

**Historial de Reparación**

Estado	Fecha	Técnico	Observaciones
Ingreso	2020-12-17	Fernanda Manzano	Se reinicia a cada momento
Diagnóstico	2020-12-20	Ricardo Paul Manzano Peñafiel	El equipo necesita formateo
Reparación	2020-12-20	Ricardo Paul Manzano Peñafiel	El equipo se ha foramteado se han instalado programas utilitarios


**Entrega del Equipo**

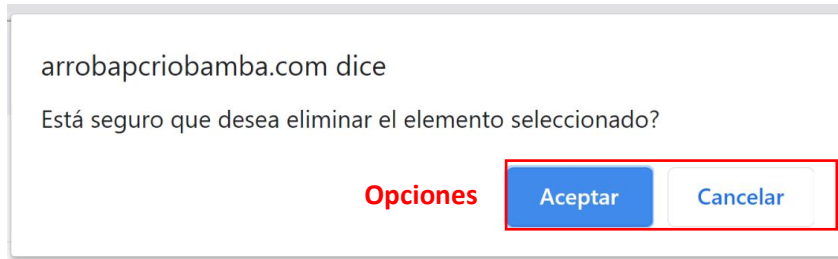
Se entrega el equipo fucionando perfectamente

Guardar

Figura 90: Opción Entrega de Equipo - Rol Recepcionista

Fuente: Autor

**Eliminar Ticket** esta opción  permite eliminar un ticket de reparación siempre y cuando el ticket aún no haya sido gestionado. Al presionar sobre el botón eliminar se mostrará un mensaje de confirmación. (*Figura 91*).



**Figura 91:** Mensaje de Confirmación de Eliminar Ticket - Rol Recepcionista

**Fuente:** Autor

## 5.4. Reportes

En esta opción el usuario con rol Recepcionista puede visualizar el listado de los tickets de reparación que los clientes han solicitado, el usuario Recepcionista puede hacer una búsqueda por cliente, técnico y estado del ticket. (*Figura 92*)

Reporte Tickets

[Imprimir](#)

Cliente  Técnico  Estado  [Buscar](#)

Ticket	Cliente	Equipo	Técnico	Fecha Ingreso	Fecha Entrega	Costo Reparacion	Estado
TR-0001	William Adriano	PORTATIL	Ricardo Paul Manzano Peñafiel	2020-12-12		0.00	Entregado
TR-0002	Ana Cervantes	PORTATIL	Ricardo Paul Manzano Peñafiel	2020-12-16		20.00	Reparación
TR-0003	William Adriano	PORTATIL	Ricardo Paul Manzano Peñafiel	2020-12-20		0.00	Ingreso

Anterior **1** Siguiente

**Figura 92:** Reporte de Tickets de Reparación - Rol Recepcionista

**Fuente:** Autor

Posterior a la aplicación de los criterios de búsqueda el usuario puede imprimir el reporte, al presionar el botón que se encuentra en la parte superior derecha el cual visualizará una pantalla con la información deseada para poder imprimirla (*Figura 93*)



Reporte Reparación de Equipos

Fecha de reporte 2020-12-21

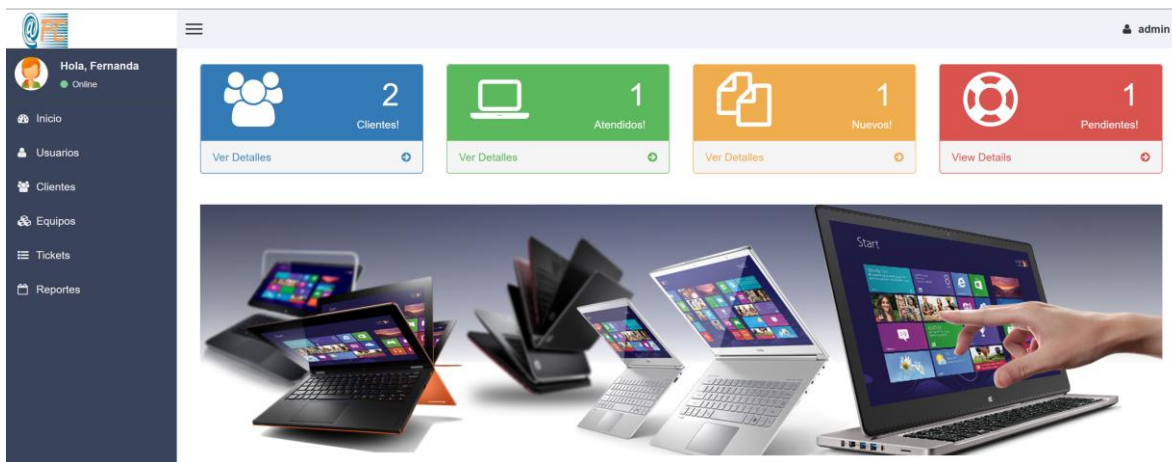
Ticket	Cliente	Equipo	Técnico	Fecha Ingreso	Fecha Entrega	Costo Reparacion	Estado
TR-0001	William Adriano	PORTATIL	Ricardo Paul Manzano Peñafiel	2020-12-12		0.00	Entregado
TR-0002	Ana Cervantes	PORTATIL	Ricardo Paul Manzano Peñafiel	2020-12-16		20.00	Reparación
TR-0003	William Adriano	PORTATIL	Ricardo Paul Manzano Peñafiel	2020-12-20		0.00	Ingreso

**Figura 93:** Reporte de Tickets de Reparación Vista de impresión - Rol Recepcionista

**Fuente:** Autor

## 6. ROL ADMINISTRADOR

Cuando el usuario con rol Administrador ingresa al sistema se muestra el siguiente menú principal en la pantalla de bienvenida (*Figura 94*), donde el usuario Administrador puede en cada opción hacer lo siguiente:



**Figura 94:** Pantalla de Bienvenida – Rol Administrador

Fuente: Autor

### 6.1. Usuarios

En esta opción el usuario con rol Administrador puede visualizar el listado de usuarios del sistema registrados y realizar una búsqueda ingresando cualquier texto de las columnas. (*Figura 95*)

Usuarios

[Añadir](#)

**Búsqueda**

Identificación	Nombres	Apellidos	Usuario	Email	Tipo Usuario	Avatar	Acciones
0601854003	Ricardo Paul	Manzano Peñafiel	tecnico	luna251215@hotmail.com	Técnico		
0604039909	Paulina Isabel	Manzano Calderon	repcionista	arq.jacomejose@gmail.com	Recepcionista		
9999999999	Fernanda	Manzano	admin	mf.manzano17@gmail.com	Administrador		

Anterior **1** Siguiente

**Figura 95:** Listado de Usuarios - Rol Administrador

Fuente: Autor

Para crear un nuevo usuario se debe presionar sobre el botón “Añadir”, ubicado en la parte superior izquierda (Figura 96)

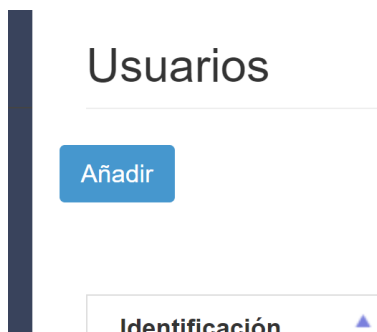


Figura 96: Opción Nuevo Usuario - Rol Administrador

Fuente: Autor

El Sistema presenta una pantalla con un formulario que tiene los siguientes campos de información (Figura 97):

A screenshot of a form titled "Usuario" with a close button (x) in the top right corner. The form contains the following fields:

- Tipo Usuario: A dropdown menu with "Seleccione" as the selected option.
- Número de Identificación: A text input field.
- Nombres: A text input field.
- Apellidos: A text input field.
- Dirección: A text input field.
- Email: A text input field.
- Teléfono: A text input field.
- Celular: A text input field.
- Genero: A dropdown menu with "Seleccione" as the selected option.
- Nombre de Usuario: A text input field.
- Contraseña: A text input field with four asterisks (\*\*\*\*).
- Repetir Contraseña: A text input field with four asterisks (\*\*\*\*).

At the bottom left of the form, there is a green button labeled "Guardar".

Figura 97: Nuevo Usuario - Rol Administrador

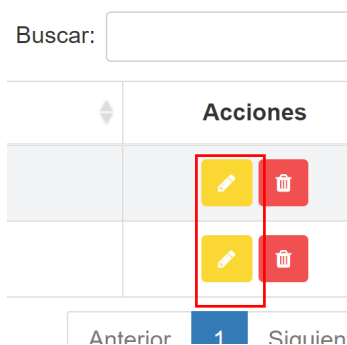
Fuente: Autor

Los campos presentados anteriormente deben ser ingresados de forma obligatoria, caso contrario se presentarán los mensajes de alerta. Cada uno de los campos debe ser ingresado como se detalla a continuación.

- Tipo de Usuario: se debe seleccionar el tipo de Usuario.
- Número de Identificación: Se debe ingresar la Cédula del Usuario.
- Nombres: Se debe ingresar el Nombre del Usuario.
- Apellidos: Se debe ingresar el Apellido del Usuario.
- Dirección: Se debe ingresar la Dirección del Usuario.
- Email: Se debe ingresar el Email del Usuario.
- Teléfono: Se ingresa el Teléfono del Usuario.
- Celular: Se debe ingresar el Celular del Usuario
- Género: Se debe ingresar el Género del Usuario.
- Nombre de Usuario: Se debe ingresar el Nombre de Usuario con el que va a iniciar sesión.
- Contraseña: Se debe ingresar la contraseña con la que se va a iniciar sesión, se debe además confirmar la contraseña.

Una vez que haya sido ingresada toda la información se debe presionar sobre el botón “**Guardar**” si desea almacenar los datos del Usuario caso contrario presionar el botón “**X**” para salir sin guardar nada ubicado en la parte superior derecha de la pantalla.

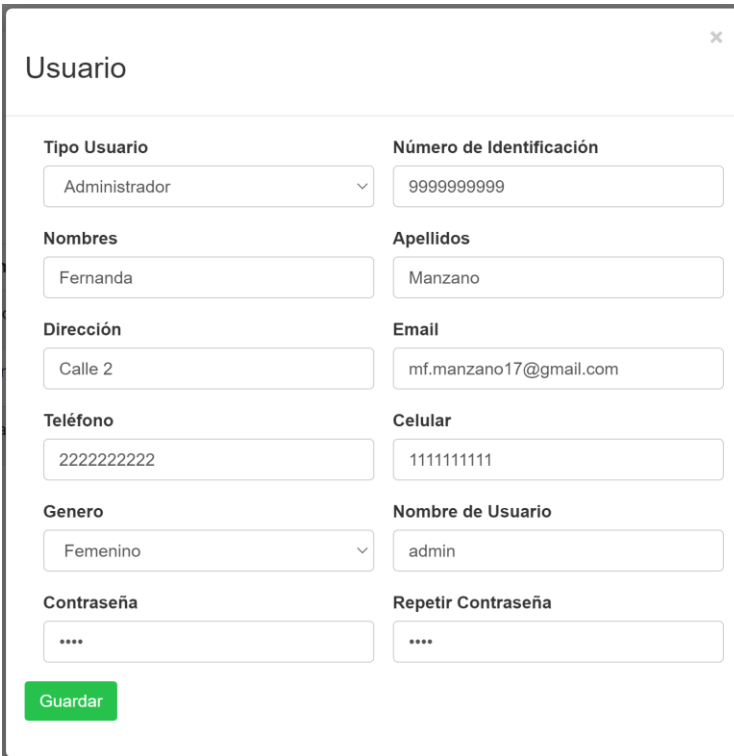
**Para editar los datos de un Usuario** se debe presionar sobre el botón editar, ubicado en cada una de los clientes del listado en la columna de Acciones (*Figura 98*).



**Figura 98:** Opción Editar Usuario - Rol Administrador

Fuente: Autor

Al presionar sobre la opción “*Editar*”, el sistema presentará una página con un formulario con todos los datos que podrían ser editados (*Figura 99*):



El formulario, titulado "Usuario", contiene los siguientes campos:

<b>Tipo Usuario</b> Administrador	<b>Número de Identificación</b> 9999999999
<b>Nombres</b> Fernanda	<b>Apellidos</b> Manzano
<b>Dirección</b> Calle 2	<b>Email</b> mf.manzano17@gmail.com
<b>Teléfono</b> 2222222222	<b>Celular</b> 1111111111
<b>Genero</b> Femenino	<b>Nombre de Usuario</b> admin
<b>Contraseña</b> ****	<b>Repetir Contraseña</b> ****

Un botón verde "Guardar" se encuentra en la parte inferior izquierda del formulario.

**Figura 99:** Editar Usuario - Rol Administrador

**Fuente:** Autor

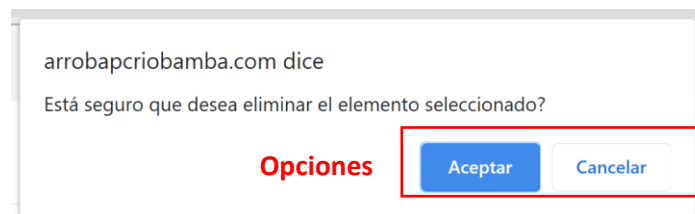
Una vez que los datos hayan sido modificados se debe presionar sobre el botón “*Guardar*” y automáticamente si están correctamente ingresados se guardarán los datos, caso contrario se emitirán mensajes de error los cuales deberían ser corregidos para continuar con la modificación de los datos. En el caso que no se requiera realizar ningún cambio solo se debe presionar sobre el botón “*X*” y se regresará a la pantalla de listado de Usuarios.

**Para eliminar un Usuario** se debe presionar sobre el botón “Eliminar”, ubicado en cada una de los usuarios que se encuentran en el listado en la columna Acciones (*Figura 100*).



**Figura 100:** Opción Eliminar Usuario - Rol Administrador  
**Fuente:** Autor

Una vez que se haya presionado sobre el botón el eliminar saldrá un mensaje de confirmación si desea realizar la eliminación del Usuario. Al presionar sobre el botón “*Aceptar*” automáticamente se borrará el usuario seleccionado (*Figura 101*).



**Figura 101:** Mensaje de Confirmación de Eliminación de Usuarios - Rol Administrador  
**Fuente:** Autor

El usuario Administrador tiene acceso a todas las acciones anteriormente descritas del usuario Recepcionista por lo cual se completa todo el menú de opciones de este usuario.