



# **INSTITUTO TECNOLÓGICO SUPERIOR**

## **“SAN GABRIEL”**

### **ESPECIALIDAD INFORMÁTICA**

#### **TRABAJO DE INVESTIGACIÓN**

#### **PREVIO A LA OBTENCIÓN DEL TÍTULO DE:**

**TECNÓLOGO EN INFORMÁTICA MENCIÓN ANÁLISIS EN SISTEMAS**

#### **TÍTULO:**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB, PARA EL PROCESO DE REGISTRO Y CONTROL DEL SERVICIO DE ALIMENTACIÓN EN LA BRIGADA DE CABALLERÍA BLINDADA N°11 “GALÁPAGOS” DESARROLLADO EN PHP ORIENTADO A OBJETOS CON MOTOR DE BASE DE DATOS MYSQL.**

#### **AUTOR:**

**DIEGO ARMANDO MACAS VILEMA**

**RIOBAMBA-ECUADOR**

**2017**

## **CERTIFICACIÓN**

Certifico que el Sr. **DIEGO ARMANDO MACAS VILEMA**, con el N° de Cédula 0604168724 ha elaborado bajo mi Asesoría el Trabajo de Investigación titulado:

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB, PARA EL PROCESO DE REGISTRO Y CONTROL DEL SERVICIO DE ALIMENTACIÓN EN LA BRIGADA DE CABALLERÍA BLINDADA N°11 “GALÁPAGOS” DESARROLLADO EN PHP ORIENTADO A OBJETOS CON MOTOR DE BASE DE DATOS MYSQL.**

Por tanto autorizo la presentación para la calificación respectiva.

---

**Ing. Andrea Moncayo**

**TUTORA DE TESIS**

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



“Yo, **DIEGO ARMANDO MACAS VILEMA** con N° de Cédula 0604168724, 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.”

---

**DIEGO ARMANDO MACAS VILEMA**

**INSTITUTO TECNOLÓGICO SUPERIOR**

**“SAN GABRIEL”**

**ESPECIALIDAD INFORMÁTICA**

**TESIS DE GRADO**

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

**TECNÓLOGO EN INFORMÁTICA MENCIÓN ANÁLISIS DE SISTEMAS**

**TÍTULO:**

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB, PARA EL PROCESO DE REGISTRO Y CONTROL DEL SERVICIO DE ALIMENTACIÓN EN LA BRIGADA DE CABALLERÍA BLINDADA N°11 “GALÁPAGOS” DESARROLLADO EN PHP ORIENTADO A OBJETOS CON MOTOR DE BASE DE DATOS MYSQL.

**APROBADO:**

**ASESOR DE TESIS DE GRADO** .....

**PRESIDENTE DEL TRIBUNAL** .....

**MIEMBRO DEL TRIBUNAL** .....

**MIEMBRO DEL TRIBUNAL** .....

## FIRMAS DE RESPONSABILIDAD

NOMBRE	FIRMA	FECHA
Ing. Andrea Moncayo Directora de Tesis		
Presidente del Tribunal		
Miembro del Tribunal		
Miembro del Tribunal		

**Nota de la Tesis:** \_\_\_\_\_

## **DEDICATORIA**

### **A Dios**

El presente proyecto dedico de manera especial a Dios por haberme dado la vida y brindarme esta oportunidad de culminar exitosamente cada uno de los objetivos que me he propuesto en mi vida, quien me dio la fortaleza de perseverar y lograr esta meta en especial.

### **A mis padres**

Por ser mis guías fundamentales en todo el transcurso de mi vida, quienes estuvieron pendientes siempre de mi formación académica como personal lo cual mediante sus enseñanzas me inculcaron que todo en la vida se consigue por medio de un sacrificio y eso me permito plasmar en realidad y quienes han permanecido a mi lado en los buenos y malos momentos.

### **A mis hijos**

Fuente de inspiración, necesaria para cumplir todas las metas que se ponga en la vida y pilar fundamental en el logro de esta meta quienes sacrificaron momentos únicos de su niñez por ver alcanzar una meta a su padre.

### **A mi Instituto**

Mi segundo hogar y a todos sus docentes, que desde el inicio de esta etapa estudiantil me formaron para lograr realizarme en todos los aspectos de mi vida.

## **AGRADECIMIENTO**

Mi infinito agradecimiento a DIOS por brindarme el don de la vida, y ayudarme a lograr cada uno de mis propósitos y sueños.

Agradezco a mis Padres y familia en general pilar fundamental en mi formación personal, profesional y hoy académica gracias por su gran apoyo, sus consejos y más que todo por el ejemplo de vida que me brindaron para convertirme en un servidor de mi pueblo.

Manifiesto mi más sincero agradecimiento a la Ing. Andrea Moncayo por el gran aporte guía y paciencia que me brindo, así como a todas las personas que participaron de alguna manera en el desarrollo del proyecto de Investigación, a los sabios consejos y enseñanzas que me ayudaron en el desarrollo del presente proyecto. También expreso mi sincero agradecimiento a la Ing. Alexandra Elizabeth Camba Totoy quien fue la coautora de este logro personal quien aun sin estar presente en mi vida, me enseñó que las promesas se cumplen y que hay personas que se aman y no están juntas, que su amor permanece allí en el silencio y las miradas de fidelidad muda. Que en algún momento fueron, pero no fue el mejor instante, aun así se amaron el resto de sus vidas.

Agradezco a todos los profesores del Instituto, quienes con sus conocimientos, supieron formarme profesionalmente y a toda la familia San Gabriel que de una u otra manera desinteresada me brindaron su ayuda para poder llegar a culminar mi tesis de grado

## ABREVIATURAS

**PHP:** Hypertext Preprocessor

**SQL:** Structured Query Language

**WWW:** World Wide Web

**HTML:** HyperText Markup Language

**XHTML:** Extensible Hypertext Markup Language

**HTTP:** Hypertext Transfer Protocol

**HTTPS:** Hypertext Transfer Protocol Secure

**MVC:** Modelo-Vista-Controlador

**ONE PAGE:** Refleja toda la información mediante la elaboración de una sola página.

**BLOGS:** Sitios personales donde la principal función es crear información

**CC.FF.AA:** Comando conjunto de Fuerzas Armadas.

**IIS:** Servicios de información de internet

**ACL:** Lista control de acceso

**PDO:** extensión Objetos de Datos

**ORM:** Asignación relacional de objetos

**UML:** Lenguaje unificado de modelado

**SVN:** herramienta de control de versiones open source

**CVS:** Sistema concurrente de control de versiones

**BOBSTRA:** Desarrolle proyectos receptivos

**CSC:** Hojas de Estilo en Cascada

**FRAMEWOR:** Entorno de trabajo

**FT:** Fuerza terrestre

**DDIE:** Dirección de desarrollo institucional del Ejercito.

**PK:** Primario (Identificar elementos de cada tabla)

**NN:** No nulo (Campo obligatorio)

**AI:** auto incremental (de almacenamiento automático)

## GLOSARIO DE TÉRMINOS

**GRADO:** Jerarquía que se mantiene dentro del ejército.

**UNIDAD:** Reparto militar determinada.

**COMANDAS:** Registro de control del servicio de alimentación.

**CONFRONTAS:** Documento de respaldo del detalle de consumo del servicio de alimentación.

**AMANUENSE:** Militar que cumple funciones y responsabilidades en un cargo asignado y acorde al departamento asignado.

**OFICIAL:** Corresponde a la carrera, categoría o nivel del miembro de las fuerzas armadas que de forma en la escuela superior militar del ejército.

**VOLUNTARIO:** Corresponde al grupo de miembros de las fuerzas armadas formados en la escuela de formación de soldados del ejército.

**CONSCRIPTO:** Ciudadano ecuatoriano que realiza el servicio militar voluntario.

**CONFRONTA:** Asignación del servicio de alimentación para un miembro de las fuerzas armadas.

**EXTRA CONFRONTA:** Generación de un servicio de alimentación adicional al usuario que no fue considerado para el mismo.

**CONSOLIDADO:** Reporte generado por cada una de las unidades militares del detalle de confrontas de todo su personal.

**ARMA/ SERVICIOS /ESPECIALISTA:** Especialidad asignada al miembro e las fuerzas armadas para su formación y desempeño profesional.

**GRADO:** Corresponde a la jerarquía de cada uno de los miembros de las fuerzas armadas.

**RANCHERO:** Oficial designado para la administración, control y supervisión del proceso de confección del servicio de alimentación.

**GESTIÓN DE CONFRONTAS:** Proceso de registro de las comandas para el servicio de alimentación de un miembro de las fuerzas armadas.

# ÍNDICE GENERAL

CERTIFICACIÓN.....	II
FIRMAS DE RESPONSABILIDAD .....	V
DEDICATORIA.....	VI
AGRADECIMIENTO .....	VII
ABREVIATURAS .....	VIII
GLOSARIO DE TÉRMINOS .....	X
ÍNDICE GENERAL.....	12
ÍNDICE DE TABLAS.....	16
ÍNDICE DE GRÁFICOS.....	17
INTRODUCCIÓN.....	<b>¡Error! Marcador no definido.</b>
RESUMEN.....	<b>¡Error! Marcador no definido.</b>
SUMMARY.....	23
CAPÍTULO I.....	24
1    MARCO REFERENCIAL.....	24
1.1    FORMULACIÓN GENERAL DEL PROYECTO DE TESIS.....	25
1.1.1    ANTECEDENTES DEL PROBLEMA DE INVESTIGACIÓN .....	25
1.1.2    DATOS INFORMATIVOS .....	26
1.2    DEFINICIÓN DEL PROBLEMA .....	27
1.3    JUSTIFICACIÓN .....	27
1.4    OBJETIVOS .....	28
1.4.1    OBJETIVO GENERAL.....	28
1.4.2    OBJETIVOS ESPECÍFICOS .....	28
CAPÍTULO II.....	29
2    MARCO TEÓRICO .....	29

2.1	CONCEPTUALIZACIONES GENERALES.....	30
2.1.1	Sistema web.....	30
2.1.2	Sitio web.....	31
2.1.3	Aplicaciones web.....	35
2.1.4	Servidor web.....	43
2.1.5	Apache.....	48
2.1.6	Características.....	48
2.2	PROGRAMACIÓN ORIENTADA A OBJETOS.....	51
2.2.1	Evolución de los Lenguajes de Programación.....	51
2.2.2	Conceptos de la Programación Orientada a Objetos.....	52
2.2.3	Características del Lenguaje Orientados a Objetos.....	53
2.3	PHP.....	55
2.3.1	Importancia.....	55
2.3.2	Características.....	56
2.3.3	Estructuras del Código Fuente en Php.....	59
2.3.4	Elementos Básicos del Lenguaje PHP.....	60
2.3.5	Sentencias de control.....	64
2.4	MYSQL.....	73
2.4.1	¿Por qué usar MySQL?.....	73
2.4.2	Características.....	74
2.4.3	Ventajas y Desventajas.....	75
2.4.4	Estructuras de Tabla y de Índice.....	76
2.4.5	Estructura física de los registros.....	79
2.4.6	Procedimientos almacenados y funciones.....	80
2.5	MVC (MODELO VISTA CONTROLADOR).....	82

2.5.1	CAPAS DEL MVC.....	83
2.5.2	Características.....	84
2.6	HERRAMIENTAS DE DESARROLLO .....	85
2.6.1	Herramientas para Modelado de Software.....	85
2.6.2	Herramientas de Desarrollo de Software .....	86
2.7	NETBEANS.....	88
2.7.1	Plataforma Netbeans .....	89
2.7.2	Características.....	89
2.7.3	Ventajas y Desventajas de Netbeans .....	92
2.7.4	Características de NetBeans integrado a PHP .....	92
2.8	WAMPSEVER.....	95
2.8.1	Características.....	95
2.8.2	Herramientas para el diseño y desarrollo de páginas web .....	95
2.8.3	Funcionalidades de WampServer .....	96
2.8.4	Utilidad .....	96
2.9	BOOTSTRAP .....	97
2.9.1	Ventajas de usar Bootstrap .....	98
2.9.2	Desventajas de usar Bootstrap .....	99
	CAPÍTULO III .....	100
3	ANÁLISIS Y DISEÑO DEL SISTEMA.....	100
3.1	RECOPIACIÓN DE INFORMACIÓN.....	101
3.1.1	Alcance .....	101
3.2	ANÁLISIS .....	102
3.2.1	Estudio de Factibilidad .....	102
3.2.2	Análisis de los requerimientos.....	105

3.2.3	Casos de Uso.....	112
3.1	DISEÑO.....	117
3.1.1	Diseño Conceptual.....	117
3.1.2	Modelo Relacional.....	119
3.1.3	Diccionario de Datos .....	120
3.1.4	Diseño de Interfaces.....	129
CAPÍTULO IV .....		132
4	IMPLEMENTACIÓN DEL SISTEMA WEB.....	132
4.1	HERRAMIENTAS DE DESARROLLO .....	133
4.1.1	Instalación del WampServer.....	133
4.1.2	Instalación del entorno de desarrollo NetBean .....	139
4.2	ARQUITECTURA DEL SISTEMA WEB.....	142
4.3	IMPLEMENTACIÓN DEL SISTEMA.....	143
4.3.1	Definición de los subsistemas de implementación .....	143
4.3.2	Desarrollo de la aplicación implementación.....	144
4.3.3	Implementación del sistema web.....	144
4.4	PRUEBAS DEL SISTEMA WEB.....	145
4.5	CAPACITACIÓN AL PERSONAL.....	148
4.6	MANTENIMIENTO DEL SISTEMA WEB.....	148
CAPÍTULO V.....		149
5	CONCLUSIONES Y RECOMENDACIONES .....	149
5.1	CONCLUSIONES .....	150
5.2	RECOMENDACIONES.....	150
BIBLIOGRAFÍA .....		152
ANEXOS .....		156

## ÍNDICE DE TABLAS

Tabla N° 1. Recurso humano.....	103
Tabla N° 2. Recurso técnico hardware .....	103
Tabla N° 3. Recurso técnico software .....	104
Tabla N° 4. RF 01 Presentar el Portal Web Informativo.....	105
Tabla N° 5. RF 02 Presentar Inicio de Sesión .....	106
Tabla N° 6. RF 03 Administra gestión de Usuarios .....	106
Tabla N° 7. RF 04 Generar reportes .....	106
Tabla N° 8. RF 05 Configurar parámetros .....	107
Tabla N° 9. RF 06 Generar Personas Usuarios .....	107
Tabla N° 10. RF 07 Generar Novedades .....	107
Tabla N° 11. RF 08 Generar Confrontas .....	108
Tabla N° 12. RF 09 Configurar Parámetros Unidad .....	108
Tabla N° 13. RF 10 Generar Consumos .....	108
Tabla N° 14. RF 11 Visualizar Confrontas Anteriores.....	109
Tabla N° 15. RF 12 Visualizar Confrontas Unidades .....	109
Tabla N° 16. RF 13 Visualizar y crear Extra-confrontas .....	109
Tabla N° 17. RF 14 Consolida información .....	110
Tabla N° 18. RF 15 Consolida información .....	110
Tabla N° 19. RNF 01 Interfaz sencilla y navegable .....	111
Tabla N° 20. RNF 02 Escalabilidad .....	111
Tabla N° 21. RNF 03 Seguridad de Acceso .....	111
Tabla N° 22. Diccionario de datos.....	128

## ÍNDICE DE GRÁFICOS

Gráfico N° 1. Proceso de construcción de páginas dinámicas .....	46
Gráfico N° 2. Tendencia de uso de servidores web.....	49
Gráfico N° 3. Esquema 1 http.....	50
Gráfico N° 4. Esquema 2 interpretado con PHP .....	50
Gráfico N° 5. Características de la POO .....	53
Gráfico N° 6. Variables .....	61
Gráfico N° 7. Constantes .....	62
Gráfico N° 8. Precedencia de los operadores .....	64
Gráfico N° 9. Ejemplo sentencia if.....	65
Gráfico N° 10. Ejemplo sentencia else .....	66
Gráfico N° 11. Ejemplo de sentencia elseif.....	67
Gráfico N° 12. Sintaxis de estructuras de control .....	67
Gráfico N° 13. Ejemplo sentencia if evaluada como TRUE .....	68
Gráfico N° 14. Ejemplo sentencia if evaluada como FALSE .....	69
Gráfico N° 15. Ejemplo sentencia do-while.....	69
Gráfico N° 16. Sentencia FOR .....	70
Gráfico N° 17. Ejemplo sentencia for each .....	71
Gráfico N° 18. Ejemplo sentencia switch.....	72
Gráfico N° 19. Estructura de tabla e índice MySQL.....	76
Gráfico N° 20. Capas del Modelo Vista Controlador.....	83
Gráfico N° 21. Funcionamiento del MVC.....	84
Gráfico N° 22. Ejemplo de diagramas UML.....	86
Gráfico N° 23. Caso de uso-ingreso al sistema .....	113
Gráfico N° 24. Caso de uso-gestión de usuarios .....	114

Gráfico N° 25. Caso de uso-gestión de reportes.....	114
Gráfico N° 26. Caso de uso-gestión de catálogos .....	115
Gráfico N° 27. Caso de uso-configuración de parámetros .....	115
Gráfico N° 28. Caso de uso-proceso de consumos.....	116
Gráfico N° 29. Caso de uso-configuración de parámetros .....	117
Gráfico N° 30. Diseño conceptual.....	118
Gráfico N° 31. Modelo relacional .....	119
Gráfico N° 32. Pantalla principal del sistema web.....	129
Gráfico N° 33. Inicio de sesión del sistema web.....	129
Gráfico N° 34. Usuario administrador en el sistema web .....	130
Gráfico N° 35. Usuario amanuense en el sistema web.....	130
Gráfico N° 36. Usuario ranchero en el sistema web.....	131
Gráfico N° 37. Descarga WampServer.....	133
Gráfico N° 38. Ejecución de WampServer.....	134
Gráfico N° 39. Aceptación de términos y condiciones de Wamp Server.....	134
Gráfico N° 40. Selección de destino de instalación del WampSever .....	135
Gráfico N° 41. Creación de acceso directo del WampServer.....	135
Gráfico N° 42. Instalación de WampServer .....	136
Gráfico N° 43. Proceso de instalación de WampServer .....	136
Gráfico N° 44. Determinación del navegador a utilizar para WampServer .....	137
Gráfico N° 45. Parámetros para uso de PHP en WampServer .....	137
Gráfico N° 46. Instalación completa de WampServer .....	138
Gráfico N° 47. Ejecución de WampServer en segundo plano.....	138
Gráfico N° 48. Descarga de NetBeans .....	139
Gráfico N° 49. Opciones de descarga de NetBeans .....	140

Gráfico N° 50. Descarga del depurador.....	140
Gráfico N° 51. Descarga del compilador para NetBeans .....	141
Gráfico N° 52. Configuración Patch.....	142
Gráfico N° 53. Arquitectura del sistema web.....	142
Gráfico N° 54. Subsistemas del sistema web .....	143
Gráfico N° 55. Página principal del sitio web.....	144
Gráfico N° 56. Prueba de funcionalidad.....	145
Gráfico N° 57. Prueba de estabilidad .....	146
Gráfico N° 58. Prueba de escalabilidad.....	147
Gráfico N° 59. Prueba de seguridad de acceso.....	147

## INTRODUCCIÓN

El presente trabajo de investigación se desarrolla bajo el marco de la realidad problemática de la Brigada de Caballería Blindada N° 11 “Galápagos”, que no cuenta con un sistema de control de alimentación para un número riguroso de integrantes a quienes se brinda el servicio de comida diaria, haciendo que los alimentos o comandas genere una gran cantidad de información la misma se maneja actualmente por medio de tablas de Excel, lo que ha generado que la información, se entregue con lentitud, de forma desorganizada y la aparición de inconsistencias de datos así como inconformidad por parte del personal, los cuales, imposibilitan la optimización del tiempo al momento de realizar la liquidación mensual de este servicio.

El sistema Web para el proceso de Registro y Control del Servicio de Alimentación en la Brigada de Caballería Blindada N° 11 “Galápagos” permitió que la información se la maneje por medio de métodos sistemáticos y tecnológicos, haciendo que los datos puedan manipularse fácilmente, permitiendo a cualquier entidad tener una organización adecuada de los procesos con información confiable, actualizada, real que permitirá mejorar los rubros económicos, financieros de del personal de la brigada de caballería blindada Galápagos.

Como futuro Tecnólogo realicé el análisis, diseño y desarrollo de este sistema para suplir la necesidad informativa de los procesos de alimentación de la brigada Galápagos motivada por los conocimientos adquiridos en el área de desarrollo web durante mi proceso académico estudiantil.

Por lo cual, el objetivo principal de la investigación es diseñar e implementar un sistema web, para el proceso de registro y control del servicio de alimentación en la Brigada de

Caballería Blindada N°11 “GALÁPAGOS” desarrollado en PHP Orientado a Objetos con motor de base de datos MySQL.

Finalmente se da a conocer que, este trabajo de investigación consta de 5 capítulos los cuales se describen a continuación:

El Capítulo I. describe el marco referencial que se encuentra compuestos por la formulación, los antecedentes y la definición del problema, También, consta de la justificación de la investigación y los objetivos propuestos para la elaboración del sistema web.

En el Capítulo II. Describe el Marco teórico por medio de conceptualizaciones generales hasta las más específicas que sirvieron como guía para el desarrollo del sitio web.

En el Capítulo III. Describe todas las actividades necesarias para la realización del análisis y diseño del sistema web, donde se detallan factores que van desde el alcance hasta el diseño de las interfaces.

En el Capítulo IV. En este apartado se describieron todos los pasos necesarios para implementación del sistema web por medio del uso de herramientas de desarrollo, la arquitectura del sistema, las pruebas y finalmente la capacitación al personal.

Finalmente en el Capítulo V. Correspondiente a la elaboración de las conclusiones y recomendaciones que fueron establecidas de acuerdo a los objetivos planteados.

## RESUMEN

El trabajo de investigación titulado Diseño e Implementación de un Sistema Web, para el Proceso de Registro y Control del Servicio de Alimentación se desarrolló para la Brigada de Caballería Blindada N°11 “Galápagos”, con el propósito de automatizar los procesos de alimentación que brinda la institución para los señores militares. Para el desarrollo de la aplicación se utilizó el lenguaje de programación PHP versión 5.6.31 para crear formularios, colores, textos, contenidos permitiendo mostrar las interfaces graficas que los usuarios ejecutan al momento de subir información en el sistema. Al mismo tiempo el Gestor de Base de Datos MYSQL proporciona una Base de Datos de almacenamiento físico de: usuarios, imágenes, enlaces entre tablas, contenidos y contraseñas administrados por la persona encargada de manejar el sistema de alimentación, de modo que pueda realizar fácilmente operaciones como la creación, actualización, eliminación. Una vez culminado el desarrollo del aplicativo se realizaron pruebas de validación mediante entrevista dirigidas al personal lo que corroboró que el sitio fue creado cumpliendo los requerimientos, pruebas y tiempos de procesos que confirmaron la optimización de los mismos.

La implementación de este sitio permitirá el ingreso de datos con respecto a las comandas, a ser manejado por tres usuarios directos como lo son el administrador, amanuense y rancharo. Quienes serán los encargados del manejo del sistema y del ingreso de información, para así generar un reporte organizado y de una manera sencilla, acción que permitirá a la Unidad de Caballería N°11 “Galápagos” tener un control permanente sobre el servicio de alimentación evitando así el desperdicio de alimentos y posibles pérdidas económicas.

## **SUMMARY**

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

## **1.1 FORMULACIÓN GENERAL DEL PROYECTO DE TESIS**

### **1.1.1 ANTECEDENTES DEL PROBLEMA DE INVESTIGACIÓN**

En la actualidad el registro y control del servicio de alimentación dentro de la Brigada de Caballería Blindada N° 11 “Galápagos”, se lo realiza a través del Departamento de Recursos Humanos (B-1), en conjunto con las oficinas de Recursos Humanos (P-1) de cada una de las unidades que tiene bajo su responsabilidad brindar asistencia alimentaria a los señores Oficiales, Voluntarios y Conscriptos de su dependencia.

El proceso antes mencionado se lo realiza de forma manual, en base a la Directiva N° 04 del 2010 para la administración de la asignación del rancho del personal militar de las Fuerzas Armadas, Instructivo N° FT-DDIE-2015-004-O para la administración del servicio de alimentación en las unidades militares de la fuerza terrestre, publicada en Orden General N° 03 de fecha 06-ENE-2016. El AMANUENSE DE PERSONAL (persona encargada de la recopilación de información referente a la confronta diaria) de las unidades emite para el control del servicio documentos elaborados en tablas de Excel, enfrentando problemas de retraso, complejidad en el uso del programa, desorganización e inconsistencias de datos así como inconformidad por parte del personal. Además, imposibilita la optimización del tiempo al momento de realizar la liquidación mensual de este servicio.

El manejo de la cantidad alimenticia requerida es emitida a través de tickets por el Sr. Oficial rancharo según la cantidad de las confrontas diarias requeridas, esta actividad es insegura, sea por la duplicidad o pérdida de dichos documentos.

Por todos los agravantes mencionados, el establecimiento se ha visto en la necesidad de utilizar un sistema tecnológico que le permita registrar, organizar y procesar la

información de todos los requerimientos necesarios para la entrega de un buen servicios de alimentación dentro de la Brigada de Caballería Blindada N° 11 “Galápagos”.

### **1.1.2 DATOS INFORMATIVOS<sup>1</sup>**

#### **Misión**

Defender la Soberanía y la Integridad Territorial, apoyar con su contingente al desarrollo nacional, contribuir con la seguridad pública y del Estado y participar en operaciones de paz y ayuda humanitaria (Comando Conjunto de las Fuerzas Armadas del Ecuador, 2016).

#### **Visión**

Institución de más alto nivel de credibilidad; sistémicamente integrada, con capacidades conjuntas e interoperabilidad, personal profesional, ético y moralmente calificado, para enfrentar los cambios y nuevos escenarios, que garanticen la paz, seguridad y el bienestar de la nación (Comando Conjunto de las Fuerzas Armadas del Ecuador, 2016).

#### **Valores institucionales**

- Subordinación al Interés Nacional
- Entrega Irrestriccta a la Defensa de la Patria
- Lealtad
- Espíritu Militar
- Disciplina
- Respeto a la Jerarquía
- Cohesión

---

<sup>1</sup> <https://www.cffaa.mil.ec/mision-vision-valores-institucionales/>

- Cultura Democrática
- Moral Militar
- Honor
- Honestidad
- Ética militar

## **1.2 DEFINICIÓN DEL PROBLEMA**

Actualmente la Brigada de Caballería Blindada N° 11 “Galápagos”, utiliza un sistema obsoleto para el registro y control del servicio de alimentación, causando molestias con la entrega de un servicio deficiente en varias ocasiones a los señores Oficiales, Voluntarios y Conscriptos. Además de provocar pérdidas económicas con el desperdicio de los recursos alimenticios.

## **1.3 JUSTIFICACIÓN**

La investigación técnicamente se ve sustentada gracias al estudio teórico del lenguaje de programación PHP orientado a objetos, esta actividad permitió despejar varias dudas encontradas en el proceso de aprendizaje a través de la aplicación de los conocimientos adquiridos los cuales fueron puestos en práctica en el desarrollo de un sistema web que promovió el adecuado manejo, control y registro del servicio alimenticio que oferta la Brigada de Caballería Blindada N° 11 a sus Oficiales, Voluntarios y Conscriptos.

Metodológicamente el proyecto resulta viable porque servirá como documento guía para futuras investigaciones que requieran la implementación de un sistema web para el registro y control de servicios mediante el uso de PHP orientado a objetos.

La realización del proyecto contó con todos los requerimientos necesarios para su desarrollo sean estos de tipo técnico, humano, tecnológico y financiero.

El sistema web beneficiará de forma directa a la institución, otorgándole una herramienta tecnológica que le permitirá mejorar el control y registro de todas las comandas percibidas evitando pérdidas de recursos alimenticios y de esta manera contribuir con la estabilidad financiera de la Brigada de Caballería Blindada N° 11.

## **1.4 OBJETIVOS**

### **1.4.1 OBJETIVO GENERAL**

Diseñar e implementar un sistema web, para el proceso de registro y control del servicio de alimentación en la Brigada de Caballería Blindada N°11 “GALÁPAGOS” desarrollado en PHP Orientado a Objetos con motor de base de datos My-SQL.

### **1.4.2 OBJETIVOS ESPECÍFICOS**

- Establecer y determinar las herramientas disponibles y necesarias para el desarrollo e implementación del presente sistema.
- Analizar las características, ventajas, desventajas y todas las herramientas disponibles para la creación del presente sistema.
- Diseñar un sistema web interno para el registro y control del servicio de alimentación dentro de la Brigada de Caballería Blindada N° 11 “GALÁPAGOS”.
- Implementar el sistema web interno para el registro y control de las confrontas del servicio de alimentación dentro de la Brigada de Caballería Blindada N° 11 “GALÁPAGOS”.

**CAPÍTULO II**  
**MARCO TEÓRICO**

## **Introducción**

El marco teórico reunirá todos los temas y conceptualizaciones necesarias para comprender con mayor precisión como se elabora y cuáles son las ventajas o riesgos que se tendrá a lo largo de la investigación que se basa en la realización de un sistema web para el registro y control desarrollado en PHP orientado a objetos con motor de base de datos MySQL.

## **1.5 CONCEPTUALIZACIONES GENERALES**

### **1.5.1 Sistema web<sup>2</sup>**

Los sistemas Web se los conoce también como aplicaciones Web, su desarrollo no se ve reflejado dentro de una plataforma o sistema operativo, sino que puede ser alojado en un servidor de internet o en una red local. Su apariencia es parecida a la de un sitio web, pero con las diferencias de sus funciones, sobre todo en la potente respuesta que poseen en varios casos.

Los sistemas pueden funcionar en cualquier tipo de navegador sea Google Chrome, Firefox, Internet Explorer, etc. Además, no es necesario contar con un sistema operativo específico y el proceso de instalación no es requerido, pues los usuarios se podrán conectar desde algún servidor donde el sistema se encuentre alojado.

Un sistema Web desarrollado en una plataforma Web posee marcadas diferencias con relación a otros, estas cualidades representan un aporte importante para cualquier empresa que haga uso de ellos. Entre las ventajas se pueden mencionar a costos de desarrollo, agilidad en la entrega y obtención de la información pero sobre todo crea una gestión que promueve la optimización de recursos

---

<sup>2</sup> <http://www.knowdo.org/knowledge/39-sistemas-web>

## 1.5.2 Sitio web<sup>3</sup>

Al hablar de un sitio web se lo relaciona con páginas web, las mismas que contienen código HTML, mediante la WWW que pueden ser visualizadas en navegadores.

La funcionalidad inmersa al interior de los sitios web, depende directamente de la utilidad que se dé a las páginas que lo conforman, la característica principal es la de publicar contenidos para el público en general.

### 1.5.2.1 Clasificación de un sitio web

Los sitios web podrían clasificarse de dos formas diferentes, por ejemplo.

- **Sitio Web Estáticos:** como su nombre lo indica es un sitio que no requiere un continuo cambio de la información, es decir, no poseen una base de datos para obtener los contenidos. Generalmente se los utiliza como los propietarios del sitio.
- **Sitio Web Dinámico:** caso contrario con el sitio anterior, es este se encuentra con una base de datos para la obtención de los contenidos y se reflejan resultados de las páginas web. Este sitio podrá ser modificado, perfeccionado o limitado su acceso por medio de un sistema web que contendrá un determinado número de usuarios con su respectiva contraseña.

Al solicitar la creación de un sitio web, el cliente deberá definir la cantidad de páginas que necesita y si requiere según sus necesidades la existencia de características dinámicas o estáticas.

El contar con un sitio web a nivel empresarial puede ser de gran utilidad para el cumplimiento de los objetivos de la institución por ejemplo en las áreas de ventas, marketing, administración, gestión, compras, etc. Un sitio pretende cubrir características

---

<sup>3</sup> <http://www.codedimension.com.ar/noticias-sobre-tecnologia/noticias/-que-es-y-para-que-sirve-un-sitio-web-/1>

trascendentales como la difusión de información institucional a una gran cantidad de usuarios a través de la promoción y que todos los datos obtenidos puedan verse organizados para su utilización cuando la entidad así lo requiera.

#### 1.5.2.2 Tipos de sitios web<sup>4</sup>

En el mercado se pueden encontrar varios tipos de sitios web y se los puede describir de la siguiente manera:

- **Institucionales:** poseen páginas básicas con las generalidades principales de la empresa y se la puede establecer con los siguientes componentes:
  - Página Principal o Home
  - Misión, Visión, Valores, objetivos
  - Servicios que oferta
  - Ubicación
  - Información de contactos
- **One page:** Refleja toda la información mediante la elaboración de una sola página y que el usuario puede acceder a los datos al bajar la vista sea con el teclado o el mouse. Es de bajo costo y es muy efectiva si se encuentra bien desarrollada.
- **Blogs:** representan aquellos sitios personales donde la principal función es crear información con un orden cronológico y ordenado. Además es incluyente con el almacenamiento de varios artículos pudiendo crear un debate o feedback entre los interesados. Pertenecen al grupo de los sitios web dinámicos.
- **Sitios e-commerce:** sitios destinados para la compra y venta de artículos, conocidos como los carritos de compra y que cuentan con las opciones de generar una

---

<sup>4</sup> <http://www.codedimension.com.ar/noticias-sobre-tecnologia/noticias/-que-es-y-para-que-sirve-un-sitio-web-/1>

adquisición mediante una tarjeta de crédito, online o realizar reservas de productos o servicios

### 1.5.2.3 Fases para el desarrollo de un sitio web<sup>5</sup>

Para el desarrollo de los sitios web existen seis etapas clave que se describen a continuación

- **Investigación:** la investigación debe considerar toda la cartera de internet disponible con relación a trabajos similares sobre la realización del sitio web y los recursos de marketing y características disponibles. Todas estas actividades deberán generar un estilo adecuado de diseño que satisfaga las necesidades del consumidor.
- **Planificación:** la planificación es el proceso por el cual se pretende considerar ciertos parámetros para satisfacer la necesidad del usuario entre ellos se detallan:
  - Descubrir los gustos y preferencias
  - Establecer la función y metas a alcanzar con el sitio web
  - Definir los temas que se desean cubrir
  - Elegir un nombre para el sitio
  - Verificar la disponibilidad del nombre del dominio y registrarlo
  - Comparar con otros sitios elaborados
- **Arquitectura del sitio:** con relación a las necesidades encontradas se debe realizar un mapa que organice todo el contenido del sitio de una forma estructural.
- **Diseñar, construir y hacer pruebas:** con el diseño final se verifica si realmente se está transmitiendo lo que se quería mostrar, posterior a ello se incorporan imágenes, fotografías y algunos componentes para ser finalmente puesto a prueba por los usuarios y grupos de interés.

---

<sup>5</sup> <http://disenowebakus.net/fases-para-el-desarrollo-de-un-proyecto-web.php>

- **Operar, mantener y evaluar:** esta etapa promueve la constante manutención del sistema donde se pueden sugerir posibles mejoras o actualizaciones por medio de informes de rendimiento que en su defecto otorgan información importante en la toma de decisiones sobre la reconstrucción del sitio.

#### 1.5.2.4 Elementos del diseño de un sitio web<sup>6</sup>

El diseño del sitio web se ve marcado de acuerdo a su constitución o al conjunto de páginas web que lo conforman. Por tanto, una página debe contener una serie de elementos para cumplir con ciertos requisitos que cubran las necesidades del usuario, entre ellos se encuentran:

- **Contenido:** Las páginas no solamente deben captar la atención de los usuarios de forma visual sino también con la actualización de contenidos que satisfagan las expectativas del lector. Esta actividad aumentará el número de visitas y otorgará información nueva cada vez que el visitante ingrese a la página.
- **Imágenes:** Enmarcado en el grupo visual conformado por fotografías, imágenes e ilustraciones. Cada uno de los elementos descritos deberá contener una alta resolución y evitar la distorsión. Preferentemente se recomienda que se encargue del desarrollo de este material un profesional para brindar al sitio márgenes de calidad adecuados.
- **Tipografía:** Se debe utilizar fuentes sencillas y simples pero con un gran sustento, conservando las especificaciones del diseño y variando el tamaño de la fuente según la importancia de temas relevantes.
- **Botones e iconos:** La utilización de botones vistosos, representativos y de gran tamaño ayudará a que el usuario pueda guiarse en la página con gran facilidad.

---

<sup>6</sup> <http://redgrafica.com/Elementos-del-diseno-web-estilos-y>

- **Animaciones y movimiento:** las animaciones generan dinamismo en la página, pero no se debe exceder del 20% porque causaría una saturación y la distracción del usuario pudiendo optar en salir del sitio.
- **Fondos claros y sencillos:** La importancia de la sencillez se ve reflejada en la claridad de los contenidos. Un fondo blanco generalmente se utiliza para resaltar imágenes o la tendencia con el color negro cuando existen lecturas extensas evitando el cansancio visual del lector por el reflejo de luz del monitor.
- **Enlaces a redes:** La utilización de redes sociales o de sitios de interés dentro de una comunidad es un gran soporte para el aumento de oferta, proveedores o de clientes nuevos.
- **Sobriedad:** las páginas web deben poseer contenidos dinámicos y sencillos que permitan la fácil navegación del usuario brindando respuestas inmediatas y así evitar la saturación cibernauta que podría desinteresar la oferta que se desea transmitir al lector.

### 1.5.3 Aplicaciones web<sup>7</sup>

Dentro de la ingeniería del software una aplicación web es aquella donde el usuario tiene acceso a un espacio de comunicación a través de un servidor web que se puede encontrar en una conexión de internet o intranet. Esta herramienta para su codificación utiliza lenguajes de programación soportados por los navegadores web.

Son prácticas y de gran utilidad en la dinámica comunicativa Cliente-Servidor. La gran versatilidad de desarrollo de las aplicaciones permite la actualización de las mismas por medio del internet sin necesidad de instalar un software específico o versiones individuales por usuario.

---

<sup>7</sup> <https://www.scribd.com/doc/136052164/APLICACIONES-WEB-pdf>

Las aplicaciones se conforman por páginas desarrolladas en formato HTML o XHTML, que son las más comunes para el soporte de los navegadores. Los lenguajes de programación por su parte se los utiliza para añadir elementos dinámicos en la interfaz del usuario. Ordinariamente las páginas web se las envía al consumidor como un componente estático sin embargo la secuencia de las mismas generará un sentido interactivo promoviendo mayor transmisión en la comunicación.

### **1.5.3.1 Ventajas<sup>8</sup>**

Dentro de las ventajas se encuentran:

- Rapidez en la realización de trabajos a distancia.
- Los elementos necesarios para el desarrollo de la aplicación web son un buen navegador web y conexión a internet.
- No se requiere conocimiento previo del funcionamiento de una aplicación web
- Permite la centralización de todas las áreas de trabajo.
- La compatibilidad de las aplicaciones web son multiplataforma, es decir, el procesos de compatibilidad son mucho más sencillo que las aplicaciones de software descargable.
- Las actualizaciones de las aplicaciones web son constantes según la última versión publicada.
- No requieren de procesos de descarga, instalación y configuración por lo que su acceso se lo hace de forma inmediata a través de una cuenta online sin distinción de algún hardware o una configuración previa.
- Mayor demanda de memoria RAM que los programas instalados de forma local.

---

<sup>8</sup> <https://www.scribd.com/doc/136052164/APLICACIONES-WEB-pdf>

- Son menos propensas a fallos o conflictos de software por lo que no se cuelgan o colapsan fácilmente. Como todas las aplicaciones web utilizan la misma versión, todos los bugs pueden ser encontrados fácilmente y ser corregidos.
- Es multidinámico, es decir, varios usuarios pueden acceder a la aplicación web al mismo tiempo.

### **1.5.3.2 Seguridades para una aplicación Web<sup>9</sup>**

La seguridad de una aplicación web es un tema muy amplio porque se requiere la comprensión de todas las vulnerabilidades existentes. A continuación, se describen algunas funciones para contrarrestar amenazas:

- **Recomendaciones generales de seguridad**

Para evitar el acceso de usuarios malintencionados se sugiere seguir las siguientes instrucciones:

- Archivar copias de seguridad en un lugar seguro
- Adecuar al servidor en un lugar físico seguro y evitar dejarlo en espacios de acceso a cualquier usuario.
- Utilizar preferentemente el sistema de archivos NTFS de Windows.
- La creación de contraseñas rigurosas para proteger los equipos de la red y del servidor Web.
- Proteger los servicios de IIS o los servicios de información de internet para aumentar considerablemente la seguridad de la infraestructura web con la desactivación de varias funciones opcionales contra posibles ataques al servidor IIS.
- Los puertos no utilizados se los deben cerrar o desactivar servicios que no se utilicen.

---

<sup>9</sup> [https://msdn.microsoft.com/es-es/library/zdh19h94\(v=vs.100\).aspx#cpconbestsecuritypracticesforwebapplicationsanchor3](https://msdn.microsoft.com/es-es/library/zdh19h94(v=vs.100).aspx#cpconbestsecuritypracticesforwebapplicationsanchor3)

- Contar con un antivirus que monitoree la entrada y salida
- Delimitar una política que prohíba a los usuarios la constatación escrita de sus contraseñas en lugares de fácil acceso.
- Establecer el uso de un firewall, que permita delimitar o bloquear el tráfico entrante o saliente.
- Realizar la instalación de las últimas revisiones de seguridad de Microsoft o de cualquier otro fabricante para actualizarse en los últimos boletines de seguridad publicados.
- Usar las funciones de registro de Windows para conocer el estado de entrada de usuarios, actividades sospechosas o el número de solicitudes que se encuentran en el servidor Web.

- **Ejecutar aplicaciones con privilegios mínimos**

Para la ejecución de los privilegios mínimos en las aplicaciones web se sugiere lo siguiente:

- Evitar la ejecución de la aplicación con el nombre de un usuario de sistema o administrador.
- La ejecución de la aplicación debe contener el mínimo de privilegios factibles para un usuario.
- Definir una lista de control de acceso o ACL en todos los recursos y con una configuración poco descifrable.
- Ubicar la carpeta de los archivos de la aplicación debajo de la raíz evitando generar rutas de acceso a los usuarios.

- **Conocer a los usuarios**

Cuando el acceso a la aplicación se lo hace de forma anónima. Se recomienda restringir el acceso solamente a aquellos usuarios que se hayan identificado previamente y para ello se deben seguir las siguientes instrucciones:

- Sí, la aplicación se encuentra ejecutada por medio de una intranet, se sugiere utilizar la suplantación ASP.NET de Windows, cuando se hace uso de esta herramienta se puede ejecutar la cuenta de usuario según la solicitud emitida.
- Así también, con ASP.NET permitirá recabar las credenciales del usuario.

- **Protegerse contra entradas malintencionadas**

La entrada de los usuarios no siempre es segura porque varios usuarios malintencionados podrían enviar información potencialmente peligrosa para evitar esta agravante se recomienda seguir los siguientes parámetros:

- En la herramienta ASP.NET se puede filtrar las entradas de los usuarios mediante la verificación de las etiquetas HTML que podrían contener un Script.
- Codificar las HTML de tal forma que las Scripts sean visibles y no ejecutables.
- Se recomienda el uso de una base de datos antes de almacenar la información emitida por un usuario para poder filtrarla.
- Cuando sea requerido la aceptación de un archivo con codificación HTML por parte de un usuario se debe realizar un filtro de forma manual en donde conste con la aprobación de los elementos a aceptar.
- Las afirmaciones de que el encabezado de solicitud HTTP es segura, representan un gran error. Debe proteger todas las cadenas de consulta, cookies, etc. Tomando en consideración que el explorador puede enviar información al servidor que puede ser suplantada.

– No almacene información confidencial en un espacio de acceso libre como en campos ocultos o cookies.

- **Tener acceso seguro a la base de datos**

A pesar de que cada base de datos tiene su propia seguridad, es importante, definir un mecanismo adecuado de acceso de la aplicación hacia la base de datos, para ello, se puede seguir los siguientes lineamientos:

– La seguridad inherente de la base de datos permitirá limitar el acceso a la misma. La seguridad sobre la base de datos dependerá de la connotación que tenga la aplicación

- ✓ En la aplicación se podrá utilizar la seguridad integrada para autenticar a los usuarios que tenga acceso a la base de datos.

- ✓ Si la aplicación hace referencia a un acceso de forma anónima, sería factible la creación un único usuario que posee limitados permisos y la pueda realizar consultas una vez conectado con su cuenta.

– Evitar la creación de instrucciones SQL con información resultante de los usuarios, es recomendable la creación de una consulta bajo parámetros y valores que pueden ser establecidos mediante la entrada de cada usuario.

– Si se desea almacenar los nombres de usuarios y contraseñas como credenciales se recomienda realizarlo en un archivo cuya configuración establezca la seguridad necesaria.

- **Crear mensajes de error seguros**

La información presentada en los mensajes de error puede ser usado por usuarios malintencionados y mediante deducciones adquirir datos importantes que pondrían en riesgo a la aplicación, para evitar estos riesgos se debe considerar las siguientes instrucciones:

- Emitir información simple por medio de los mensajes de error para evitar el uso de la misma por usuarios malintencionados, como por ejemplo no incluir el nombre de usuario.
- Evitar la información detallada, en el caso de necesitar con el cumplimiento de este requerimiento, confirmar primero cual será el usuario local que recibirá la información con respecto al servidor Web.
- Se puede utilizar el elemento de configuración `customErrors` para evidenciar quien puede apreciar las excepciones desde el servidor.
- Se considera crear un sistema que administre las situaciones propensas a errores y al acceso de la base de datos.

- **Mantener segura la información confidencial**

Para evitar el acceso de la información confidencial por parte de un usuario malintencionado y que se vean expuestos los datos de la aplicación siga las siguientes instrucciones:

- Cuando la información confidencial se encuentra distribuida entre el servidor y el explorador se recomienda la utilización de un protocolo de seguridad Secure Socket Layer, el mismo, permite que todos los datos estén seguros mediante el proceso de cifrado o encriptado.
- Procure proteger la información confidencial en archivos de configuración `Web.config` o `Machine.config` que le permitirá cifrar los datos en un archivo.
- El almacenamiento de la información confidencial no debe generarse en una página Web ni siquiera en un código de servidor.
- Hacer uso directo de los algoritmos de cifrado de alta seguridad cuyos nombres pueden ser `System.Security.Cryptography`.

- **Usar cookies de forma segura**

Las cookies representan un mecanismo de almacenamiento de información específica referente a los usuarios, como estos datos son enviados al explorador son expuestos y vulnerables pudiendo llegar a ser suplantados o darles un uso malintencionado, para evitar esta acción se podrían delimitar los siguientes aspectos:

- Evite el almacenamiento de información importante o confidencial en una cookie inclusive temporalmente, la suplantación de la misma podría verse inmersa en el inadecuado funcionamiento de la aplicación. Los datos recomendables que se sugieren guardar en una cookie puede ser la referencia de ubicación del servidor en la que se encuentra la información.
- Evitar las cookies permanentes de ser posible, es decir, delimite tiempos de expiración
- Cifrar toda la información de las cookies
- Tomar en consideración las opciones Secure y HttpOnly como true.

- **Protegerse contra amenazas de denegación de servicio**

Una forma en la que un usuario malintencionado puede afectar a la aplicación es manteniéndola saturada evitando el uso a otros, bloquearla o comprometerla para que no se encuentre habilitada, esta acción puede ser suprimida mediante los siguientes aspectos:

- Se recomienda el uso de controles de error que incluya un bloque final para de esta manera liberar recursos cuando se detecte algún error.
- La configuración de los servicios IIS para regular los procesos, generará un control adecuado que evitará el uso excesivo de la aplicación en tiempo con respecto al CPU.
- Previo a la utilización y uso de la entrada del usuario verifique los límites que posee.

- Proporcione límites del tamaño de consultas para la base de datos.
- Se considera establecer un límite sobre las cargas de archivos, mediante Visual Studio.

#### 1.5.4 Servidor web<sup>10</sup>

La web surge en el año de 1989 con un proyecto estructurado por la CER (Consejo Europeo para la Investigación Nuclear), la finalidad fue la de compartir de una manera más dinámica los archivos de texto entre científicos y que se pueda revisar las referencias de los artículos mediante la creación de un sistema de hipertexto.

La evolución de la web tuvo incidencia en la generación de mayor recurso a las páginas haciendo de ella no solamente un archivo de texto sino que además se le incluyeron opciones como gráficos, contenido dinámico, música y animaciones.

Así la evolución de la web se basaría en tres estándares descritos como:

- **El localizador uniforme de recursos (URL):** referente a la localización de las páginas de información en una sola dirección.
- **El protocolo de transferencia de hipertexto (HTTP):** determina la comunicación que debe existir entre el navegador y el servidor a través de un método que incluye peticiones y respuestas.
- **El lenguaje de marcación de hipertexto (HTML):** se refiere al método que codifica la información procedente de documentos y sus enlaces.

El servicio web con el internet determina su funcionamiento mediante la transmisión de páginas web. Cuando las mismas están almacenadas en un solo o varios ordenadores se le denominan servidor web. Este servicio, procede de acuerdo al método cliente-

---

<sup>10</sup> <http://elara.site.ac.upc.edu/documentacion/INTERNET%20-%20UD8%20-%20Protocolo%20HTTP%20y%20servidores%20WEB.pdf>

servidor. Donde el servidor representa al ente que presta el servicio y el cliente es quien recibe el mismo

#### **1.5.4.1 Características de un servidor web<sup>11</sup>**

Dentro de las características que posee el servicio web en el proceso de funcionamiento son las siguientes:

- Una vez obtenidas las peticiones de los clientes el servidor ofrece respuestas enviando ficheros de texto que pueden ser planos (html, php) o binarios (gif, jpeg)
- Las peticiones son distribuidas en puertos 80 para HTTP y 443 para HTTPS.
- La atención de peticiones se basa en un proceso donde el servidor al encontrar el archivo solicitado lo transmite y caso contrario emite un mensaje de error.
- El servidor web determina si el usuario es apto o tiene los permisos necesarios para el acceso a los documentos

#### **1.5.4.2 Proceso de transferencia**

- El usuario especifica la página que desea consultar mediante la URL en el cliente web
- El cliente a su vez determina la conexión con el servidor mediante la página solicitada
- Posteriormente el servidor busca la página solicitada en su sistema de ficheros si es encontrada se transfiere caso contrario emite un mensaje de error.
- Con la transferencia realizada el cliente interpreta el código HTML y procede a mostrar la página al usuario.
- Finalmente, al cerrar la transmisión esta siempre es liberada.

---

<sup>11</sup> <http://elara.site.ac.upc.edu/documentacion/INTERNET%20-%20UD8%20-%20Protocolo%20HTTP%20y%20servidores%20WEB.pdf>

#### **1.5.4.3 Modelo transaccional<sup>12</sup>**

- Utiliza el modelo de transacciones donde, el cliente realiza una petición al servidor y atendida la misma se cierra la comunicación de forma inmediata.
- La transacción es independiente para cada documento u objeto que se transmite
- Entre peticiones sucesivas no se mantiene una memoria como varios ejemplos se encuentran el acceso a los bancos, sitios seguros, etc.
- Esta problemática puede verse resuelta gracias a cookies o el acceso a una base de datos

#### **1.5.4.4 Localización de servidores web**

- El servidor puede encontrarse en varios lugares uno de ellos es dentro de la misma red de área local donde se ubica el cliente web conocido como servidor local o LAN, uno ejemplo de ello es el servidor web intranet.
- Servidores remotos, contrario al área local de la máquina cliente. Un ejemplo de esta clasificación es [www.google.com](http://www.google.com).

#### **1.5.4.5 Cliente web**

- Es aquel programa donde el usuario puede interactuar mediante el envío de páginas web a un servidor. Estas páginas generalmente vienen codificadas en formato HTML, la función del cliente es interpretarlas de forma que llegue al usuario en un formato adecuado.
- En otro caso cuando el documento recibido es un objeto multimedia sea video o sonido, el cliente habilita una aplicación externa que sea capaz de gestionarlo.
- Utiliza protocolos HTTP o FTP

---

<sup>12</sup> <http://elara.site.ac.upc.edu/documentacion/INTERNET%20-%20UD8%20-%20Protocolo%20HTTP%20y%20servidores%20WEB.pdf>

- Los clientes web o navegadores más utilizados son Netscape, IE explorer, Mozilla, Chrome.

#### 1.5.4.6 Páginas dinámicas<sup>13</sup>

Son aquellas páginas creadas al instante, su proceso combina una plantilla y lo relaciona con la consulta realizada a una base de datos

Para la construcción de las páginas el servidor web le cede el control al servidor de aplicaciones quien a su vez regresa la información una vez creada la página como se muestra en la siguiente gráfica

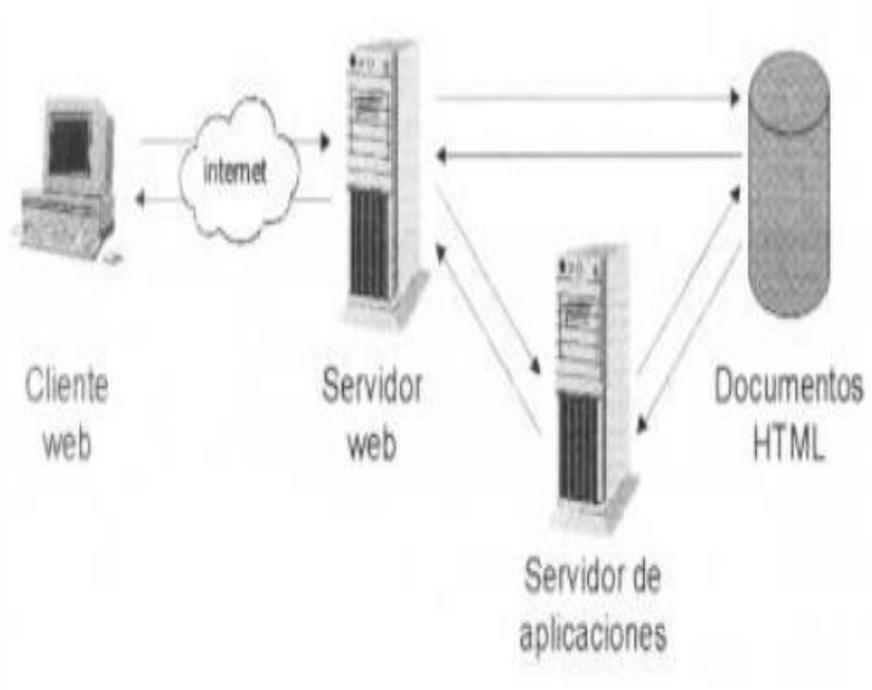


Gráfico N° 1. Proceso de construcción de páginas dinámicas

Fuente: <http://elara.site.ac.upc.edu/documentacion/INTERNET%20-%20UD8%20-%20Protocolo%20HTTP%20y%20servidores%20WEB.pdf>

<sup>13</sup> <http://elara.site.ac.upc.edu/documentacion/INTERNET%20-%20UD8%20-%20Protocolo%20HTTP%20y%20servidores%20WEB.pdf>

### 1.5.4.7 Protocolo HTTP<sup>14</sup>

Representa el protocolo o procesos que se utilizan para cada transacción de la web (www), es decir por medio de HTTP se promueven los accesos, respuestas e información de las páginas web y se lo hace mediante el uso de Hipertexto (HTML).

Es un protocolo meramente transaccional por lo que no almacena información sobre conexiones anteriores.

Su modelo basado en la modalidad cliente-servidor, define el proceso donde el cliente HTTP realiza una solicitud al servidor, este último la gestiona, finaliza la actividad se cierra la conexión y los datos se pierden. Por tal motivo se han creado las herramientas conocidas como cookies, que representan pequeños ficheros almacenados en el propio ordenador, donde al establecer una conexión se puede leer una página web y así reconocer a un visitante antiguo.

En solicitudes y respuestas la línea inicial es diferente como se muestra a continuación:

- **Solicitud HTTP**

Dentro de las solicitudes se aprecian campos separados por un espacio en blanco como se muestra:

Método          recurso          versión\_del\_protocolo

P.e: “GET /recursos/notas/notas.html HTTP/1.1”

- **Respuesta HTTP**

Existen 3 campos separados por un espacio en la línea inicial de respuesta como se muestra:

Versión\_del\_protocolo          codigo\_respuesta mensaje

P.e.: “ HTTP/1.0 200 Ok” ó “HTTP/1.1 404 Not found”

---

<sup>14</sup> <http://elara.site.ac.upc.edu/documentacion/INTERNET%20-%20UD8%20-%20Protocolo%20HTTP%20y%20servidores%20WEB.pdf>

### 1.5.5 Apache<sup>15</sup>

Apache representa uno de los servidores web HTTP más populares y robustos dentro del mercado. Además utiliza un código abierto, es gratuito, representa un servidor multiplataforma con un gran rendimiento y seguridad.

Se encuentra desarrollado dentro del proyecto HTTP Sever (httpd) de la Apache Software Foundation.

Su inicio se remonta al año 1995, con el proyecto basado en el servidor Apache httpd, cuya aplicación fue denominada NCSA. Dentro del grupo de trabajo en el emprendimiento se marchó Rob McCool uno de los más importantes integrantes por lo que se frenó el plan. Pero fueron Brian Behlendorf y Cliff Skolnick quienes tomaron las riendas y por medio de una lista de correo gestionaron todo el trabajo promoviendo un espacio libre en el acceso para desarrolladores.

Para el año 1999, con estos antecedentes se apertura la Fundación de Software Apache con la finalidad de obtener ayuda de carácter financiera, organizativa y legal para la composición del servidor

### 1.5.6 Características<sup>16</sup>

- Puede ser ejecutable en varios sistemas operativos como: Unix, FreeBSD, Linux, Solaris, Novell NetWare, OS X, Microsoft Windows.
- Permite el uso de múltiples lenguajes de programación script sean estos: PHP, Perl, Tcl, Python
- Soporta plataformas como J2EE (Java™ 2 Platform, Enterprise Edition), con la herramienta tomcat.

---

<sup>15</sup> <http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>

<sup>16</sup> <https://www.fdi.ucm.es/profesor/jpavon/web/31-ServidoresWeb-Apache.pdf>

- Puede contener hosting virtual, es decir, consigue hacer funcionar varias direcciones IPs y nombres de dominio con un mismo servidor físico.
- Cuenta con un diseño modular, donde se pueden crear nuevos módulos con el API
- Su configuración es sencilla, se basa en directivas modificables por medio de ficheros como: httpd.conf, access.conf (en Unix), .htaccess, .htpasswd
- La instalación permite el entorno XAMPP que representa uno de los más populares con desarrollo PHP.

Por todas las características mencionadas y demás Apache representa uno de los servidores más usados en el mercado con respecto a otros como se muestra en la siguiente ilustración

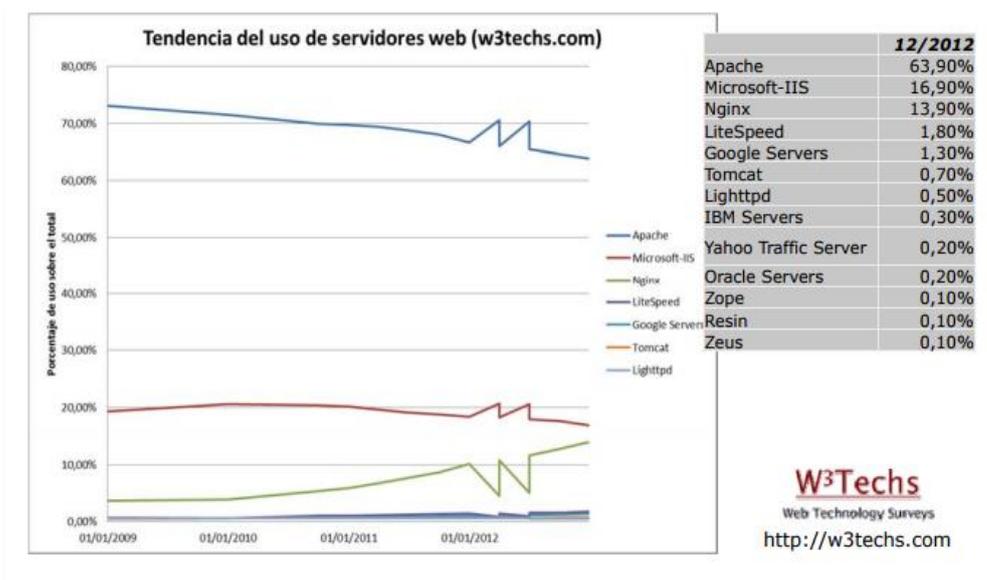


Gráfico N° 2. Tendencia de uso de servidores web

Fuente: <https://www.fdi.ucm.es/profesor/jpavon/web/31-ServidoresWeb-Apache.pdf>

### 1.5.6.1 Esquemas de funcionamiento del servidor web Apache<sup>17</sup>

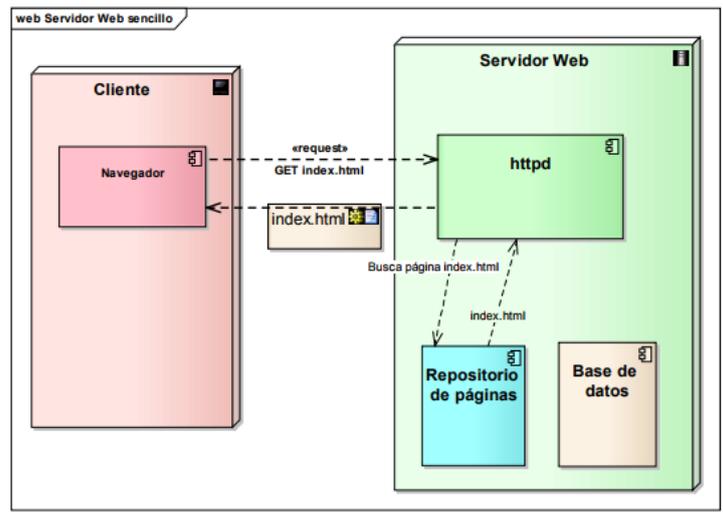


Gráfico N° 3. Esquema 1 http

Fuente: <https://www.fdi.ucm.es/profesor/jpavon/web/31-ServidoresWeb-Apache.pdf>

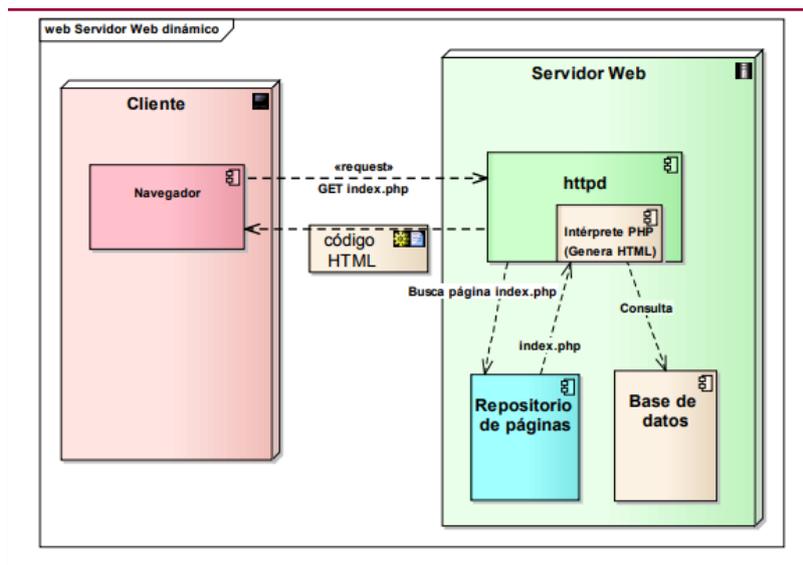


Gráfico N° 4. Esquema 2 interpretado con PHP

Fuente: <https://www.fdi.ucm.es/profesor/jpavon/web/31-ServidoresWeb-Apache.pdf>

<sup>17</sup> <https://www.fdi.ucm.es/profesor/jpavon/web/31-ServidoresWeb-Apache.pdf>

## 1.6 PROGRAMACIÓN ORIENTADA A OBJETOS<sup>18</sup>

Este tipo de programación pretende mediante la actualización de conceptos mejorar la programación estructurada facilitando la tarea del programador mediante la creación de un paradigma enfocado en la representación del mundo real a través de un lenguaje de programación. Para lograrlo, se apoya en unidades auto contenidas a las cuales se las denomina objetos

### 1.6.1 Evolución de los Lenguajes de Programación

**Primera etapa, lenguajes ensambladores.** - Usa lenguajes de bajo nivel en los cuales se manejan instrucciones compuestas por un operador y sus operandos. No existe abstracción de datos en esta etapa.

**Segunda etapa, lenguajes estructurados.** - Aparecen los lenguajes de programación de alto nivel, donde el diseño del software se enfoca en representar los procedimientos en función del lenguaje elegido. Aparecen nuevos conceptos tales como: refinamiento progresivo, modularidad, procedimientos y programación estructurada. La abstracción de datos en esta etapa es mayor.

**Tercera etapa.** - Los conceptos de atracción y ocultación se visualizan con mayor facilidad en esta etapa.

**Cuarta etapa.** - Al entrar en la etapa cuarta, sabiendo que los problemas del mundo real se representan mediante objetos, aquí se aplica la abstracción de datos, siendo estos de gran importancia.

---

<sup>18</sup> <https://compu2poo.wordpress.com/>

## 1.6.2 Conceptos de la Programación Orientada a Objetos<sup>19</sup>

**Abstracción:** Es el proceso que permite representar una entidad, la misma que puede realizar una acción o tarea, recibir, procesar o producir información mediante la interacción con otras entidades. Se puede abstraer sus distintos elementos y al hacerlo se extiende la funcionalidad de los mismos.

**Encapsulamiento:** Los elementos que componen una entidad pueden agruparse según características comunes, es decir, tienen el mismo nivel de abstracción. Mediante esto se incrementa la relación existente entre los componentes de un sistema.

**Modularidad:** Un sistema puede ser dividido en pequeñas partes, llamados módulos, cada uno de los cuales mantiene su independencia de los demás, pero que interactúan entre sí.

**Principio de ocultación:** Para proteger la integridad de los objetos se impide el libre acceso a ciertas partes de los mismos, pero se implementa mecanismos controlados que permite a otros objetos acceder, leer o modificarlos solamente si se posee el permiso adecuado. Únicamente los métodos del propio objeto pueden acceder a su información, previniendo efectos inesperados.

**Polimorfismo:** Es la capacidad que posee un objeto de comportarse de formas distintas dependiendo la manera que es invocado, es decir, un mismo método puede tener varias implementaciones que durante la ejecución del programa serán ejecutadas según sean necesarias.

**Herencia:** Las propiedades y el comportamiento de un objeto pueden ser heredados por otro. Los objetos pueden ser organizados tomando en cuenta su funcionalidad y

---

<sup>19</sup> <https://algonzalezpoo.wordpress.com/caracteristicas-de-poo/>

características comunes, esto permite implementar objetos básicos de los cuales extender hacia objetos especializados.

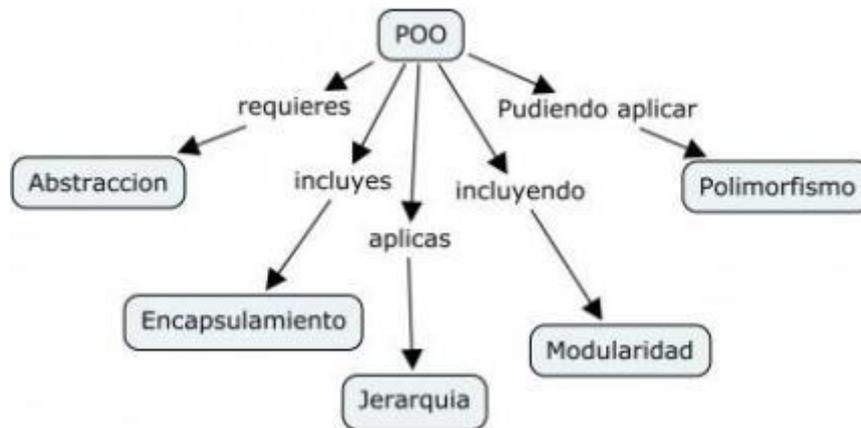


Gráfico N° 5. Características de la POO

Fuente: <https://compu2poo.wordpress.com/>

### 1.6.3 Características del Lenguaje Orientados a Objetos<sup>20</sup>

Los lenguajes de programación orientados a objetos, deben contener las siguientes características:

**Tipificación fuerte.** - El tipo de datos que soportará cada variable debe ser declarado explícitamente durante las fases de diseño e implementación.

**Manejo de excepciones.** - Debe establecer mecanismos para detectar y manipular excepciones que pueden presentarse durante la ejecución de un programa.

**Paso de mensajes.** - Debe soportar el paso de mensajes bidireccionales entre los módulos que componen un programa.

**Generalidad.** - Debe permitir la reutilización de clases generales, las cuales contengan parámetros formales que serán instanciados por parámetros reales.

<sup>20</sup> <https://cesarlagg.files.wordpress.com/2016/09/programacion-estructurada1.pptx>

**Multitarea.** - Debe permitir la creación de procesos simultáneos en tiempo de ejecución sin depender el sistema operativo que sea utilizado.

**Persistencia.** - En caso de ser requerido, los objetos creados pueden permanecer en memoria después de la ejecución de un programa.

**Datos compartidos.** - Debe permitir el acceso al mismo espacio de memoria a objetos distintos (memoria compartida) o permitir la comunicación mediante mensajes.

### **1.6.3.1 Ventajas de la Tecnología Orientada a Objetos<sup>21</sup>**

- Al establecer módulos independientes cada uno con su propia definición de clases, brinda la flexibilidad de definir nuevas clases a partir de ellos y construir así los programas.
- Gracias a la flexibilidad de la Programación Orientada a Objetos se puede reutilizar las clases definidas en programas anteriores y aprovechar la funcionalidad previamente definida.
- Los programas tienen mejor mantenibilidad, es decir, se puede realizar cambios a una clase en particular sin afectar el comportamiento general de la aplicación.
- Debido a la extensibilidad, modularidad y herencia, las aplicaciones pueden ser extensibles y así cubrir nuevas necesidades sin mayores dificultades.

### **1.6.3.2 Desventajas de la tecnología orientada a objetos**

- Se requiere un amplio conocimiento del paradigma por parte del programador, para aprovechar todas sus características.
- Las aplicaciones creadas bajo este paradigma suelen tener un tamaño excesivo.

---

<sup>21</sup> <https://objetosweb.wordpress.com/2016/05/18/ventajas-y-desventajas-de-la-programacion-orientada-a-objetos/>

- Un mismo problema puede ser resuelto de maneras distintas, esto se complica cuando se tiene un equipo de trabajo diverso.
- Al heredar clases ya existentes, se hereda también todos los miembros de esta, aun cuando no sean utilizadas.
- Las aplicaciones pueden llegar a ser innecesariamente pesadas, y por lo tanto su ejecución volverse más lenta.

## **1.7 PHP<sup>22</sup>**

PHP apareció a inicios de 1994, cuando Rasmus Lerdorf, necesitaba solucionar sus problemas de sobrecarga en el servidor web.

PHP se encuentra entre los más potentes lenguajes, junto con su módulo de servidor o como binario CGI, el cual, puede acceder, ejecutar y abrir conexiones, desde su módulo servidor, gracias a que se encuentra dentro de los lenguajes de código abierto

Durante el diseño de PHP se pensó que sea un lenguaje más seguro que Perl o C, partiendo de un correcto ajuste de las opciones de configuración en tiempo de ejecución y en tiempo de compilación. El uso de prácticas de programación adecuadas puede proporcionarle la combinación de libertad y de seguridad que necesita el programador.

### **1.7.1 Importancia<sup>23</sup>**

En el mundo cada día es mayor la presencia de Internet dentro del entorno y vida diaria. Gracias a este crecimiento también el área de desarrollo de proyectos web se ha incrementado. Un proyecto web puede ser un pequeño paso para un gran emprendimiento, y elegir la tecnología adecuada para su desarrollo será la diferencia entre el éxito o fracaso de una idea.

---

<sup>22</sup> <http://www.maestrosdelweb.com/rasmus-lerdorf-php/>

<sup>23</sup> <https://fergarciaac.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>

El desarrollo de páginas web con PHP se ha convertido en uno de los más importantes, por ser un lenguaje libre y abierto que proporciona una curva de aprendizaje muy bajo, que cumple con estándares básicos de la programación orientada a objetos. Gracias a esto no se necesita de lenguajes con entornos complejos, este lenguaje se puede programar en un simple de bloc de notas, la configuración de su entorno es fácil y de costos prácticamente gratuitos.

Este lenguaje de programación es fácil de desplegar porque dispone de paquetes totalmente autoinstalables que integran PHP: como son Apache y MySQL, tanto para UNIX (LAMPP), como para Windows (XAMPP).

### 1.7.2 Características <sup>24</sup>

Las características principales de PHP que lo convierten en uno de los más potentes son:

- **Gran extensión de documentación.** - Al tener una gran comunidad, su soporte es mayor a la de otros lenguajes, por lo cual PHP es usado en la actualidad por millones de usuarios.
- **Variedad de herramientas para aprender.** - Este lenguaje es de fácil aprendizaje no solo permite contratar un servidor remoto para subir archivos, sino que también se puede crear un servidor local con una máquina virtual.
- **Mucha oferta de trabajo.** - tiene como objetivo facilitar la creación de webs complejas, por lo que muchas empresas requieren los servicios de un desarrollador con conocimientos en este lenguaje.

---

<sup>24</sup> <http://www.programandolaweb.info/posts/php/6/Ventajas-y-desventajas-en-el-uso-de-php.html>

- **Permite programación orientada a objetos.** - Otra de las características es que podemos dividir los scripts en métodos, clases, etc. para hacer más ágil de cara al servidor el procesamiento de los datos.
- **Módulos externos para mejorar la aplicación web.** - Podemos implementar funcionalidades, que de por sí no existen, siendo una gran solución a las necesidades del usuario.
- **Se puede separar la estructura.** - Ya que podemos tener de manera independiente el código que se encarga de mover los datos para enlazarlos a la interfaz. Esto se conoce como Modelo Vista/Controlador (MVC). Gracias a esta característica, no “contaminaremos” el código con líneas innecesarias y lo mantendremos limpio y ordenado.
- **La libertad es un bien muypreciado.** - Al poder contratar un servidor remoto para subir nuestros archivos, nos permite que los recursos se encuentren en nuestro sitio dentro de la red, esto lo logramos creando un servidor local con una máquina virtual.

### 1.7.2.1 Ventajas y desventajas<sup>25</sup>

#### Ventajas

- Lenguaje totalmente libre y abierto.
- Curva de aprendizaje muy baja.
- Los entornos de desarrollo son de rápida y fácil configuración.
- Fácil despliegue: paquetes totalmente autoinstalables que integran PHP.
- Fácil acceso a bases de datos.
- Comunidades de desarrollo muy grande.

---

<sup>25</sup> <http://www.programandolaweb.info/posts/php/6/Ventajas-y-desventajas-en-el-uso-de-php.html>

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Su código es totalmente transparente, de esto se encarga el servidor, el cual ejecuta, y envía el resultado en forma de una HTML al usuario.
- Existe una gran conexión a una variedad de motores para bases de datos.
- Capacidad de expandir su potencial utilizando módulos (llamados extensiones).
- Gracias a que cuenta con una comunidad extensa, su sitio siempre cuenta con una amplia documentación, y archivos de ayuda gratuitos.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Su flexibilidad al momento de programar, no exige una determinada metodología, esto permite que el programador aplique sus técnicas de desarrollo.

#### **1.7.2.2 Desventajas<sup>26</sup>**

- Al ser un lenguaje que se interpreta en la ejecución existen ciertos casos en los cuales su código no puede ser ocultado.
- Los scripts de PHP suelen considerarse lentos, a comparación de otros scripts.
- En las versiones previas a la 7, las variables no son tipificadas, lo cual dificulta a los diferentes IDEs ofrecer asistencias para el tipificado del código.
- Considerando que PHP es un lenguaje que se ejecuta en el servidor esto lo convierte en un lenguaje inseguro.
- Al tener una variedad de opciones garantizadas en su ejecución, lo lleva también a tener una configuración menos segura.

---

<sup>26</sup> <http://www.programandolaweb.info/posts/php/6/Ventajas-y-desventajas-en-el-uso-de-php.html>

### 1.7.3 Estructuras del Código Fuente en Php

#### 1.7.3.1 Aspectos Sobre la Sintaxis del Lenguaje<sup>27</sup>

En la programación con PHP se debe tener conocimiento de programación en HTML (o XHTML), pero al no ser una programación de nivel alto se puede ir aprendiendo con ejemplos y con autoaprendizaje. En este aspecto se inicia la sintaxis considerando algunos datos al momento de la escritura del código como son:

- No es Key Sensitive, a diferencia de otros lenguajes similares, permite que las sentencias se escriban con cualquier combinación de mayúsculas o minúsculas a excepción de las variables.
- Con el objeto que el código fuente sea más legible, los espacios en blanco y cambios de línea, no están limitados por el lenguaje ya que el intérprete de PHP los ignora.
- En la programación toda línea culmina con “;”, cuando tenemos el terminador ¿> no es necesario que antes termine con “;”, esto se aplica para estructuras condicionales o bucles
- En la programación el código puede partirse en diferentes líneas esto mejora la legibilidad de las instrucciones.
- Uno de los aspectos más notables es que PHP permite la programación, en una misma línea sí así se desea, varias instrucciones diferentes separadas por “;”.

---

<sup>27</sup> Cobo, A. (2005). PHP y MySQL: Tecnología para el desarrollo de aplicaciones web. 6: Ediciones Diaz de Santos.

## 1.7.4 Elementos Básicos del Lenguaje PHP<sup>28</sup>

### 1.7.4.1 Tipos de Datos

PHP utiliza varios tipos de datos ya sean primitivos o especiales:

- **Datos Primitivos**

- **BOOLEAN** .- Con valores simples como TRUE (verdadero) o FALSE (falso).
- **INTEGER**.- Este tipo de datos se expresan en base 10, 16, 8 o 2, y pueden estar precedidos por un signo (- o +).
- **FLOAT**.- Son números conocidos como punto flotante los cuales se pueden definir también como:
  - STRING
  - ARRAY
  - OBJECT
  - CALLABLE

- **Datos Especiales**

- **RESOURCE** .- Son recursos que nos permiten crear y usar funciones específicas
- **NULL** .- Son variables que no contiene valor y se considera como NULL si:
  - Se le ha asignado la constante NULL.
  - No se le ha asignado un valor todavía.
  - Se ha destruido con UNSET().

---

<sup>28</sup> <http://php.net/manual/es/language.operators.php>

### 1.7.4.2 Variables

Variable se llama a toda letra del alfabeto que se utiliza en programación para dar un valor X, estas variables se representan en este lenguaje con un signo de dólar seguido por el nombre de la variable (\$A). Los nombres de variables siguen las mismas reglas que las otras etiquetas en PHP. Las variables son válidas si empiezan con una letra o un carácter de subrayado (underscore), seguido de cualquier número de letras, números y caracteres.

```
<?php
$var = 'Roberto';
$Var = 'Juan';
echo "$var, $Var";      // imprime "Roberto, Juan"

$4site = 'aun no';     // inválido; comienza con un número
$_4site = 'aun no';   // válido; comienza con un carácter de subrayado
$täyte = 'mansikka';  // válido; 'ä' es ASCII (Extendido) 228
?>
```

Gráfico N° 6. Variables

Fuente: <http://php.net/manual/es/language.operators.php>

### 1.7.4.3 Variables Predefinidas

Dichas variables dependen del servidor, las mismas, no están totalmente documentadas, y dependen del script que se ejecute.

- Superglobals
- \$GLOBALS
- \$\_SERVER
- \$\_GET
- \$\_POST
- \$\_FILES
- \$\_REQUEST

- \$\_SESSION
- \$\_ENV
- \$\_COOKIE
- \$php\_errormsg
- \$HTTP\_RAW\_POST\_DATA
- \$http\_response\_header
- \$argc
- \$argv

#### 1.7.4.4 Constantes

Se denomina constante a un valor que tiene como característica que no se puede modificar su valor. Una constante es un identificador para expresar un valor simple, este valor no puede variar durante la ejecución del script. Se puede definir una constante usando la función **define** () que permite definir un constante con una expresión arbitraria o con la palabra reservada **const** que tiene una restricción como se muestra a continuación.

```
<?php
// Funciona a partir de PHP 5.3.0
const CONSTANTE = 'Hola Mundo';

echo CONSTANTE;

// Funciona a partir de PHP 5.6.0
const OTRA_CONSTANTE = CONSTANTE.' Adiós Mundo';
echo OTRA_CONSTANTE;

const ANIMALES = array('perro', 'gato', 'pájaro');
echo ANIMALES[1]; // muestra "gato"

// Funciona a partir de PHP 7
define('ANIMALES', array(
    'perro',
    'gato',
    'pájaro'
));
echo ANIMALES[1]; // muestra "gato"
?>
```

Gráfico N° 7. Constantes

Fuente: <http://php.net/manual/es/language.operators.php>

Las características más importantes de una constante son:

- Los identificadores de constantes siempre suelen declararse en mayúsculas
- Las constantes tienen las mismas reglas que otras etiquetas
- Una constante válida, es aquella que empieza con una letra seguida del sub-guion, y una combinación de letras, números y sub-guiones.

#### 1.7.4.5 Operadores

Este tipo de elemento puede tomar 1 o más valores en la jerga de programación y produce otro valor que la construcción de sí misma se convierte en una expresión.

Los operadores se pueden agrupar de acuerdo con el número de valores que toman:

- **Operadores Unarios.** - Son aquellos que toman sólo un valor como: ! (el operador lógico de negación) o ++ (el operador de incremento).
- **Operadores Binarios.** - Son aquellos que toman dos valores, como los familiares operadores: aritméticos + (suma) y - (resta), y la mayoría de los operadores de PHP entran en esta categoría.
- **Operador Ternario.** - Es aquel que toma tres valores; usualmente a este se le refiere simplemente como "el operador ternario", ? :, o conocido también como operador condicional.

##### 1.7.4.5.1 Precedencia de los operadores

En el siguiente gráfico se muestran los operadores en orden de precedencia, desde los de más alta precedencia al inicio hasta el de más baja. Los operadores en una misma línea tienen igual precedencia, en este caso la asociatividad decide el agrupamiento.

Precedencia de operadores		
Asociatividad	Operadores	Información adicional
no asociativo	<code>clone new</code>	<a href="#">clone</a> and <a href="#">new</a>
izquierda	<code>[</code>	<a href="#">array()</a>
derecha	<code>**</code>	<a href="#">aritmética</a>
derecha	<code>++ -- - (int) (float) (string) (array) (object) (bool) @</code>	<a href="#">tipos e incremento/decremento</a>
no asociativo	<code>instanceof</code>	<a href="#">tipos</a>
derecha	<code>!</code>	<a href="#">lógico</a>
izquierda	<code>* / %</code>	<a href="#">aritmética</a>
izquierda	<code>+ - .</code>	<a href="#">aritmética y string</a>
izquierda	<code>&lt;&lt; &gt;&gt;</code>	<a href="#">bit a bit</a>
no asociativo	<code>&lt; &lt;= &gt; &gt;=</code>	<a href="#">comparación</a>
no asociativo	<code>== != === !== &lt; &gt; &lt;=&gt;</code>	<a href="#">comparación</a>
izquierda	<code>&amp;</code>	<a href="#">bit a bit</a> y <a href="#">referencias</a>
izquierda	<code>^</code>	<a href="#">bit a bit</a>
izquierda	<code> </code>	<a href="#">bit a bit</a>
izquierda	<code>&amp;&amp;</code>	<a href="#">lógico</a>
izquierda	<code>  </code>	<a href="#">lógico</a>
derecha	<code>??</code>	<a href="#">comparación</a>
izquierda	<code>? :</code>	<a href="#">ternario</a>
derecha	<code>= += -= *= **= /= .= %= &amp;=  = ^= &lt;&lt;= &gt;&gt;=</code>	<a href="#">asignación</a>
izquierda	<code>and</code>	<a href="#">lógico</a>
izquierda	<code>xor</code>	<a href="#">lógico</a>
izquierda	<code>or</code>	<a href="#">lógico</a>

Gráfico N° 8. Precedencia de los operadores

Fuente: <http://php.net/manual/es/language.operators.php>

### 1.7.5 Sentencias de control

En PHP existen a lo que se llama sentencias de control es lo que permite, como su nombre lo define el control de la programación. Estas pueden ser la asignación, llamada también función, o ciclo, pueden ser también sentencias condicionadas o incluso una sentencia que no hacen nada, conocidas como sentencia vacía.

Generalmente se finaliza las sintaxis con “;”, las mismas que se pueden encapsular en corchetes.

#### 1.7.5.1 Sentencia if

Esta sentencia se evalúa con un valor booleano, si es TRUE la sentencia se ejecuta, en cambio si el FALSE esta sentencia es ignorada.

- **Sintaxis**

**if** (expr)

```
{  
  
    sentencia  
  
}
```

- **Ejemplo**

```
<?php  
if ($a > $b) {  
    echo "a es mayor que b";  
    $b = $a;  
}  
?>
```

Gráfico N° 9. Ejemplo sentencia if

Fuente: <http://php.net/manual/es/language.operators.php>

### 1.7.5.2 Sentencia else

Utilizada cuando se desea extender una sentencia if, en caso de que esta se evalué como FALSE.

- **Sintaxis**

**if** (expr)

```
{  
  
    sentencia  
  
}  
  
else  
  
    {  
  
        sentencia  
  
    }
```

- **Ejemplo**

```
<?php
if ($a > $b) {
    echo "a es mayor que b";
} else {
    echo "a NO es mayor que b";
}
?>
```

Gráfico N° 10. Ejemplo sentencia else

Fuente: <http://php.net/manual/es/language.operators.php>

### 1.7.5.3 Sentencia elseif/else if

A diferencia de IF ELSE esta sentencia es una expresión alternativa que solo se ejecutará si ELSE IF se evalúe como TRUE.

- **Sintaxis**

**if** (expr)

```
{
    sentencia
}
```

**elseif**

```
{
    sentencia
}
```

**else**

```
{
    sentencia
}
```

- **Ejemplo**

```
<?php
if ($a > $b) {
    echo "a es mayor que b";
} elseif ($a == $b) {
    echo "a es igual que b";
} else {
    echo "a es menor que b";
}
?>
```

Gráfico N° 11. Ejemplo de sentencia elseif

Fuente: <http://php.net/manual/es/language.operators.php>

#### 1.7.5.4 Estructuras de control

Como su nombre lo define estas estructuras son aquellas que permiten el control, a una aplicación, y señala el inicio y el final de una consulta en la programación.

Las estructuras de control, ofrecen sintaxis alternativas como: if, while, for, foreach, y switch. Para cada caso la forma básica de la sintaxis alternativa es cambiar el corchete de apertura por dos puntos (:) y el corchete de cierre por endif;, endwhile;, endfor;, endforeach;, o endswitch;, respectivamente.

```
<?php if ($a == 5): ?>
A es igual a 5
<?php endif; ?>
```

Gráfico N° 12. Sintaxis de estructuras de control

Fuente: <http://php.net/manual/es/language.operators.php>

##### 1.7.5.4.1 Sentencia while

Ejecuta esta sentencia como TRUE o FALSE, las mismas que se anidan como la expresión while.

- **Sintaxis**

while (expr)

```
{  
  
    Sentencia  
  
}
```

- **Ejemplo**

```
$i = 1;  
while ($i <= 10) {  
    echo $i++; /* el valor presentado sería  
              $i antes del incremento  
              (post-incremento) */  
}
```

Gráfico N° 13. Ejemplo sentencia if evaluada como TRUE

Fuente: <http://php.net/manual/es/language.operators.php>

Si la expresión se ejecuta como FALSE estas no se ejecutarán.

- **Sintaxis**

while (expr):

```
{  
  
    sentencias  
  
    ...  
  
}
```

endwhile;

- **Ejemplo**

```
$i = 1;
while ($i <= 10):
    echo $i;
    $i++;
endwhile;
?>
```

Gráfico N° 14. Ejemplo sentencia if evaluada como FALSE

Fuente: <http://php.net/manual/es/language.operators.php>

#### 1.7.5.4.2 Sentencia do-while

Son similares a los WHILE pero estas sentencias son verificadas al final.

- **Sintaxis**

Do

```
{
    sentencias
    ...
}
```

While (expr);

- **Ejemplo**

```
<?php
$i = 0;
do {
    echo $i;
} while ($i > 0);
?>
```

Gráfico N° 15. Ejemplo sentencia do-while

Fuente: <http://php.net/manual/es/language.operators.php>

### 1.7.5.4.3 Sentencia for

El bucle FOR evalúa la expresión al inicio del mismo si se evalúa como TRUE esta continua su ejecución, si evalúa como FALSE finaliza la ejecución.

- **Sintaxis**

```
for (expr1; expr2; expr3)
```

```
{  
  
    Sentencia  
  
}
```

Cada una de las expresiones puede estar vacía o contener múltiples expresiones separadas por comas.

- **Ejemplo**

```
<?php  
/* ejemplo 1 */  
  
for ($i = 1; $i <= 10; $i++) {  
    echo $i;  
}  
  
/* ejemplo 2 */  
  
for ($i = 1; ; $i++) {  
    if ($i > 10) {  
        break;  
    }  
    echo $i;  
}  
  
/* ejemplo 3 */  
  
$i = 1;  
for (; ; ) {  
    if ($i > 10) {  
        break;  
    }  
    echo $i;  
    $i++;  
}  
  
/* ejemplo 4 */  
  
for ($i = 1, $j = 0; $i <= 10; $j += $i, print $i, $i++);  
?>
```

Gráfico N° 16.Sentencia FOR

Fuente: <http://php.net/manual/es/language.operators.php>

#### 1.7.5.4.4 Sentencia for each

Este constructor proporciona un modo sencillo de interactuar sobre arrays, foreach funciona sólo con arrays y objetos, y emitirá un error al intentar usarlo con una variable de un tipo diferente de datos o una variable no inicializada.

- **Sintaxis**

foreach (expresión\_array as \$valor)

```
{  
  
    sentencias  
  
}
```

foreach (expresión\_array as \$clave => \$valor)

```
{  
  
    sentencias  
  
}
```

- **Ejemplo**

```
<?php  
$array = array(1, 2, 3, 4);  
foreach ($array as &$valor) {  
    $valor = $valor * 2;  
}  
// $array ahora es array(2, 4, 6, 8)  
unset($valor); // rompe la referencia con el último elemento  
?>
```

Gráfico N° 17. Ejemplo sentencia for each

Fuente: <http://php.net/manual/es/language.operators.php>

#### 1.7.5.4.5 Sentencia switch

Esta sentencia es similar a una serie de sentencias IF en la misma expresión. En muchas ocasiones, es posible que se quiera comparar la misma variable, con muchos valores

diferentes, y ejecutar una parte de código distinto dependiendo de qué valor es igual, para esto es exactamente la expresión switch.

- **Sintaxis**

```
switch(expr1)
```

```
{
```

```
    case 0: sentencia 0
```

```
    case 1: sentencia 1
```

```
    ...
```

```
    Case N: sentencia N
```

```
}
```

- **Ejemplo**

```
<?php
if ($i == 0) {
    echo "i es igual a 0";
} elseif ($i == 1) {
    echo "i es igual a 1";
} elseif ($i == 2) {
    echo "i es igual a 2";
}

switch ($i) {
    case 0:
        echo "i es igual a 0";
        break;
    case 1:
        echo "i es igual a 1";
        break;
    case 2:
        echo "i es igual a 2";
        break;
}
?>
```

Gráfico N° 18. Ejemplo sentencia switch

Fuente: <http://php.net/manual/es/language.operators.php>

## 1.8 MYSQL<sup>29</sup>

Es un sistema de gestión de bases de datos relacionales, ya que almacena los datos en tablas separadas las cuales se enlazan mediante relaciones entre ellas. Esto permite que se pueda consultar la información de forma rápida y flexible facilitando el desarrollo.

MySQL es un sistema Open Source esto quiere decir que se puede ser modificado o usado como el desarrollador lo desee o lo necesite. Este sistema tiene varias versiones que se consideran como hijos que se llaman derivados (forks). Algunos de ellos son:

- Drizzle
- MariaDB
- Percona

### 1.8.1 ¿Por qué usar MySQL?<sup>30</sup>

Al ser un sistema FREE el servidor de bases de datos MySQL es muy rápido, seguro, y fácil de usar. Desde el inicio fue desarrollado para el manejo eficiente de grandes bases de datos. Este sistema se encuentra siempre en constante desarrollo, para beneficio de los usuarios que deseen una alternativa rápida y confiable ya que este sistema ofrece hoy en día un conjunto rico y útil de funciones. Y también debemos acotar que al ser un sistema flexible puede adaptarse a varias plataformas como:

- AIX
- BSD
- FreeBSD
- Kurisu OS

---

<sup>29</sup> <https://mysqldaniel.wordpress.com/>

<sup>30</sup> <https://prezi.com/923yydsinkww/concepto-caracteristicas-ventajas-y-desventajas-de-mysql-y-workbench/>

- GNU/Linux
- Mac Osx
- Net BSD
- Open BSD
- OS/2 Warp
- QNX
- SGI IRIX
- Solaris
- Sun Os
- SCO Open Server
- SCO Unix Ware
- Tru 64
- eBD
- Versiones de Win desde la 95 hasta la actualidad
- Open VMS

### **1.8.2 Características<sup>31</sup>**

Al ser MySQL en sus inicios un sistema que carecía de elementos, que se los consideraba esenciales para BD relacionales, como son integridad referencial y transacciones este sistema atrajo a desarrolladores que necesitaban de una aplicación con contenido dinámico.

Gracias a que este lenguaje está en constante cambio, se incorporaron elementos de los que carecía, tanto por sus creadores como desarrolladores de software libre.

---

<sup>31</sup> <https://prezi.com/923yydsinkww/concepto-caracteristicas-ventajas-y-desventajas-de-mysql-y-workbench/>

- Al ser un sistema de desarrollo libre consta de una cantidad de plataformas y sistemas.
- Mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica, transacciones...
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda en indexación de campos de texto.
- Interioridades y portabilidad
- Se le permite hasta 64 bytes por tabla

### **1.8.3 Ventajas y Desventajas<sup>32</sup>**

#### **1.8.3.1 Ventajas**

- Es un sistema de alta velocidad para resolver operaciones.
- Es un gestor de alto rendimiento.
- Es un lenguaje de bajo costo para el desarrollador, gracias a su bajo consumo.
- Puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.

#### **1.8.3.2 Desventajas**

- Un gran porcentaje de las utilidades de MySQL no están documentadas.

---

<sup>32</sup> <https://mysqldaniel.wordpress.com/>

- No es intuitivo, como otros programas (ACCESS).

### 1.8.4 Estructuras de Tabla y de Índice<sup>33</sup>

Los ficheros .frm son los que almacenan la información de su diccionario estos se encuentran en el directorio de Base de Datos, al igual que las tablas InnoDB estas también almacenan su diccionario interno. Al eliminar una tabla o una base de datos, también debe eliminar uno o más ficheros .frm, que corresponden a las entradas dentro del diccionario de datos de InnoDB. Esta es la razón por la cual no se pueden mover tablas entre bases de datos sencillamente moviendo los ficheros .frm.

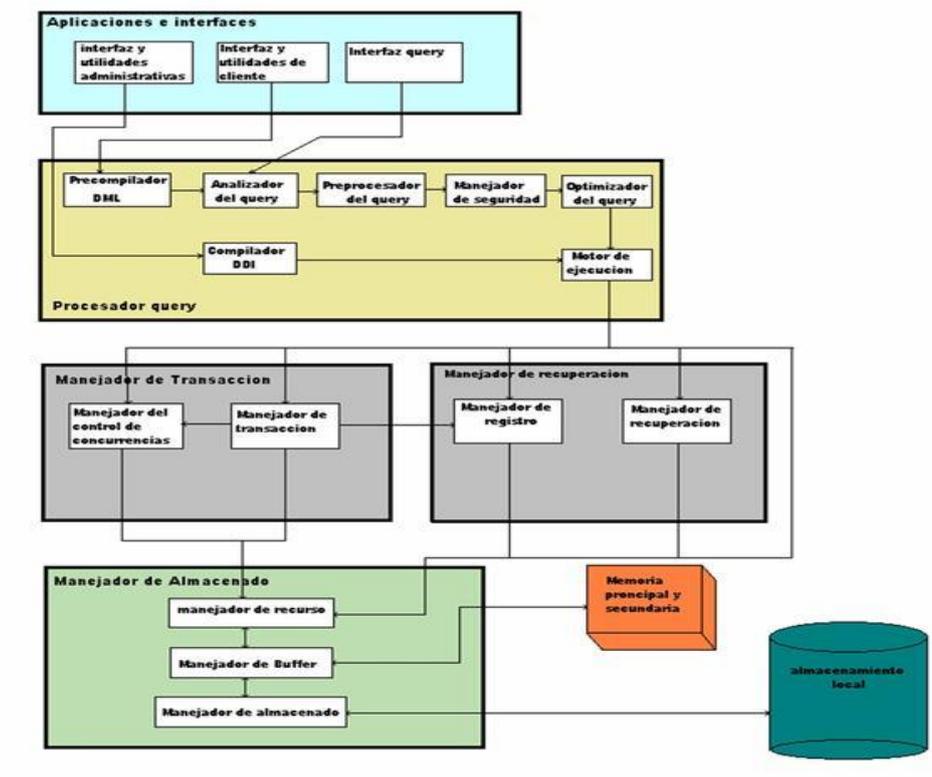


Gráfico N° 19. Estructura de tabla e índice MySQL

Fuente: <http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/table-and-index.html>

<sup>33</sup> <http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/table-and-index.html>

Cada tabla del motor de almacenamiento posee un índice especial denominado clustered index (índice agrupado) el cual registra los datos de las filas. Al definir la Clave Primaria (Primary Key) de una tabla, el índice de esta clave será el índice agrupado.

Al definir una PRIMARY KEY para la tabla, se debe tomar como clave primaria el primer índice UNIQUE que tenga solamente columnas NOT NULL, al cual InnoDB utiliza como índice agrupado. Si no hay en la tabla un índice con esas características, InnoDB generará internamente un índice agrupado donde las filas estarán ordenadas por el identificador de fila (row ID) que InnoDB asigna a las columnas en tal tabla. El identificador de fila es un campo de 6 bytes que se incrementa automáticamente a medida que se agregan nuevas filas. Por lo tanto, las filas ordenadas por este identificador están en el orden físico de inserción.

Debido a que la fila de datos y el índice de búsqueda se encuentran en la misma página el acceso a la información almacenada es bastante rápido. Esta arquitectura de índice agrupado ahorra operaciones de E/S del disco, en tablas grandes comparado con otros servidores de bases de datos, los cuales separan en páginas diferentes estos elementos.

InnoDB almacena el valor de la clave primaria de la fila en índices no agrupados o también llamados secundarios, y los utiliza para realizar la búsqueda de la fila a partir del índice agrupado. Por lo tanto, mientras más larga sea la clave primaria, los índices secundarios utilizarán más espacio.

InnoDB realiza la comparación de cadenas de tipo CHAR y VARCHAR, que tengan distinta longitud, suponiendo que después del último carácter de la cadena más corta existen espacios en blanco hasta completar la longitud de la cadena más larga.

#### **1.8.4.1 Estructura física de un índice<sup>34</sup>**

En InnoDB se representa a los índices como arboles binarios (B-trees) que almacenan en las hojas del árbol las páginas que contienen las entradas de índice, estas páginas tienen por defecto un tamaño de 16 Kb y cuando se realiza la inserción de nuevas entradas se reserva 1/16 de la página para nuevas inserciones o actualizaciones del índice.

#### **1.8.4.2 Búfer de inserciones**

En las aplicaciones que utilizan bases de datos es frecuente que las claves primarias sean identificadores únicos y las filas se insertan en orden ascendente en relación a esa clave, por lo que para realizar estas inserciones en el índice agrupado no se necesitan lecturas aleatorias del disco. En cambio, las inserciones de índices secundarios por lo general se realizan de forma aleatoria, ya que no son únicos. Esto generalmente provoca una gran cantidad de operaciones de E/S en el disco.

Para solucionar este problema, InnoDB implementa un pool de buffer en el cual se almacena una estructura especial de inserciones, cada vez que se va a realizar la inserción de un índice secundario no único, InnoDB verifica si este se encuentra en el pool de buffer, si es así, se realiza la inserción directamente en la página de índice. De lo contrario se insertará una entrada en el pool de buffer. Así las inserciones se realizan rápidamente.

Este procedimiento puede acelerar hasta 15 veces la velocidad de inserción de datos en una tabla, integrando en la misma página del árbol de índices varias inserciones lo que reduce considerablemente las operaciones de E/S del disco.

---

<sup>34</sup> <http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/table-and-index.html>

### 1.8.4.3 Índices hash adaptables<sup>35</sup>

Si una tabla cabe casi completamente en la memoria principal, la manera más rápida de ejecutar consultas sobre ella es empleando índices hash. InnoDB posee un mecanismo automático que supervisa las búsquedas realizadas sobre los índices de una tabla. Si InnoDB advierte que las consultas se podrían beneficiar con la creación de un índice hash, lo hará automáticamente.

El índice hash siempre está basado en un índice B-tree existente en la tabla. InnoDB puede generar un índice hash sobre un prefijo de la clave definida para el B-tree de cualquier longitud, dependiendo del patrón de búsquedas que InnoDB observe en el índice B-tree. Un índice hash puede ser parcial: no se necesita que el índice B-tree completo sea alojado en el pool de buffer. InnoDB genera índices hash según sea necesario basándose en las páginas de índice que son utilizadas más frecuentemente.

Podría decirse que, a través del mecanismo de índices hash adaptativos, InnoDB se adapta a la memoria principal, acercándose así a la arquitectura de las bases de datos de memoria.

### 1.8.5 Estructura física de los registros

Los registros de las tablas InnoDB tienen las siguientes características:

- Cada registro de índice en InnoDB contiene un encabezado de seis bytes. El encabezado se emplea para enlazar juntos registros consecutivos, y también en el bloqueo a nivel de fila.
- Los registros en el índice agrupado contienen campos para todas las columnas definidas por el usuario. Adicionalmente, hay un campo de seis bytes para el IDentificador de transacción y un campo de siete bytes para el roll pointer.

---

<sup>35</sup> <http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/table-and-index.html>

- Si no se definió una clave primaria para la tabla, cada registro de índice agrupado contiene también un campo de identificación de fila de seis bytes.
- Cada registro de índice secundario contiene también todos los campos definidos para la clave del índice agrupado.
- Un registro contiene además un puntero a cada campo del mismo. Si la longitud total de los campos en un registro es menos de 128 bytes, el puntero medirá un byte, de lo contrario, tendrá dos bytes de longitud. La matriz de estos punteros se conoce como el directorio de registros. El área a donde señalan estos punteros se denomina la parte de datos del registro.
- Internamente, InnoDB almacena las columnas de caracteres de longitud fija (como CHAR (10)) en un formato de longitud fija. InnoDB trunca los espacios sobrantes de las columnas VARCHAR. Nótese que MySQL puede convertir internamente columnas CHAR a VARCHAR.
- Un valor NULL SQL reserva 1 o 2 bytes en el directorio de registros. No reservará ningún byte en la parte de datos del registro si se lo almacena en una columna de longitud variable. En una columna de longitud fija, reservará en la parte de datos la longitud asignada a dicha columna. La razón por la que se reserva este espacio fijo a pesar de tratarse de un valor NULL, es que en el futuro se podrá insertar en su lugar un valor no-NULL sin provocar la fragmentación de la página de índice

### **1.8.6 Procedimientos almacenados y funciones<sup>36</sup>**

Conjunto de instrucciones que controla la lógica y contiene cierta lógica, que se encuentran reposando en el servidor.

A continuación se detallan algunas directivas de MYSQL:

---

<sup>36</sup> <https://tecnologiaenvivo.com/mysql-procedimientos-almacenados-y-funciones/>

### 1.8.6.1 Create Procedure

Relaciona y está ligado a la BD igual se crea una tabla, a este procedimiento se lo denomina reservada CALL

- **Sintaxis**

**CREATE PROCEDURE** nombre (parámetro) [características] definición

- **Parámetros**

Separados por comas, si no es así, este debe estar seguida por paréntesis.

Los parámetros tienen la siguiente estructura:

- **modo:** es opcional y puede ser IN (el valor por defecto, son los parámetros que el procedimiento recibirá), OUT (son los parámetros que el procedimiento podrá modificar) INOUT (mezcla de los dos anteriores).
- **nombre:** es el nombre del parámetro.
- **tipo:** es cualquier tipo de dato de los provistos por MySQL.
- **definición:** es el cuerpo del procedimiento y está compuesto por el procedimiento en sí: aquí se define qué hace, cómo lo hace y bajo qué circunstancias lo hace.

### 1.8.6.2 Create Function

La sintaxis RETURNS es obligatoria cuando se define una función la misma que especifica qué tipo de dato será devuelto por la función.

- **Por ejemplo**

MAX() o COUNT().

- **Sintaxis**

**CREATE FUNCTION** nombre (parámetro)

**RETURNS** tipo

[características] definición

- **Parámetros**

Los parámetros contienen varios o ninguna “;”, ya sea que exista o no parámetros y su estructura es:

Los parámetros tienen la siguiente estructura:

- **nombre:** es el nombre del parámetro.
- **tipo:** es cualquier tipo de dato de los provistos por MySQL.
- **procedimiento en sí:** aquí se define qué hace, cómo lo hace y cuándo lo hace (UNEWEB, 2016).

## 1.9 MVC (MODELO VISTA CONTROLADOR)<sup>37</sup>

Es un controlador que permite separar el código en diversas responsabilidades, mediante un patrón de programación

MVC se utiliza donde se requiere el uso de interfaces, con una propuesta de diseño, por lo cual, surgió de acuerdo a la necesidad de un software más robusto con un ciclo de vida óptimo.

---

<sup>37</sup> <https://desarrolloweb.com/articulos/que-es-mvc.html>

## 1.9.1 CAPAS DEL MVC<sup>38</sup>

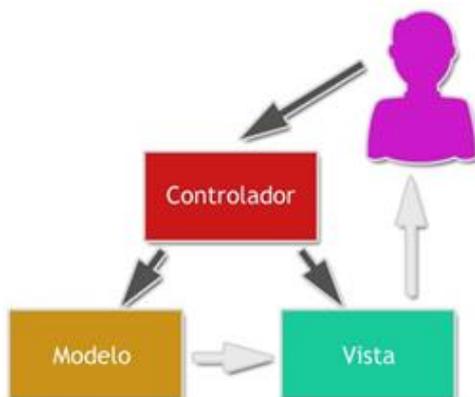


Gráfico N° 20. Capas del Modelo Vista Controlador

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

### 1.9.1.1 Modelos

En la primera capa se utiliza librerías como PDO, ORM, Doctrine las cuales permiten al desarrollador trabajar con bases de datos y objetos abstractos. Esto permite que no utilice sentencias SQL directas.

### 1.9.1.2 Vistas

Las vistas contienen el código de nuestra aplicación que va a producir la visualización de las interfaces de usuario, o sea, el código que nos permitirá en una jerga informática referirse a una imagen o vídeo, los estados de nuestra aplicación en HTML. En esta capa generalmente trabajamos con datos, sin embargo, no se realiza un acceso directo a éstos.

### 1.9.1.3 Controladores

Permite realizar o crear el código indispensable para solicitar a la aplicación, que visualiza un elemento, como por ejemplo una compra, o buscar información específica.

Aquí el enlace entre vistas y modelos, permite que los mecanismos que se requieren sean las más óptimas y se adapten a las necesidades de la aplicación

---

<sup>38</sup> <https://desarrolloweb.com/articulos/que-es-mvc.html>

## 1.9.2 Características

- MVC, convierte la aplicación en un módulo fácil, rápido en el desarrollo con sus patrones de diseño.
- Al separar tareas, en vistas, y controladores se convierte en una aplicación más liviana.
- Agregar código nuevo a un desarrollo viejo es más sencillo de implementar.
- Los prototipos, ayudan a los programadores trabajar simultáneamente.
- La separación también permite a los desarrolladores hacer cambios en una parte de la aplicación sin afectar a los demás.

## 2.4.3 Funcionamiento del MVC<sup>39</sup>

El funcionamiento básico del patrón MVC, puede resumirse en:

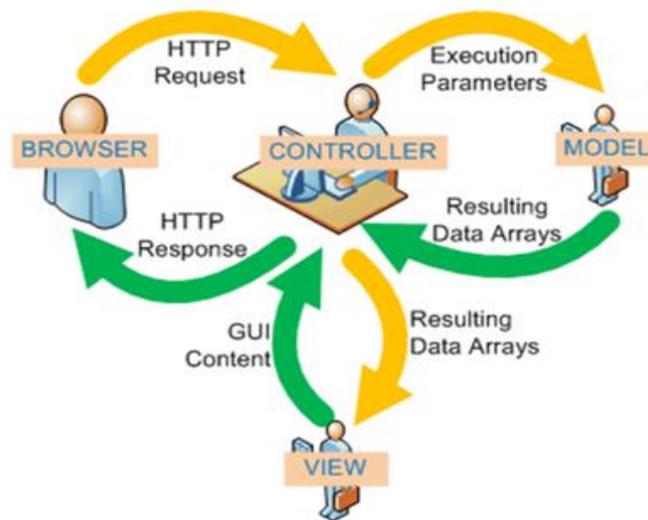


Gráfico N° 21. Funcionamiento del MVC

Fuente: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

- El usuario realiza una petición.
- El controlador captura la petición.
- Hace la llamada al modelo correspondiente.

<sup>39</sup> <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

- El modelo será el encargado de interactuar con la base de datos.
- El controlador recibe la información y la envía a la vista.
- La vista muestra la información.
- Y el controlador reenvía la información al cliente.

## **1.10 HERRAMIENTAS DE DESARROLLO<sup>40</sup>**

En el desarrollo de aplicaciones web es uno de las ramas más importantes, este desarrollo se compone de 5 etapas, las cuales permite realizar el desarrollo más óptimo, las mismas que cuentan con sus respectivas herramientas de desarrollo.

Las etapas de desarrollo del software más comunes, son las siguientes:

1. Modelado
2. Desarrollo
3. Pruebas
4. Depurado
5. Diseño

Cada una de las etapas llevadas a cabo en el desarrollo de software, cuenta con sus distintas herramientas, que tienen como finalidad disminuir el esfuerzo del programador y el tiempo requerido para realizar cada fase, además mejorar los resultados obtenidos.

### **1.10.1 Herramientas para Modelado de Software**

La UML, es una de las herramientas más comunes, que utilizan los desarrolladores en la creación de modelos de software, esta representación gráfica de los diversos elementos dentro de los sistemas a desarrollar, ayuda a los programadores y al cliente a visualizar

---

<sup>40</sup> <https://okhosting.com/blog/herramientas-de-desarrollo-de-software/>

mejor mediante un diseño abstracto, la definición del sistema, su funcionamiento y sus funcionalidades.

## ... Diagramas de UML

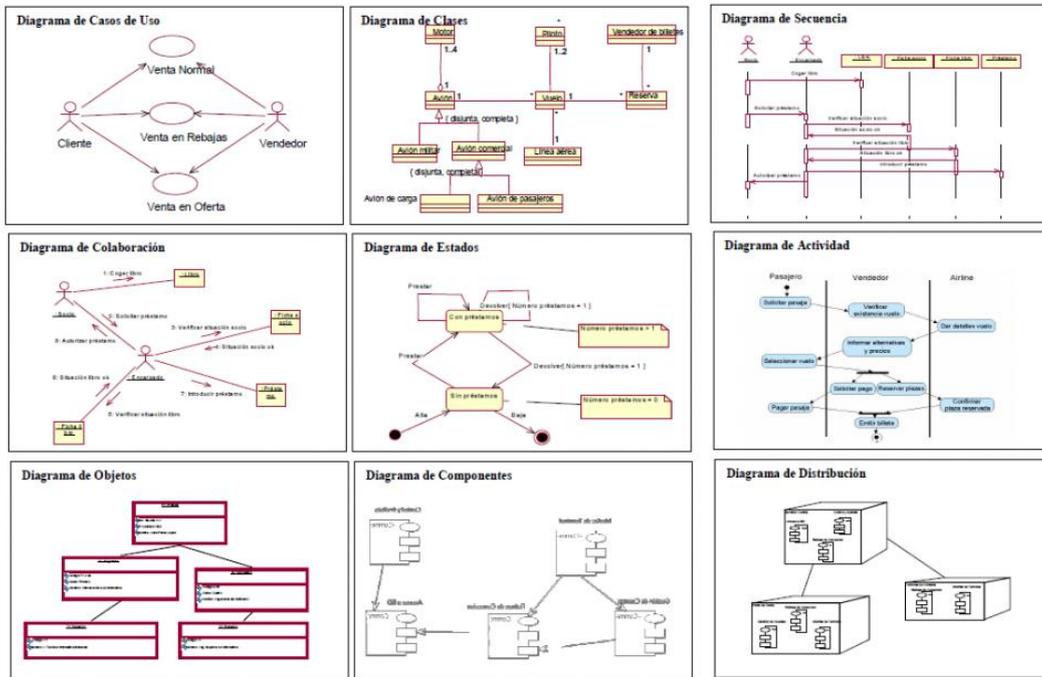


Gráfico N° 22. Ejemplo de diagramas UML

Fuente: <https://okhosting.com/blog/herramientas-de-desarrollo-de-software/>

### 1.10.2 Herramientas de Desarrollo de Software

Depende mucho de lenguaje de programación que se desee utilizar, ya que son las aplicaciones o programas para el desarrollo del sistema que se ha definido en la UML. Estas herramientas son de importancia vital o secundaria como las IDE ya que aquí se ensambla, se compilan las aplicaciones.

Los IDE permiten contar con un ambiente amigable, para escribir el código y analizarlo, mostrando el lugar donde la sintaxis está incorrecta. Otra de las funciones es la de compilar el código escrito para consiguientemente realizar pruebas y hacer las respectivas depuraciones. Con todos los elementos visuales y amigables permitirán el mejoramiento en cuanto a tiempo de desarrollo y la calidad del producto final.

### **1.10.2.1 Herramientas para Hacer Pruebas del Desarrollo de SW**

Una de las etapas fundamentales en el desarrollo del software, son las pruebas. Se requiere de pruebas de código, que verifique el correcto desarrollo de la aplicación y a su vez al finalizar el mismo, trabaje correctamente. Para esto, los desarrolladores deben contar con un software que ejecute el código en modo prueba para desarrollar un ambiente productivo, mediante el uso herramientas que especifiquen un conjunto de entradas de prueba para una aplicación, al finalizar las mismas, se debe recibir en la información de salida detallada de los resultados.

Al contar con herramientas IDE estas ya vienen con una herramienta de pruebas.

### **1.10.2.2 Herramientas para Depurar Código**

En la actualidad la depuración de código se lo realiza de forma intuitiva por tratarse de un proceso que no se lo considera prioridad al momento de desarrollar un programa en un lenguaje de bajo nivel. Como objetivo principal en este tipo de herramientas es obtener la habilidad de encontrar errores de código para solucionarlos.

Esta herramienta se vuelve compleja cuando la aplicación es extensa, entonces se encontrarán un mayor número de errores y problemas potenciales.

Gracias a la habilidad de este tipo de herramientas de cambiar los valores de las variables en tiempo real, quiere decir que mientras se está ejecutando el programa, los cambios solamente serán visuales al momento que esté activa la herramienta. Por lo tanto, no afectarán al código original, permitiendo que se realice las pruebas necesarias sin limitaciones. Pues no se pone en riesgo el código al modificar valores. Es por eso que se convierte en una herramienta indispensable para los programadores.

### **1.10.2.3 Herramientas para el Diseño de Software**

Son herramientas que permiten en las primeras etapas el desarrollo de la interfaz, contando con IDE que ayudan al diseño del proyecto.

Permite el diseño de ventanas, botones, con sus funcionalidades, ayudando a los programadores, la visualización de la aplicación.

Con este tipo de herramientas se crea diseños llamativos, y novedosos.

### **1.11 NETBEANS<sup>41</sup>**

El entorno de desarrollo integrado Netbeans, fue creado en principio para trabajar con Java aunque luego fue extendido, mediante el uso de módulos, para trabajar con otros lenguajes de programación. Es un entorno de desarrollo libre que se encuentra disponible sin costo para su descarga y uso.

Es bastante versátil, ya que permite desarrollar varios tipos de aplicaciones: de escritorio, sitios web o para dispositivos móviles. Además, puede ser instalado en diversos sistemas operativos. Tiene un diseño modular donde cada módulo provee una funcionalidad específica. NetBeans una vez descargado e instalado se encuentra listo para empezar a desarrollar las aplicaciones. Los módulos están conformados por dos archivos, el primero contiene las clases de java que interactúan con las interfaces de programación (API's) y, el segundo archivo se denomina "manifest" que lo identifica como módulo.

De este modo, las aplicaciones desarrolladas pueden ser extendidas fácilmente por otros desarrolladores agregándole nuevas funcionalidades, esto gracias a que cada módulo puede ser implementado de forma independiente.

---

<sup>41</sup> <https://appmundomovil.wordpress.com/2-3-entorno-de-desarrollo/>

Netbeans permite optimizar el código generado y, de esta forma, reduce el uso de recursos permitiendo a las aplicaciones ejecutarse más rápido con un uso mínimo de memoria. Además, monitorea el comportamiento de la aplicación presentando información de los recursos utilizados, así como realizar comparación de los mismos en diferentes etapas de la ejecución.

También accede a distintos gestores de bases de datos directamente, de esta forma se puede realizar cambios en las tablas sin abandonar el entorno de desarrollo.

### **1.11.1 Plataforma Netbeans<sup>42</sup>**

Es un entorno de trabajo que facilita el desarrollo de aplicaciones, contiene las herramientas necesarias para empezar a implementar plugins y aplicaciones basadas en la plataforma NetBeans. Se puede instalar módulos de forma dinámica directamente desde las aplicaciones, lo cual permite incluir, por ejemplo, un módulo de actualización que permitirá a los usuarios añadir nuevas características a la aplicación en tiempo de ejecución, sin tener que descargar toda la aplicación nuevamente. Se puede utilizar servicios reusables comunes en las aplicaciones de escritorio, lo que permite al desarrollador enfocarse en la lógica de la aplicación.

### **1.11.2 Características<sup>43</sup>**

Entre sus características se describen las siguientes:

- Soporte JavaScript.
  - Sintaxis Resaltada
  - Completación de Código y Análisis de Típo
  - Soluciones Rápidas (Quick Fixes) y Verificación de Sintaxis

---

<sup>42</sup> <http://laprogramacionentuidiaria.blogspot.com/2015/11/netbeans.html>

<sup>43</sup> [https://netbeans.org/community/releases/61/index\\_es.html](https://netbeans.org/community/releases/61/index_es.html)

- Refactorización
- Menor consumo de memoria
- Soporte de Estructuras Spring
  - Agregado de la librería Spring Framework 2.5
  - Asistentes para configuración de archivos XML y controladores Spring Web MVC
  - Completación de Código de nombres bean y clases y propiedades Java
  - Soporte de entorno Spring Web MVC en proyectos web
  - Completación de Código Spring,
- Nuevo Soporte MySQL en Exploración de Bases de Datos
  - Registro de servidores MySQL
  - Ver, crear y borrar bases de datos
  - Fácil lanzamiento de la herramienta de administración para MySQL
  - Nodo MySQL
- Soporte Java Beans
  - Modelos Bean en el Navegador
  - Generador de Propiedades Bean
  - Editor BeanInfo
  - Navegador Java Bean
- Generador JSF CRUD
  - Generador de aplicaciones JavaServer Faces CRUD a partir de clases de entidades.
  - Soporta todo tipo de relaciones de entidades.
  - Soporta todo tipo de claves principales.
- Soporte Ruby/JRuby

- Mejoras en el editor, incluyendo nuevas sugerencias y soluciones
- Soporte de depuración rápida en JRuby
- Administrador de Plataforma
- Mejoras en la integración de servidores y bases de datos en proyectos Rails
- Editor Ruby
- Completación de Código Javadoc
  - Soporte de etiquetas (tags) estandares: @param, etc.
  - Completación de Código para parámetros, excepciones, etc.
- Soporte para los Web APIs Más Usados
  - Fácil creación de aplicaciones remezcladas (mashup)
  - Operaciones de Arrastrar y soltar dentro del entorno POJO, Servlet, JSP y servicios web
  - Soporte de web APIs
  - Compartir Proyectos (Librerías Compartidas AKA)
- Nuevas Extensiones (Plugins)
  - Control de versión ClearCase
  - Soporte AXIS
  - Soporte SOAP UI
  - Soporte Hibernate Framework
  - Java Mobility (Aplicaciones para Móviles)
  - Emulador Mpowerplayer MIDP para aplicaciones MIDP en MacOS X
  - Estructurador SVG (SVG Composer) para Componentes SVG de Uso Frecuente
- Soporte de Edición para PHP:
  - Completación de código, sintaxis resaltada, navegación, depuración
  - Depurador JavaScript

- Extensión JavaFX

### **1.11.3 Ventajas y Desventajas de Netbeans<sup>44</sup>**

#### **Ventajas**

- Multiplataforma
- Multilenguaje
- Código abierto y gratuito
- Apoyo de la comunidad con el soporte
- Recursos como documentación, video, tutoriales, traductores de plugins
- Módulos
- Fácil de usar y adaptable
- Variedad de Plugins, que son herramientas para poder agregar algún componente o funciones extras.
- NetBeans soporta el desarrollo de todos los tipos de aplicación Java.

#### **Desventajas**

- Lento si no se cuenta con un buen computador y a su vez una memoria de gran capacidad.
- Entre más proyectos se coloca es más lento porque lee todo el código.

### **1.11.4 Características de NetBeans integrado a PHP**

Las características más importantes de NETBEANS integrado a PHP son:

---

<sup>44</sup> <https://programacionmovilandroid.wordpress.com/netbeans/>

#### **1.11.4.1 Creación de Proyectos PHP con NetBeans**

NetBeans provee de una estructura para los proyectos que se crean junto a este IDE, nos propone un esqueleto para organizar nuestro código fuente, el editor conjuntamente integra los lenguajes como HTML, JavaScript y CSS.

Además, NetBeans posee un sistema para examinar todos los directorios de cada proyecto, haciendo reconocimiento y carga de clases, métodos y objetos, para acelerar la programación.

#### **1.11.4.2 Integración con Symfony y ZenFramework**

Una característica de NetBeans es la integración de Framework de PHP, de hecho, realizar aplicaciones con estas herramientas es muy ágil.

Con NetBeans ya es posible dejar de lado la consola de comandos de Symfony y centrarse en desarrollar en el IDE, además se encuentra cargadas todas las clases, ayuda en línea, etc. Un espectáculo si se habla de desarrollar en NetBeans y la completa integración con Symfony y ZenFramework.

#### **1.11.4.3 Editor de Código Fuente**

A lo largo de todo este tiempo, se notó la mejora en su editor, sobre todo de PHP, es mucho más ágil y a la vez robusto, contiene más ayuda en línea, reconocimiento de sintaxis y todo lo que provee la última versión de PHP, la 5.3

#### **1.11.4.4 Integración con PHP Unit Testing**

Es posible crear test con PHPUnit, para diferentes funciones, luego realizar la comprobación y ver todos los resultados. En las propiedades PHPUnit puede definir una configuración personalizada de archivos XML, un archivo de arranque para las opciones de línea de comandos, o una serie de pruebas a medida, o puede que el IDE genera el código esqueleto para usted.

#### **1.11.4.5 Depuración de PHP**

NetBeans integra muy bien la utilización Xdebug, gracias a esto podemos inspeccionar y examinar cada variable local, establecer puntos de interrupción y evaluar el código en nuestra lógica.

El IDE de NetBeans para PHP también ofrece la línea de comandos de depuración: La salida del programa PHP aparece en una pantalla de línea de comandos en el IDE de sí mismo y se puede inspeccionar el código HTML generado sin tener que cambiar a un navegador.

#### **1.11.4.6 Integración con MySQL**

En proyectos por lo general cuando se utiliza un servidor de Bases de Datos MySQL y NetBeans posee una integración completa en términos de administración básico y avanzada de MySQL, y todo desde el mismo entorno.

#### **1.11.4.7 Integración con Sistemas de Control de Versiones**

Esta es una de las condiciones necesarias para los proyectos y es la posibilidad de contar con la integración de sistemas de control de versiones, tales como SVN, CVS, Mercurial y Git.

Desde el editor es posible realizar la administración de estos sistemas versionados, sus commit, branch, importar, exportar, revert, clonar, etc. Por todo esto y más herramientas que NetBeans 6.9.1 me brinda, definitivamente decidí utilizar este maravilloso IDE a la hora de realizar mis proyectos.

## **1.12 WAMPSEVER<sup>45</sup>**

Permite la creación de aplicaciones WEB con Apache, PHP y también en conjunto con bases de datos MySQL database. A pesar de que su entorno de desarrollo para Windows. Este sistema incluye PHPMyAdmin y SQLiteManager para manejar sus bases de datos.

### **1.12.1 Características**

Este sistema provee a los programadores de cuatro elementos necesarios para la creación de una aplicación:

- Un Sistema Operativo (Windows),
- Un manejador de base de datos (MySQL),
- Un software para servidor web (Apache)
- Un software de programación script Web (PHP (generalmente), Python o PERL),  
debiendo su nombre a dichas herramientas.

### **1.12.2 Herramientas para el diseño y desarrollo de páginas web**

En WAMPSEVER se encuentran una variedad de herramientas para mayor facilidad de desarrollo

#### **• Del lado del servidor**

Windows, como sistema operativo

- Apache, como servidor web;
- MySQL, como gestor de bases de datos;
- PHP (generalmente), Perl, o Python, como lenguajes de programación.

---

<sup>45</sup> [https://prezi.com/h0j-\\_95vhkj2/que-es-wampserver/](https://prezi.com/h0j-_95vhkj2/que-es-wampserver/)

- **Del lado del Cliente**

- Navegador de Internet

### **1.12.3 Funcionalidades de WampServer**

WampServer tiene funcionalidades que lo hacen muy completo y fácil de usar.

Con un click izquierdo sobre el icono de WampServer, será capaz de:

- Gestionar sus servicios de Apache y MySQL
- Cambiar de línea / fuera de línea (dar acceso a todos o sólo local)
- Instalar y cambiar de Apache, MySQL y PHP emisiones
- Gestión de la configuración de sus servidores
- Acceder a sus registros
- Acceder a sus archivos de configuración
- Crear alias

### **1.12.4 Utilidad**

Al momento de desarrollar aplicaciones web este sistema es muy importante, esto es gracias a que podemos trabajar en el servidor web, y ejecutar aplicaciones y visualizar como serian una vez que se encuentren subidas al hosting o servidor web.

Al igual que gestionar los datos esto es gracias al motor de base de datos de MySQL y su administrador PHPMYAdmin.

### 1.13 BOOTSTRAP<sup>46</sup>

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo. Es Open Source, por lo que lo se puede usar de forma gratuita y sin restricciones.

El beneficio de usar responsive design en un sitio web, es principalmente, que este se adapte automáticamente al dispositivo desde donde se acceda. El Framework Bootstrap permite que el contenido de una página web pueda adecuarse a condiciones como la resolución de la pantalla. Si se configura las dimensiones del contenido en porcentajes, se puede tener una web muy fluida capaz de ajustarse a casi cualquier tamaño de forma automática.

Aun ofreciendo todas las posibilidades que ofrece Bootstrap a la hora de crear interfaces web, los diseños son simples, limpios e intuitivos, esto brinda agilidad a la hora de cargar y al adaptarse a otros dispositivos. El Framework trae varios elementos con estilos predefinidos fáciles de configurar: Botones, Menús desplegables, Formularios incluyendo todos sus elementos e integración jQuery para ofrecer ventanas y tooltips dinámicos.

Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript, Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS 3, pero es compatible con la mayoría de los navegadores web. La información básica de

---

<sup>46</sup> <https://www.arweb.com/chucherias/%C2%BFque-es-bootstrap-y-como-funciona-en-el-diseno-web/>

compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores. Por ejemplo, las propiedades introducidas en CSS3 para las esquinas redondeadas, gradientes y sombras son usadas por Bootstrap a pesar de la falta de soporte de navegadores antiguos. Esto extiende la funcionalidad de la herramienta, pero no es requerida para su uso.

Desde la versión 2.0 también soporta diseños sensibles. Esto significa que el diseño gráfico de la página se ajusta dinámicamente, tomando en cuenta las características del dispositivo usado (Computadoras, tabletas o teléfonos móviles).

### **1.13.1 Ventajas de usar Bootstrap**

- Se puede tener una web bien organizada de forma visual rápidamente: la curva de aprendizaje hace que su manejo sea asequible y rápido.
- Permite utilizar muchos elementos web: desde iconos a menús desplegables, combinando HTML5, CSS y Javascript.
- El diseño será adaptable, no importa el dispositivo, la escala o resolución.
- Permite maquetar por columnas de una forma fácil y rápida. Además, son muy configurables.
- Se integra muy bien con las principales librerías Javascript.
- Al haber sido creado por Twitter, da ciertas garantías: está muy pensado y hay mucho trabajo ya hecho. Por lo tanto, hay una comunidad muy activa creando, arreglando cosas, ofreciendo plugins y mucho más.
- Cuenta con implementaciones externas para WordPress, Drupal, etc.

### **1.13.2 Desventajas de usar Bootstrap**

- Es necesario adaptarse a su forma de trabajo, su estructura y nomenclatura.
- Se debe adaptar el diseño a un grid de 12 columnas, que se modifican según el dispositivo. Aquí empiezan los problemas ya que Bootstrap por defecto trae anchos, márgenes y altos de línea.
- Es complicado, cambiar de versión si se ha realizado modificaciones profundas sobre el Core.
- Si es necesario añadir componentes que no existen, debe hacerlo manualmente en CSS y cuidar de que mantenga coherencia con el diseño y cuidando el responsive.
- Este framework no es ligero, y además, para algunas funcionalidades, será necesario tener que usar JavaScript y jQuery.

## **CAPÍTULO III**

### **ANÁLISIS Y DISEÑO DEL SISTEMA**

## **1.14 RECOPIACIÓN DE INFORMACIÓN**

La recopilación de la información, basa su estructura en el levantamiento de datos acerca de la necesidad que posee la Caballería Blindada N°11 “Galápagos”. Además, la definición de las características y los aspectos funcionales que contendrá el desarrollo del sitio web para el registro de las confrontas y el control del servicio de alimentación.

El sistema Web debe ser capaz de compensar todas las necesidades analizadas permitiendo al usuario tener un sistema de gestión, manejo y uso ágil de esta herramienta logrando la optimización de todas las actividades realizadas en la mencionada entidad.

Dentro de las funciones que cumple el sistema web se mencionan las siguientes:

- Administrar adecuadamente la información de acuerdo a cada nivel de usuario o perfiles creados promoviendo siempre la integridad de los datos.
- Determinar un acertado registro, procesamiento y almacenamiento de la información que se manejará.
- Delimitar todas las actividades a detalle, referentes al control del servicio de alimentación dentro de la Caballería Blindada N°11 “Galápagos”.

### **1.14.1 Alcance**

El sistema web está enfocado a otorgar un beneficio directo y exclusivo a todos los integrantes responsables del manejo de confrontas en la Caballería Blindada N°11 “Galápagos”, que demanden la adquisición de alimentos, el número de comidas al día por Oficiales, Voluntarios y Conscriptos, y el cobro por dicho servicio.

El presente documento detalla la funcionalidad que posee el desarrollo de un sistema web y como este, pretende resolver la problemática existente en los usuarios inmersos

en el proceso de los servicios de alimentación en el establecimiento. La funcionalidad se detalla a continuación en la temática “Casos de uso”.

## **1.15 ANÁLISIS**

Dentro de los factores y requerimientos iniciales analizados, previo al desarrollo del sistema web se mencionan ciertos puntos desarrollados:

- El sistema web establece una comunicación Cliente –Servidor, que utilizará el Internet para su acceso requiriendo una cuenta de usuario y un password para cada usuario.
- El almacenamiento de la información promueve su integridad mediante el uso de una base de datos.
- Mediante la gestión de los usuarios se establece un control de todas las confrontas diarias que ingresan en el sistema.
- Una vez ingresados y almacenados los datos, se corroborará toda la información mediante la generación de reportes.

### **1.15.1 Estudio de Factibilidad**

De acuerdo a la recopilación y el análisis de la información se realizó un estudio de factibilidad que delimitó la viabilidad operativa, técnica, legal y económica del proyecto según las características que a continuación se especifican:

#### **1.15.1.1 Operativa**

El sistema web será utilizado por tres tipos de elementos, siendo el primero el administrador representado por el técnico desarrollador quien verificará el buen uso de esta herramienta. Así también se contará con un usuario amanuense de cada unidad encargado del registro de las comandas, la generación de reportes y el manejo global del

sistema. El último componente será el encargado de controlar la información presentada por el amanuense denominado Ranchero.

Mediante la ayuda y el soporte de personal profesional especializado dentro del área de sistemas de la Caballería Blindada N°11 “Galápagos” se brindó la respectiva capacitación acerca del uso adecuado del sistema web verificando que el personal cuenta con conocimientos básicos en informático lo que facilitó el proceso de enseñanza-aprendizaje.

### 1.15.1.2 Técnica

#### Recurso humano

ACTIVIDAD	RESPONSABLE
Desarrollo	Diego A. Macas V.
Análisis y diseño	Diego A. Macas V.
Tester	Diego A. Macas V.
Documentador	Diego A. Macas V.
Tutor	Ing. Andrea Moncayo

Tabla N° 1. Recurso humano

Fuente: Autor

#### Recurso técnico

- **Hardware (Laptop Toshiba Satellite)**

DESCRIPCIÓN	
MAINBOARD	INTEL
PROCESADOR	INTEL ® CORE ™ I3 4005U
DISCO DURO	750 GB
MONITOR	15.6 PULGADAS

Tabla N° 2. Recurso técnico hardware

Fuente: Autor

- **Software (Laptop Toshiba Satellite)**

<b>DESCRIPCIÓN</b>	
SISTEMA OPERATIVO	WINDOWS 8.1
ENTORNO DE DESARROLLO WEB	WAMPSEVER
ID	MACROMEDIA NETBEANS 8.2
GESTRO DE BASE DE DATOS	MYSQL
DOCUMENTACIÓN	MICROSOFT OFFICE 2013

Tabla N° 3. Recurso técnico software

Fuente: Autor

### **1.15.1.3 Económica**

El desarrollo del sistema web no tuvo grandes inversiones porque se utilizaron herramienta con licencias gratuitas o conocidas como software libre. Sin embargo, se recurrieron en varios gastos como el hosting y el dominio que fueron un aporte del proponente de la investigación.

### **1.15.1.4 Legal**

El sitio web se encuentra dentro de un establecimiento que requiere de ciertos factores en cuanto a la restricción y confiabilidad de su información, por tanto, se siguieron ciertos parámetros establecidos a continuación:

- Directiva N° 04 del 2010 para la administración de la asignación del rancho del personal militar de las Fuerzas Armadas.
- Instructivo N° FT-DDIE-2015-004-O para la administración del servicio de alimentación en las unidades militares de la fuerza terrestre, publicada en Orden General N° 03 de fecha 06-ENE-2016.
- Los derechos de autoría serán únicos y reservados por parte del programador y diseñador del presente sistema.

### 1.15.2 Análisis de los requerimientos

El análisis de los requerimientos es un paso importante dentro de la Ingeniería del software, constituye el primer paso para definir la funcionalidad del sistema y las tareas que se desean para cubrir las necesidades de la entidad.

El análisis de los requerimientos está compuesto por aquellos que tiene una funcionalidad o determinan una tarea en el desarrollo del sistema web, y los no funcionales que establecen una serie de requisitos que necesita la herramienta para el cumplimiento de estándares de calidad en el diseño.

La determinación de los requerimientos representa un factor importante previo a la fase de diseño porque son los encargados de recibir todas las necesidades que posee el beneficiario, en este caso la Caballería Blindada N°11 “Galápagos”.

#### 1.15.2.1 Requerimientos funcionales

Los requerimientos funcionales representan aquellas acciones que el sistema web requiere para interactuar con su ambiente y como esta incide sobre dicha aplicación.

A continuación, se define mediante un modelo estándar el detalle de cada uno de los requerimientos funcionales.

<b>Nombre</b>
Presentar el Portal Web Informativo
<b>Descripción</b>
La aplicación web multimedia de forma inicial presenta el portal web informativo del sistema de gestión de confrontas y control del servicio de alimentación para la Caballería Blindada N°11 “Galápagos”.
<b>Prioridad</b>
Alta

Tabla N° 4. RF 01 Presentar el Portal Web Informativo

Fuente: Autor

<b>Nombre</b>
Presentar Inicio de Sesión
<b>Descripción</b>
La aplicación web multimedia permite presentar el inicio de sesión donde se observa el ingreso a la sección administrativa del sitio considerando el tipo de usuario.
<b>Prioridad</b>
Alta

Tabla N° 5. RF 02 Presentar Inicio de Sesión

Fuente: Autor

<b>Nombre</b>
Administra gestión de Usuarios
<b>Descripción</b>
La aplicación web multimedia admite la administración y gestión usuarios, unidades, tipos de persona, personas y grados de persona. (Opción habilitada para el Usuario Administrador).
<b>Prioridad</b>
Alta

Tabla N° 6. RF 03 Administra gestión de Usuarios

Fuente: Autor

<b>Nombre</b>
Generar reportes
<b>Descripción</b>
La aplicación web multimedia realiza la generación de reportes individuales o generales de los miembros de cada unidad o de forma general. (Opción habilitada para el Usuario Administrador).
<b>Prioridad</b>
Alta

Tabla N° 7. RF 04 Generar reportes

Fuente: Autor

<b>Nombre</b>
Configurar parámetros
<b>Descripción</b>
La aplicación web multimedia facilita la configuración de parámetro en costos del servicio y horario de generación de confrontas. (Opción habilitada para el Usuario Administrador).
<b>Prioridad</b>
Alta

Tabla N° 8. RF 05 Configurar parámetros

Fuente: Autor

<b>Nombre</b>
Generar Personas Usuarios
<b>Descripción</b>
La aplicación web multimedia gestiona a cada persona en las unidades por medio del usuario amanuense. (Opción habilitada para el Usuario Amanuense).
<b>Prioridad</b>
Alta

Tabla N° 9. RF 06 Generar Personas Usuarios

Fuente: Autor

<b>Nombre</b>
Generar Novedades
<b>Descripción</b>
La aplicación web multimedia permite la descripción de las novedades diarias previo a la generación de confrontas elaboradas por el usuario amanuense de cada una de las unidades. (Opción habilitada para el Usuario Amanuense).
<b>Prioridad</b>
Alta

Tabla N° 10. RF 07 Generar Novedades

Fuente: Autor

<b>Nombre</b>
Generar Confrontas
<b>Descripción</b>
La aplicación web multimedia permite la creación de confrontas diarias con un día de anticipación de todo el personal por unidad. (Opción habilitada para el Usuario Amanuense).
<b>Prioridad</b>
Alta

Tabla N° 11. RF 08 Generar Confrontas

Fuente: Autor

<b>Nombre</b>
Configurar Parámetros Unidad
<b>Descripción</b>
La aplicación web multimedia, permite configurar parámetros de la unidad bajo su responsabilidad de acuerdo al número de conscriptos y horarios de consumo del servicio de alimentación. (Opción habilitada para el Usuario Amanuense).
<b>Prioridad</b>
Alta

Tabla N° 12. RF 09 Configurar Parámetros Unidad

Fuente: Autor

<b>Nombre</b>
Generar Consumos
<b>Descripción</b>
La aplicación web multimedia genera los consumos del servicio de alimentación individuales o generales del personal de cada unidad. (Opción habilitada para el Usuario Amanuense).
<b>Prioridad</b>
Alta

Tabla N° 13. RF 10 Generar Consumos

Fuente: Autor

<b>Nombre</b>
Visualizar Confrontas Anteriores
<b>Descripción</b>
La aplicación web multimedia brinda la opción de constatar de manera visual las confrontas creadas en fechas anteriores. (Opción habilitada para el Usuario Amanuense).
<b>Prioridad</b>
Alta

Tabla N° 14. RF 11 Visualizar Confrontas Anteriores

Fuente: Autor

<b>Nombre</b>
Visualizar Confrontas Unidades
<b>Descripción</b>
La aplicación web multimedia, permite visualizar las confrontas que fueron creadas por los usuarios amanuenses de cada una de las unidades en los horarios establecidos. (Opción habilitada para el Usuario Ranchero).
<b>Prioridad</b>
Alta

Tabla N° 15. RF 12 Visualizar Confrontas Unidades

Fuente: Autor

<b>Nombre</b>
Visualizar y crear Extra-confrontas
<b>Descripción</b>
La aplicación web multimedia, permite crear un extra-confronta en caso que un usuario no este registrado y requiera un servicio de alimentación desayuno, almuerzo o merienda. (Opción habilitada para el Usuario Ranchero).
<b>Prioridad</b>
Alta

Tabla N° 16. RF 13 Visualizar y crear Extra-confrontas

Fuente: Autor

<b>Nombre</b>
Consolida información
<b>Descripción</b>
La aplicación web multimedia, permite la supervisión y control en la creación de las confrontas de esta manera consolida la información y realiza las adquisiciones. (Opción habilitada para el Usuario Ranchero).
<b>Prioridad</b>
Alta

Fuente: Autor

Tabla N° 17. RF 14 Consolida información

<b>Nombre</b>
Visualizar reportes
<b>Descripción</b>
La aplicación web multimedia, permite visualizar los reportes diarios o generados entre dos fechas de las confrontas de las unidades. (Opción habilitada para el Usuario Ranchero).
<b>Prioridad</b>
Alta

Tabla N° 18. RF 15 Consolida información

Fuente: Autor

### 1.15.2.2 Requerimientos no funcionales

Los requerimientos no funcionales por otra parte son aquellos componentes necesarios para el cumplimiento de estándares de calidad en el diseño del sitio web, utilizando el mismo formato, se describen a continuación:

<b>Nombre</b>
Interfaz sencilla y navegable.
<b>Descripción</b>
La aplicación web multimedia, presenta una facilidad para la navegación presentando elementos útiles para la misma y únicamente los elementos necesarios.
<b>Prioridad</b>
Alta

Tabla N° 19. RNF 01 Interfaz sencilla y navegable

Fuente: Autor

<b>Nombre</b>
Escalabilidad
<b>Descripción</b>
La aplicación web multimedia, permite el desarrollo futuro de nuevas funcionalidades, así como también modificar o eliminar las mismas después de su implementación inicial.
<b>Prioridad</b>
Alta

Tabla N° 20. RNF 02 Escalabilidad

Fuente: Autor

<b>Nombre</b>
Seguridad de Acceso
<b>Descripción</b>
La aplicación web multimedia solicita claves de acceso para el ingreso a la misma según el tipo de usuario que inicie sesión.
<b>Prioridad</b>
Alta

Tabla N° 21. RNF 03 Seguridad de Acceso

Fuente: Autor

### 1.15.3 Casos de Uso

Los casos de uso representan un método que especifica cada uno de los requisitos que contiene un sistema con relación a la interacción con los actores, estos se los puede representar con gráficos secuenciales. Para el presente trabajo se han evidenciado a cuatro usuarios descritos a continuación:

**Usuario Administrador-Técnico:** es la persona que tiene como funcionalidad brindar soporte técnico a todo el sistema para su correcto funcionamiento, está representado por el especialista con conocimiento pleno de dicha aplicación.

**Usuario Administrador:** referente a la persona encargada de administrar a los usuarios que podrán ingresar al sistema, elaborar reportes de las comandas diarias individuales y a nivel general. Por tanto, la función principal del administrador será la de configurar parámetros y establecer la adecuada planificación para el manejo eficiente del sitio web.

**Usuario Amanuense:** es el usuario que realiza la mayor cantidad de tareas para la planificación de las confrontas dentro del sistema web. Entre sus funciones se encuentran el manejo del listado del personal con el cual verifica las confrontas necesarias, novedades ocurridas, los consumos efectuados y los valores a cancelar de los consumidores.

**Usuario Ranchero:** es la persona que se encarga de monitorear las comandas y de crear pedidos extras en el caso de algún nuevo comensal. Además, visualiza la información de los reportes generados por el amanuense.

## Caso de Uso 1. Ingreso del Usuario al Sistema

**Objetivo:** Permitir a los Usuarios Registrador el ingreso al sistema

**Proceso:**

El Usuario Registrado solicita el ingreso dado su nombre de usuario y su clave. Esta información es buscada en la base de datos.

Se valida la información y una vez validada presenta el resultado sea aceptado y negado el ingreso.

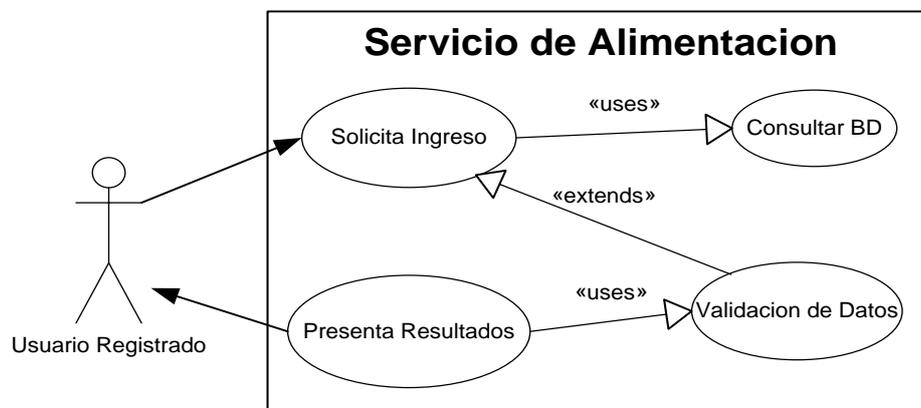


Gráfico N° 23. Caso de uso-ingreso al sistema

Fuente: Autor

## Caso de Uso 2. Gestión de Usuarios

**Objetivo:** Permitir al Administrador la Creación, Cambio (datos personales), eliminación de un usuario al igual que visualizarlo.

**Proceso:**

El Administrador escoge la opción a ser requerida, la misma que es procesada y enviada a buscar en la BD para mostrar los requerimientos solicitados.

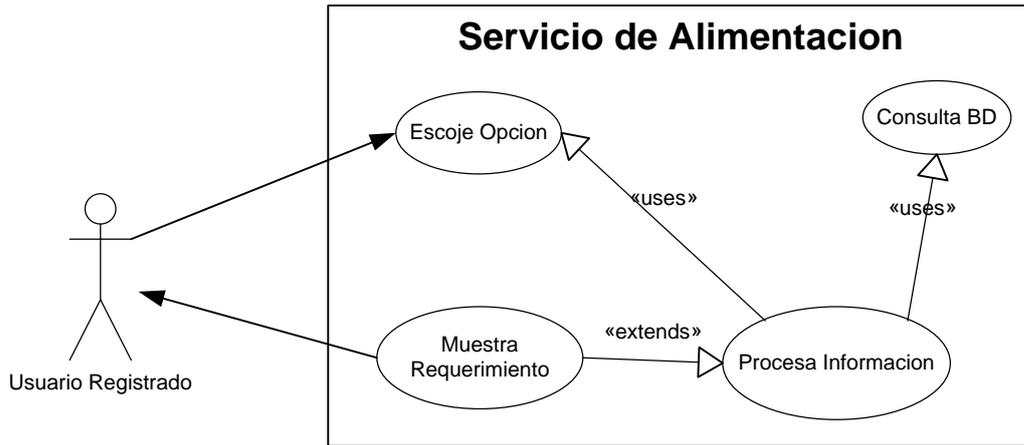


Gráfico N° 24. Caso de uso-gestión de usuarios

Fuente: Autor

**Caso de Uso 3. Generación de Reportes**

**Objetivo:** Grabar en la base de Datos la información ingresada

**Proceso:**

Solicitan las opciones de Reportes, la misma que en una de las opciones Visualiza la información que regreso de la BD, en el otro caso Crea Reportes, esta información validada regresa e ingresa en la BD y presenta reportes.

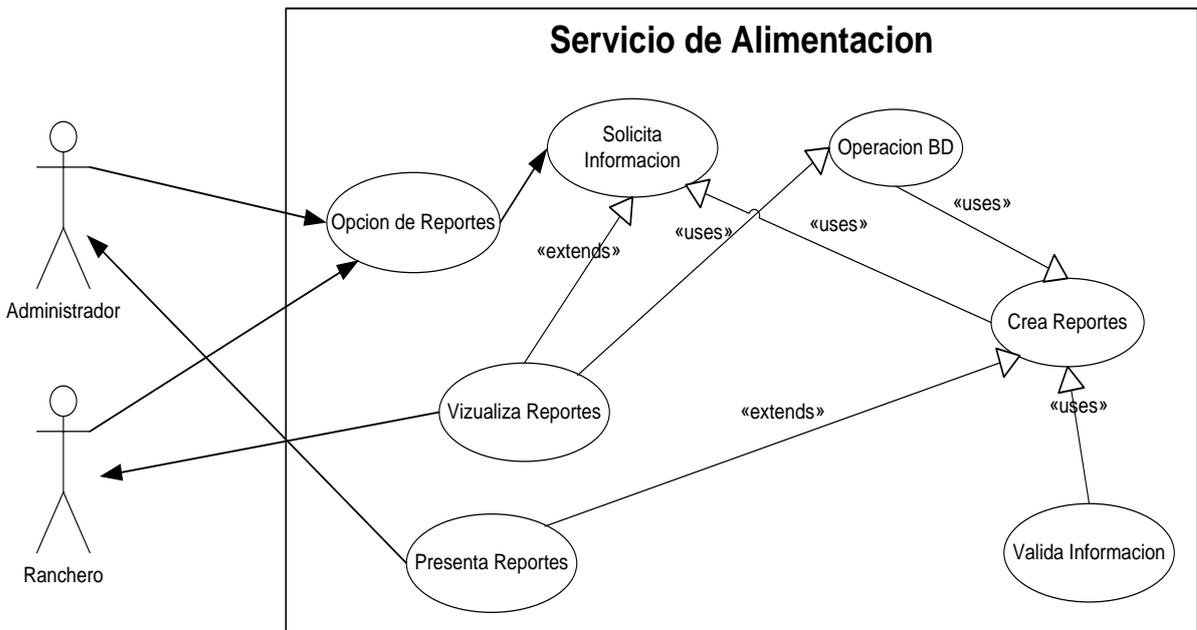


Gráfico N° 25. Caso de uso-gestión de reportes

Fuente: Autor

#### Caso de Uso 4. Generación de Catálogos

**Objetivo:** Mostrar al Administrador la información requerida para el reporte.

**Proceso:** Utiliza el caso de uso anterior, para realizar una solicitud de los catálogos, y la BD devuelve la información, presenta la opción al administrador, se procesa el requerimiento en la BD, presenta los requerimientos al administrador.

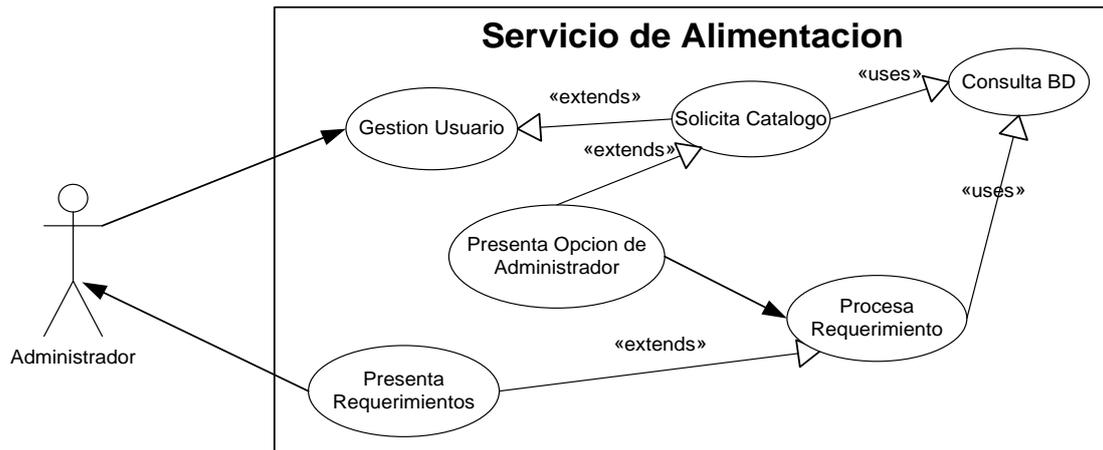


Gráfico N° 26. Caso de uso-gestión de catálogos

Fuente: Autor

#### Caso de Uso 5. Configuración de Parámetros

**Objetivo:** Presentar los parámetros para tener un buen control de datos.

**Proceso:** Selecciona primero la Opción Requerida, mediante esta solicitud, se ingresa los parámetros a ser validados, se procede a enviar a la BD, y devuelve para los datos ser procesados.

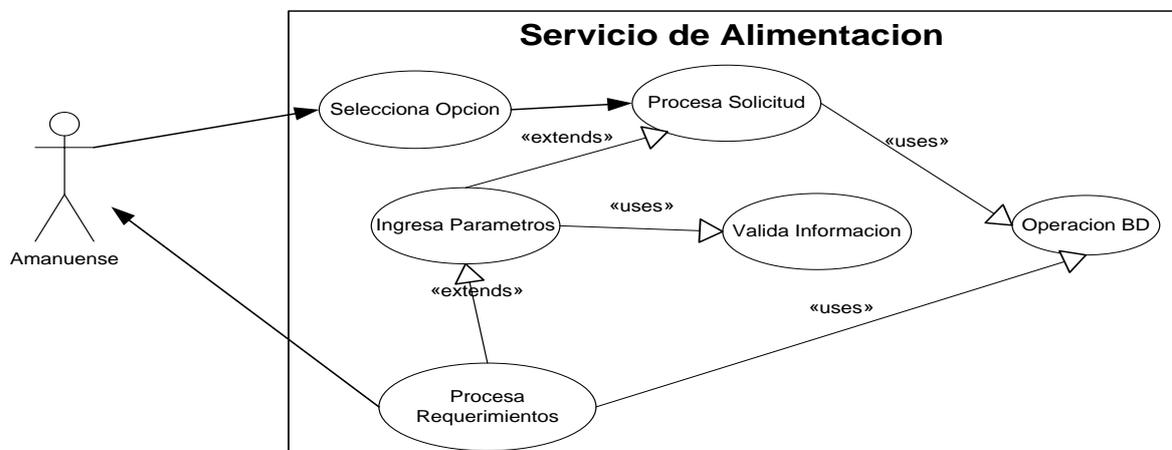


Gráfico N° 27. Caso de uso-configuración de parámetros

Fuente: Autor

## Caso de Uso 6. Proceso de Consumos

**Objetivo:** Generar los Consumos Individuales y/o generales

**Proceso:** Solicita la generación, que es adquirida de la base de datos, esta información es validada para luego generar y presentar la información.

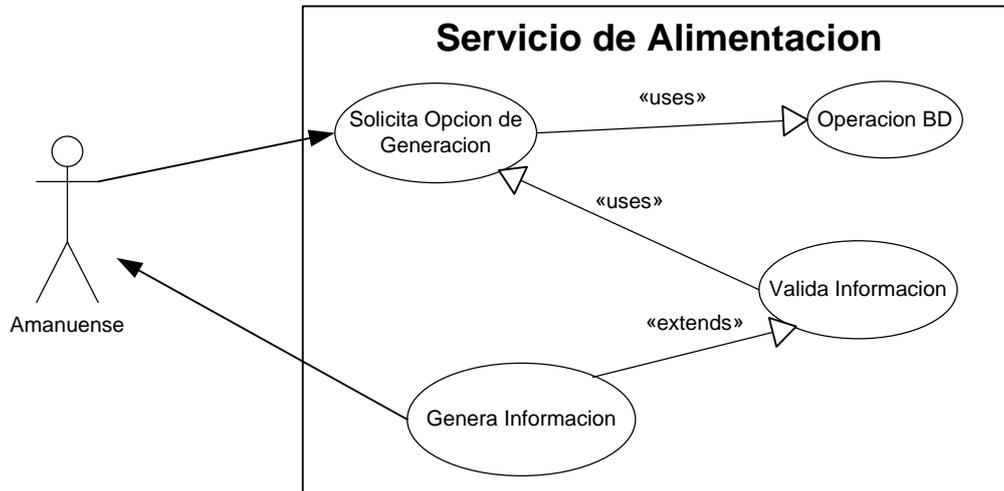


Gráfico N° 28. Caso de uso-proceso de consumos

Fuente: Autor

## Caso de Uso 7. Procesa Confrontas

**Objetivo:** La visualización, creación y creación de confrontas.

**Proceso:** Escoge la Opción este se bifurca de la solicitud de la información requerida, esta se solicita a la BD, el primer caso solo visualiza y presenta la información, en la segunda permite la creación de confrontas esta información, es guardada en la BD después de ser validada, y por último presenta la información.

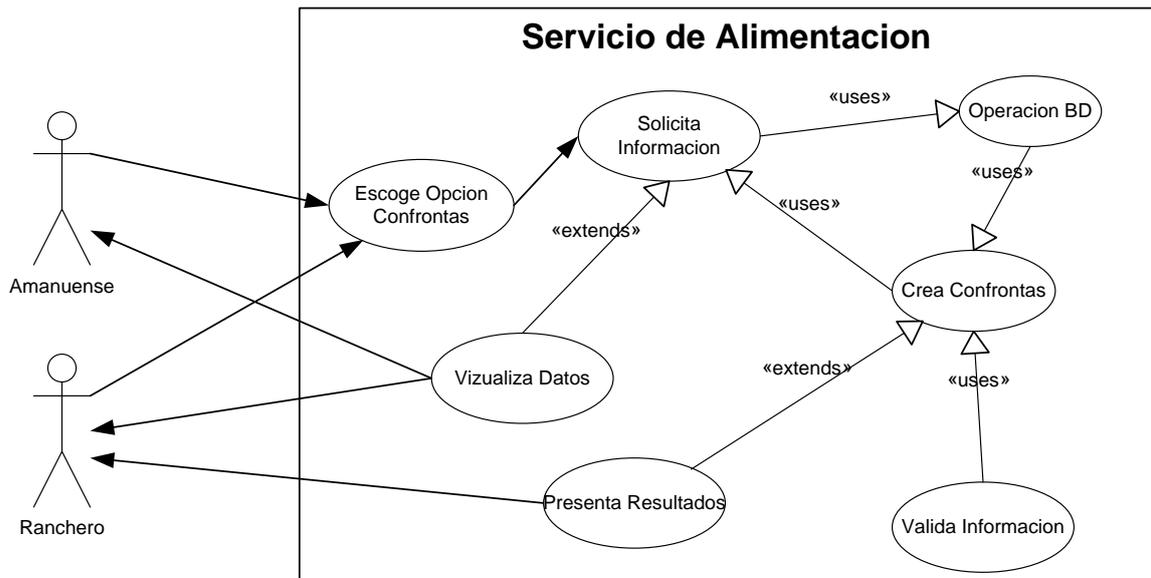


Gráfico N° 29. Caso de uso-configuración de parámetros

Fuente: Autor

## 1.1 DISEÑO

El diseño de sistema web describe ciertos componentes necesarios para su desarrollo entre ellos se encuentran:

### 1.1.1 Diseño Conceptual

El diseño conceptual muestra los diagramas técnicos de la base de datos requerida en el sistema web teniendo como principal función la administración y acumulación de información referente a la gestión de confrontas de cada unidad perteneciente a la Caballería Blindada N°11 "Galápagos".

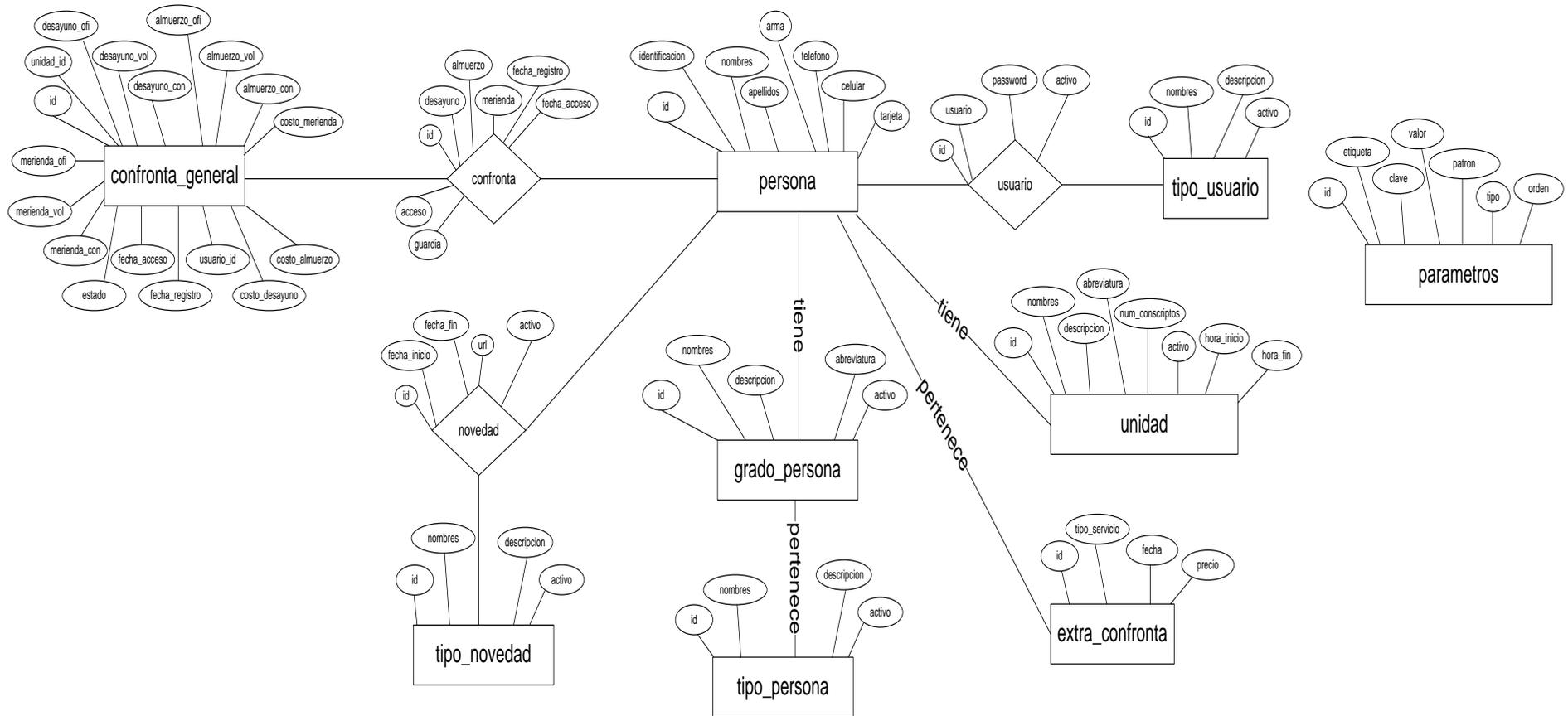


Gráfico N° 30. Diseño conceptual

Fuente: Autor

### 1.1.2 Modelo Relacional

Este modelo relaciona la información emitida en los diagramas técnicos del diseño conceptual para estandarizarlos cambiando ciertas denominaciones como las entidades en tablas y las propiedades en campos, lo que permite otorgar una característica a cada campo como se verifica a continuación.

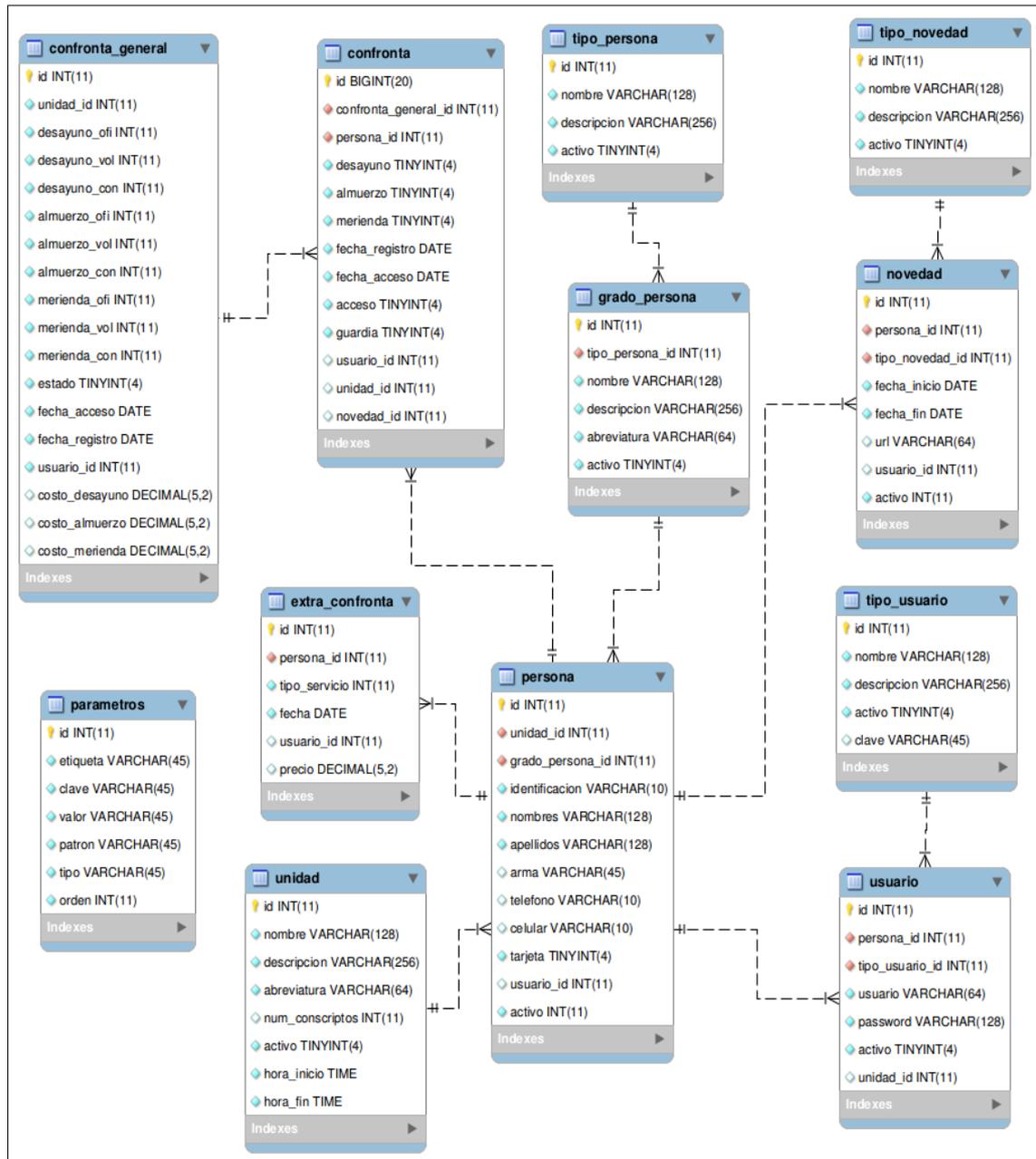


Gráfico N° 31. Modelo relacional

Fuente: Autor

### 1.1.3 Diccionario de Datos

El diccionario de datos representa aquellas características lógicas que cada uno de los campos de las tablas descritas anteriormente deben cumplir, por ejemplo:

PK= Primario (Identificar elementos de cada tabla)

NN= No nulo (Campo obligatorio)

AI= auto incremental (de almacenamiento automático)

De esta manera se describe a continuación por medio de tablas estructuradas el diccionario de datos

#### **Tablas**

confronta

confronta\_general

extra\_confronta

grado\_persona

novedad

parametros

persona

tipo\_novedad

tipo\_persona

tipo\_usuario

unidad

usuario

Confronta

Column name	DataType	PK	NN	AI	Default	Comment
id	BIGINT(20)	✓	✓	✓		
confronta_general_id	INT(11)		✓			
persona_id	INT(11)		✓			
desayuno	TINYINT(4)		✓		'0'	
almuerzo	TINYINT(4)		✓		'0'	
merienda	TINYINT(4)		✓		'0'	
fecha_registro	DATE		✓			
fecha_acceso	DATE		✓			
acceso	TINYINT(4)		✓		'1'	
guardia	TINYINT(4)		✓		'0'	
usuario_id	INT(11)				NULL	
unidad_id	INT(11)				NULL	
novedad_id	INT(11)				NULL	

confronta\_general

Column name	DataType	PK	NN	AI	Default	Comment
id	INT(11)	✓	✓	✓		
unidad_id	INT(11)		✓		'0'	
desayuno_ofi	INT(11)		✓		'0'	
desayuno_vol	INT(11)		✓		'0'	
desayuno_con	INT(11)		✓		'0'	
almuerzo_ofi	INT(11)		✓		'0'	
almuerzo_vol	INT(11)		✓		'0'	
almuerzo_con	INT(11)		✓		'0'	
merienda_ofi	INT(11)		✓		'0'	
merienda_vol	INT(11)		✓		'0'	
merienda_con	INT(11)		✓		'0'	
estado	TINYINT(4)		✓		'0'	
fecha_acceso	DATE		✓			
fecha_registro	DATE		✓			
usuario_id	INT(11)		✓		'0'	
costo_desayuno	DECIMAL(5,2)				'0.00'	

costo_almuerzo	DECIMAL(5,2)				'0.00'	
costo_merienda	DECIMAL(5,2)				'0.00'	

extra\_confronta

Column name	DataType	PK	NN	AI	Default	Comment
id	INT(11)	✓	✓	✓		
persona_id	INT(11)		✓			
tipo_servicio	INT(11)		✓			
fecha	DATE		✓			
usuario_id	INT(11)				NULL	
precio	DECIMAL(5,2)				'0.00'	

grado\_persona

Column name	DataType	PK	NN	AI	Default	Comment
id	INT(11)	✓	✓	✓		
tipo_persona_id	INT(11)		✓			
nombre	VARCHAR(128)		✓			
descripcion	VARCHAR(256)		✓			

abreviatura	VARCHAR(64)		✓			
activo	TINYINT(4)		✓		'1'	

Novedad

Column name	DataType	PK	NN	AI	Default	Comment
id	INT(11)	✓	✓	✓		
persona_id	INT(11)		✓			
tipo_novedad_id	INT(11)		✓			
fecha_inicio	DATE		✓			
fecha_fin	DATE		✓			
url	VARCHAR(64)				NULL	
usuario_id	INT(11)				NULL	
activo	INT(11)		✓		'1'	

Parámetros

Column name	DataType	PK	NN	AI	Default	Comment
id	INT(11)	✓	✓	✓		
etiqueta	VARCHAR(45)		✓			

clave	VARCHAR(45)		✓			
valor	VARCHAR(45)		✓			
patron	VARCHAR(45)		✓			
tipo	VARCHAR(45)		✓			
orden	INT(11)		✓			

Persona

Column name	DataType	PK	NN	AI	Default	Comment
id	INT(11)	✓	✓	✓		
unidad_id	INT(11)		✓			
grado_persona_id	INT(11)		✓			
identificacion	VARCHAR(10)		✓			
nombres	VARCHAR(128)		✓			
apellidos	VARCHAR(128)		✓			
arma	VARCHAR(45)				NULL	
telefono	VARCHAR(10)				NULL	
celular	VARCHAR(10)				NULL	
tarjeta	TINYINT(4)		✓		'0'	

usuario_id	INT(11)				'0'	
activo	INT(11)		✓		'1'	

tipo\_novedad

Column name	DataType	PK	NN	AI	Default	Comment
id	INT(11)	✓	✓	✓		
nombre	VARCHAR(128)		✓			
descripcion	VARCHAR(256)		✓			
activo	TINYINT(4)		✓		'1'	

tipo\_persona

Column name	DataType	PK	NN	AI	Default	Comment
id	INT(11)	✓	✓	✓		
nombre	VARCHAR(128)		✓			
descripcion	VARCHAR(256)		✓			
activo	TINYINT(4)		✓		'1'	

tipo\_usuario

--

Column name	DataType	PK	NN	AI	Default	Comment
id	INT(11)	✓	✓	✓		
nombre	VARCHAR(128)		✓			
descripcion	VARCHAR(256)		✓			
activo	TINYINT(4)		✓		'1'	
clave	VARCHAR(45)				NULL	

Unidad

Column name	DataType	PK	NN	AI	Default	Comment
id	INT(11)	✓	✓	✓		
nombre	VARCHAR(128)		✓			
descripcion	VARCHAR(256)		✓			
abreviatura	VARCHAR(64)		✓			
num_conscriptos	INT(11)				'0'	
activo	TINYINT(4)		✓		'1'	
hora_inicio	TIME		✓			
hora_fin	TIME		✓			

Usuario

Column name	DataType	PK	NN	AI	Default	Comment
id	INT(11)	✓	✓	✓		
persona_id	INT(11)		✓			
tipo_usuario_id	INT(11)		✓			
usuario	VARCHAR(64)		✓			
password	VARCHAR(128)		✓			
activo	TINYINT(4)		✓		'1'	
unidad_id	INT(11)				'0'	

Tabla N° 22. Diccionario de datos

Fuente: Autor

#### 1.1.4 Diseño de Interfaces

**Sistema web:** Requerimiento requerido como administrador del manejo de la información y generación de confrontas dentro de la Caballería N° 11 Galápagos

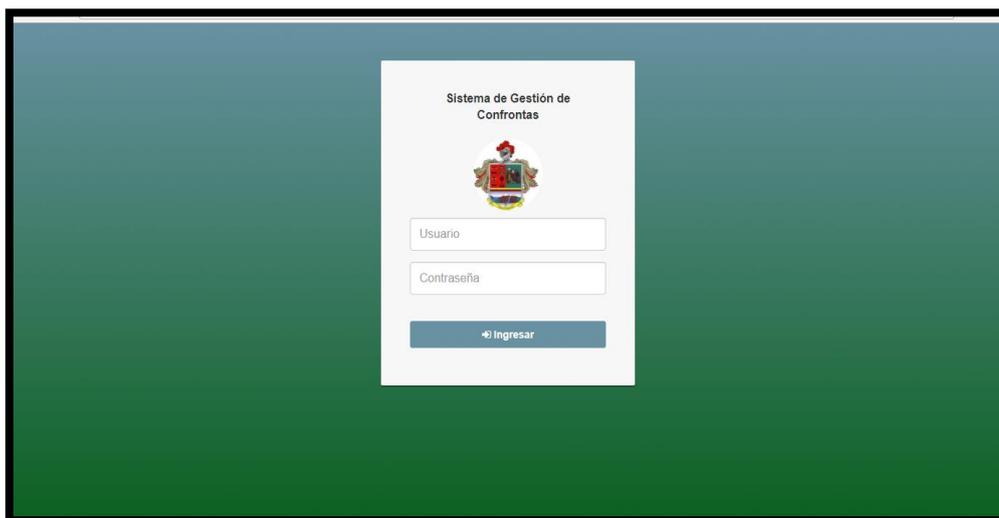


Gráfico N° 32. Pantalla principal del sistema web

Fuente: Autor

**Inicio de sesión:** Diseñada para el acceso de los involucrados a la aplicación mediante un usuario y una clave que al ser correctos podrá ingresar al sistema web.



Gráfico N° 33. Inicio de sesión del sistema web

Fuente: Autor

**Usuario Administrador:** El sistema posee varios usuarios, cuando el denominado administrador tenga ingreso al sistema se desplegarán opciones como la gestión de usuarios, catálogos, el consolidado, la lista de la persona, la revisión de reportes y la configuración, como se muestra a continuación.



Gráfico N° 34. Usuario administrador en el sistema web

Fuente: Autor

**Usuario Amanuense:** este usuario es el que realiza la necesidad de las confrontas y tiene como opciones inicio, persona, extra confronta, consolidad y la generación de reportes.



Gráfico N° 35. Usuario amanuense en el sistema web

Fuente: Autor

**Usuario Ranchero:** este usuario enfocado al control de las confrontas creadas por el amanuense, al ingresar al sistema cuenta con las opciones de inicio, personal, novedad, confronta, unidad y los consumos registrados de forma diaria.

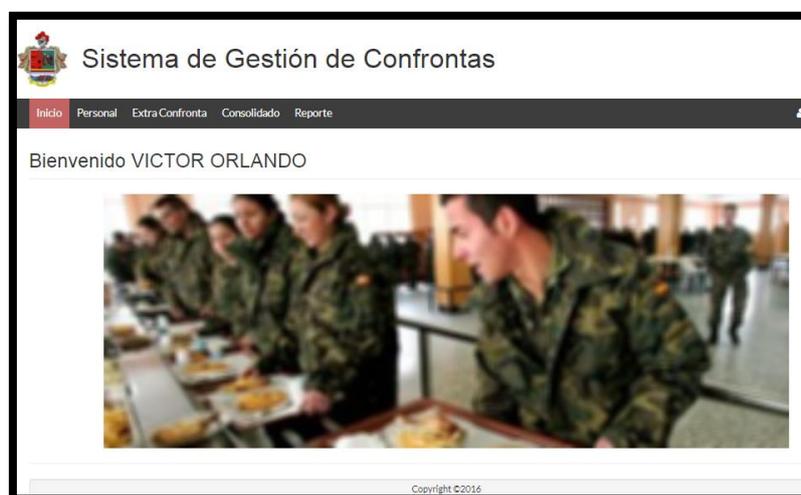


Gráfico N° 36. Usuario ranchero en el sistema web

Fuente: Autor

**CAPÍTULO IV**

**IMPLEMENTACIÓN DEL SISTEMA WEB**

Para la implementación del sistema Web fueron requeridas varias herramientas que a continuación se describen las características más importantes que permitieron la implementación del sistema web.

## 1.2 HERRAMIENTAS DE DESARROLLO

### 1.2.1 Instalación del WampServer

Para instalar el entorno de desarrollo web para Windows llamado WampServer en una de sus diferentes versiones en primera instancia debemos tomar en cuenta las propiedades en cuanto al tipo de sistema operativo de nuestro pc, si el mismo es de 32 o 64 bits.

Luego procedemos a descargar el software de este entorno de desarrollo tomando en cuenta que es un software libre que se lo puede descargar fácilmente del internet.

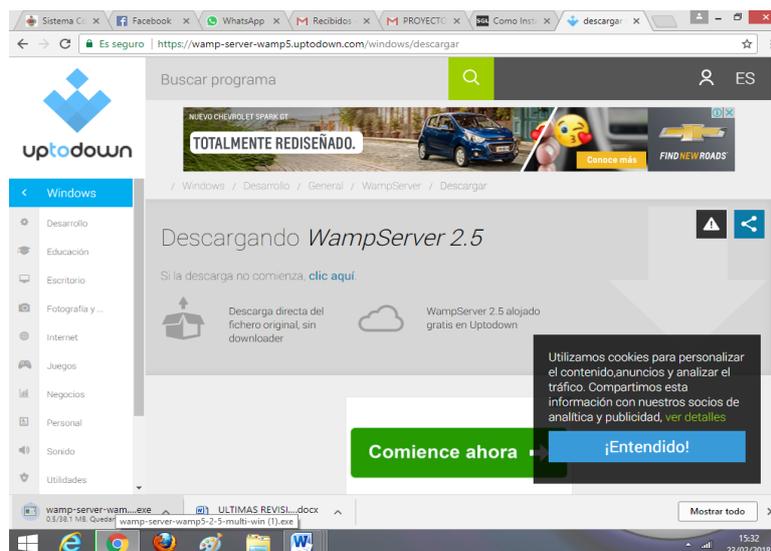


Gráfico N° 37. Descarga WampServer

Fuente: Autor

Una vez que contamos con el software realizamos el proceso de instalación del mismo.

Ya descargado lo ejecutamos con doble clic nos debe salir el Wizard de Instalación, clic en siguiente o Next.



Gráfico N° 38. Ejecución de WampServer

Fuente: Autor

Aceptamos los términos y condiciones para seguir con la instalación.

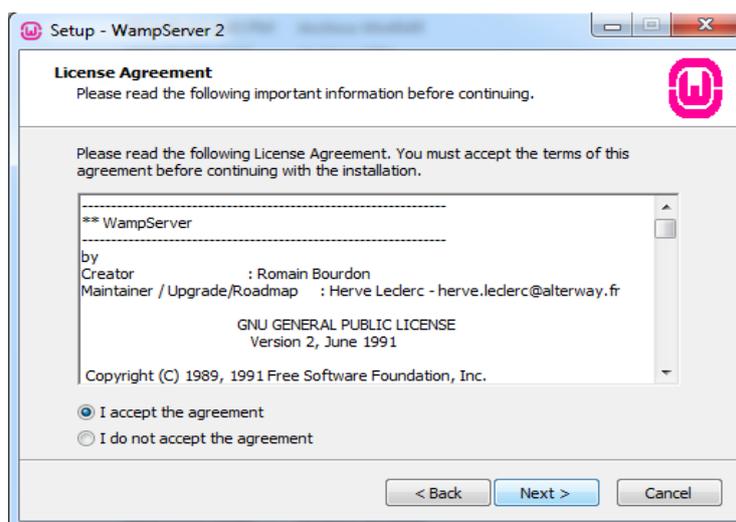


Gráfico N° 39. Aceptación de términos y condiciones de Wamp Server

Fuente: Autor

Seleccionamos el destino de instalación, por defecto lo dejamos así o si lo prefiere puede seleccionar otra ubicación. Next.

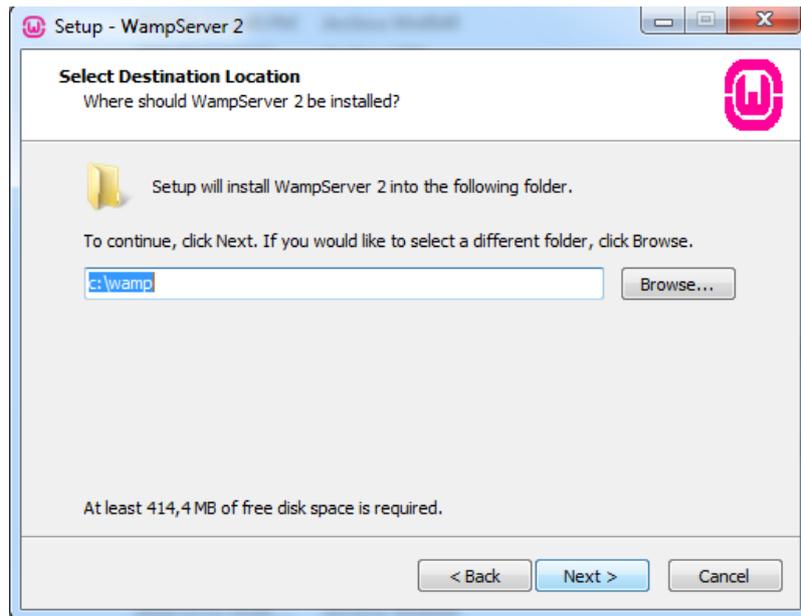


Gráfico N° 40. Selección de destino de instalación del WampServer

Fuente: Autor

Para crear un acceso directo seleccionamos la primera opción.

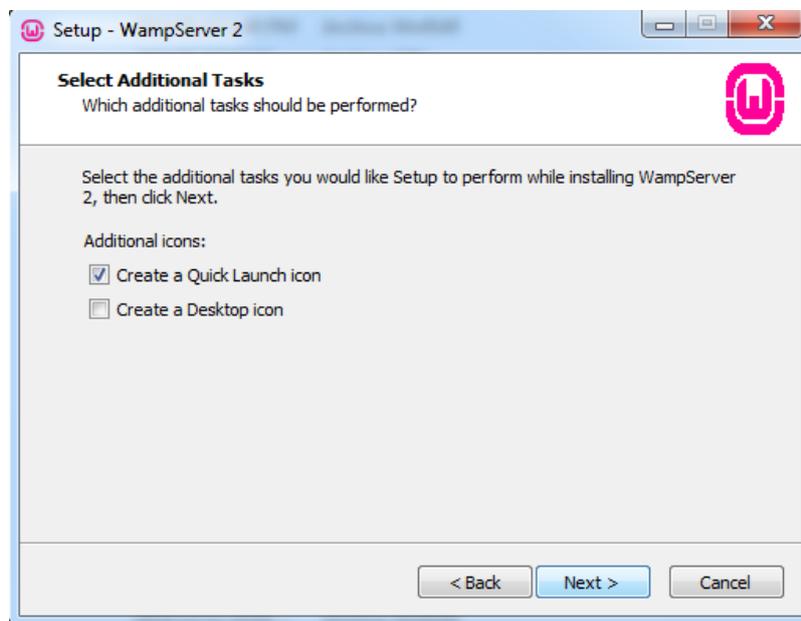


Gráfico N° 41. Creación de acceso directo del WampServer

Fuente: Autor

Damos siguiente.

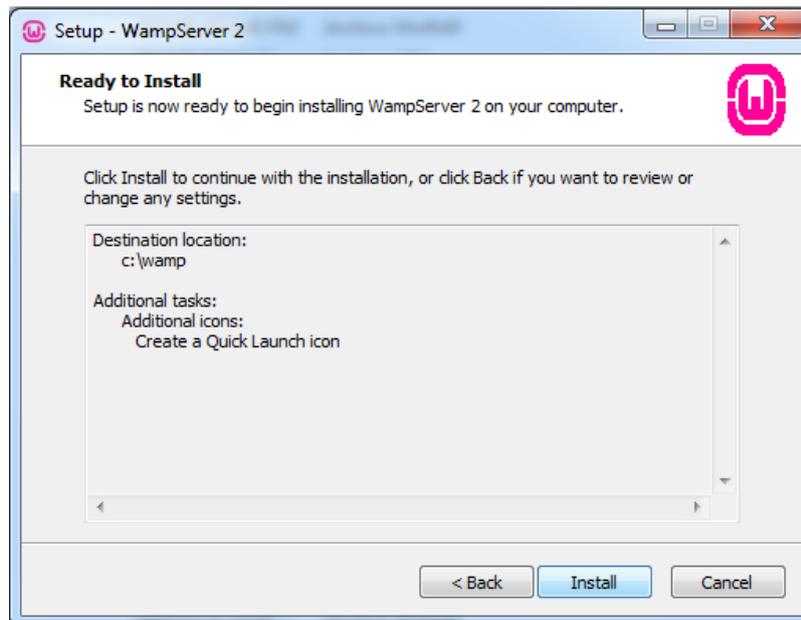


Gráfico N° 42.Instalación de WampServer

Fuente: Autor

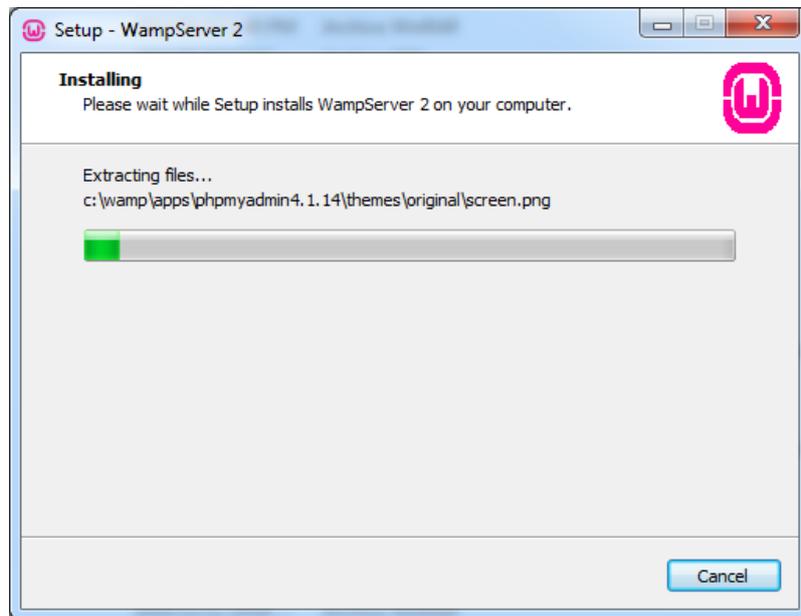


Gráfico N° 43. Proceso de instalación de WampServer

Fuente: Autor

Determinamos el navegador a utilizar.

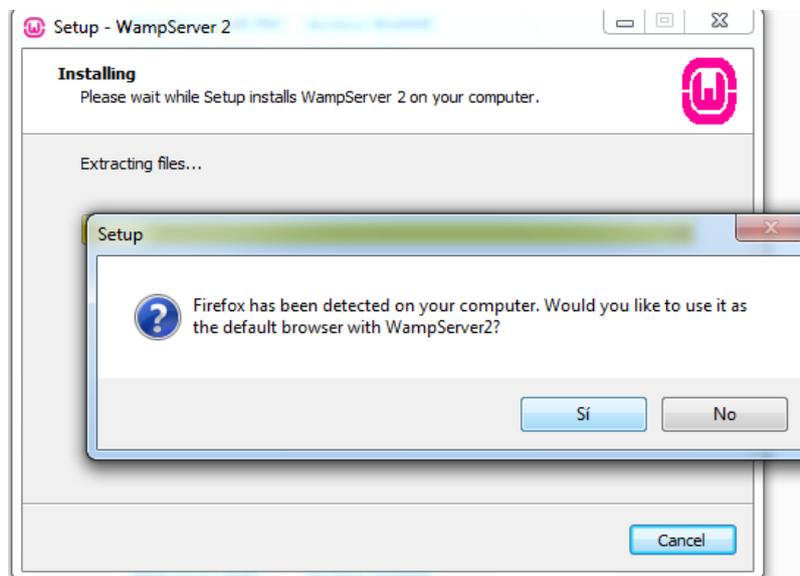


Gráfico N° 44. Determinación del navegador a utilizar para WampServer

Fuente: Autor

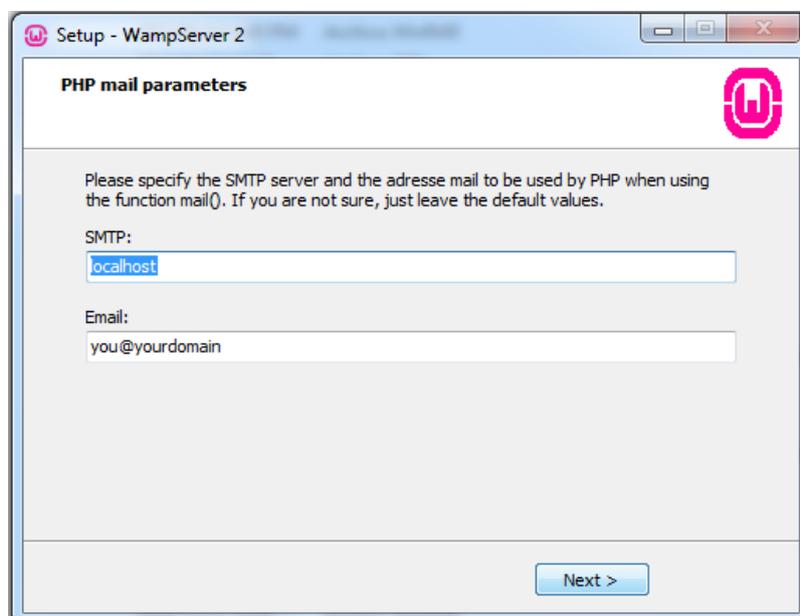


Gráfico N° 45. Parámetros para uso de PHP en WampServer

Fuente: Autor

El proceso de instalación ha terminado.



Gráfico N° 46. Instalación completa de WampServer

Fuente: Autor

WampServer se ejecutará en segundo plano en la barra de tareas y se tornara de un color verde.

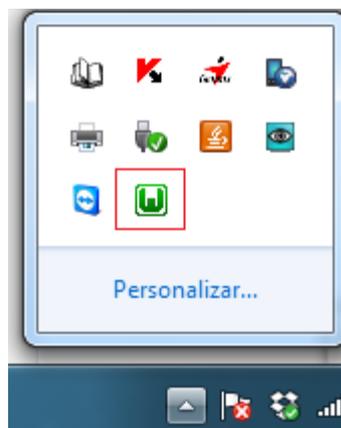


Gráfico N° 47. Ejecución de WampServer en segundo plano

Fuente: Autor

## 1.2.2 Instalación del entorno de desarrollo NetBean

Para la instalación de este entorno de desarrollo que utilizaremos como herramienta para la programación, compilación y ejecutar lo desarrollado para lo cual detallaremos los pasos para la instalación de esta herramienta.

Para ello lo descargamos e instalar Java JDK 7 del siguiente enlace:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html?ssSourceSiteId=otnes>

Dependiendo de si el sistema operativo es de 32 o 64 bits procedemos a descargar el software ejecutable.

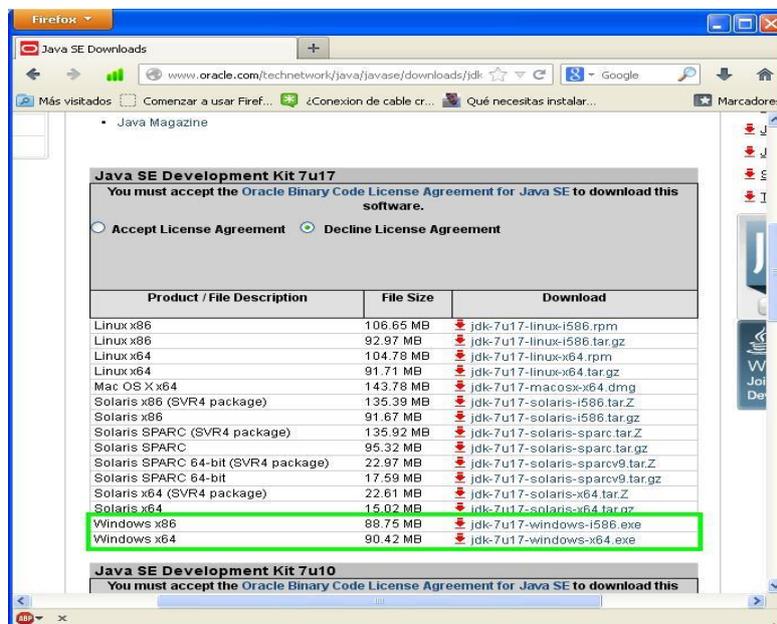


Gráfico N° 48. Descarga de NetBeans

Fuente: Autor

El siguiente paso será el de Descarga e instalar NetBeans.

En este paso nos dirigimos al siguiente enlace:

<https://netbeans.org/downloads/index.html>

Seleccionamos las opciones tal y como se muestran en el gráfico

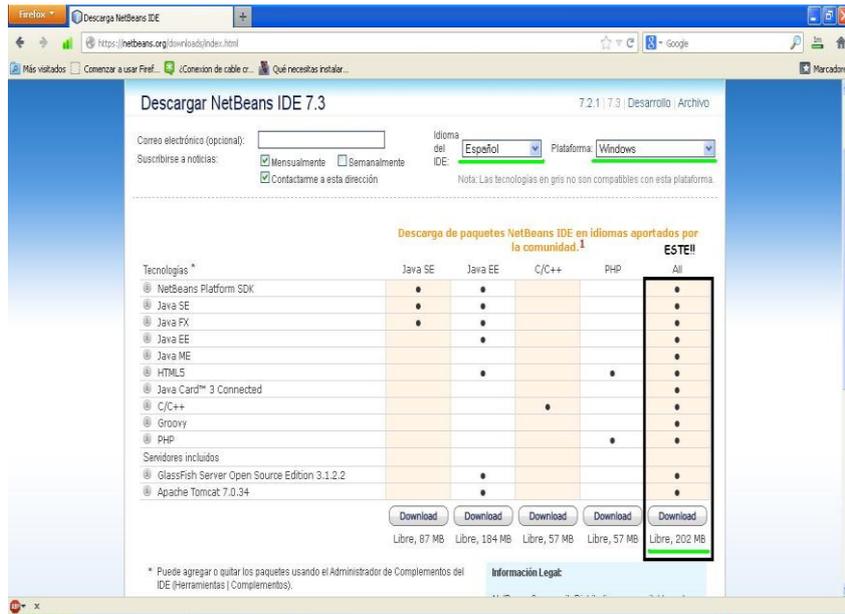


Gráfico N° 49. Opciones de descarga de NetBeans

Fuente: Autor

Descarga el depurador (Debugger)

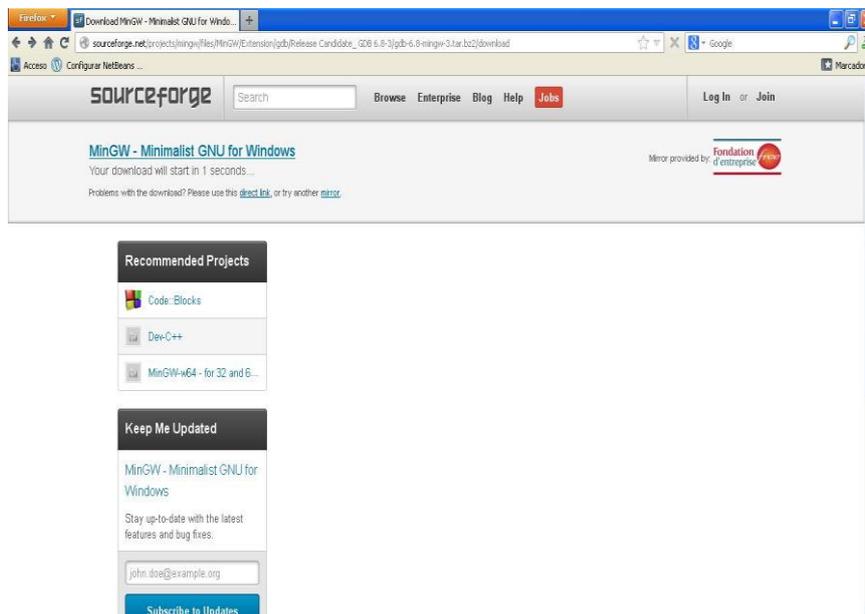


Gráfico N° 50. Descarga del depurador

Fuente: Autor

## Descarga e instalación del compilador

Hay varios compiladores dependiendo de en qué se requiera a programar.

Utilizaremos MinGW ya que es más sencillo de instalar.

Para descargarlo vamos al siguiente enlace:

<http://sourceforge.net/projects/mingw/files/>

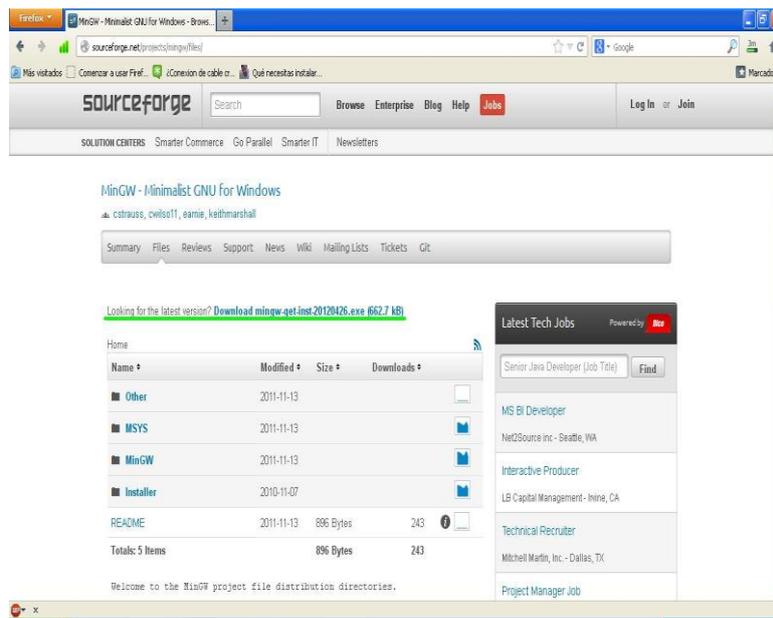


Gráfico N° 51. Descarga del compilador para NetBeans

Fuente: Autor

Necesitamos modificar la variable del sistema llamada “Patch” poniendo al final de la variable lo siguiente:

```
;C:\MinGW\bin;C:\MinGW\msys\1.0\bin;
```

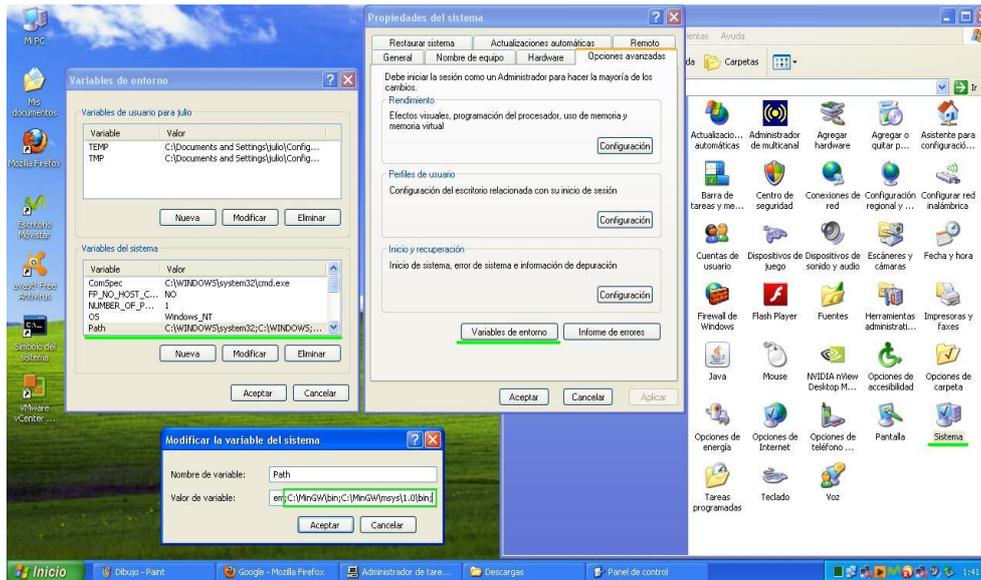


Gráfico N° 52. Configuración Patch

Fuente: Autor

### 1.3 ARQUITECTURA DEL SISTEMA WEB

La arquitectura del sistema web requiere de varios componentes, funciones y actores para su correcta operatividad. En este apartado se describe la estructura, el funcionamiento y la interacción de todos los actores cuya finalidad será la de resolver los problemas o necesidades para lo cual se requirió la elaboración del sistema.

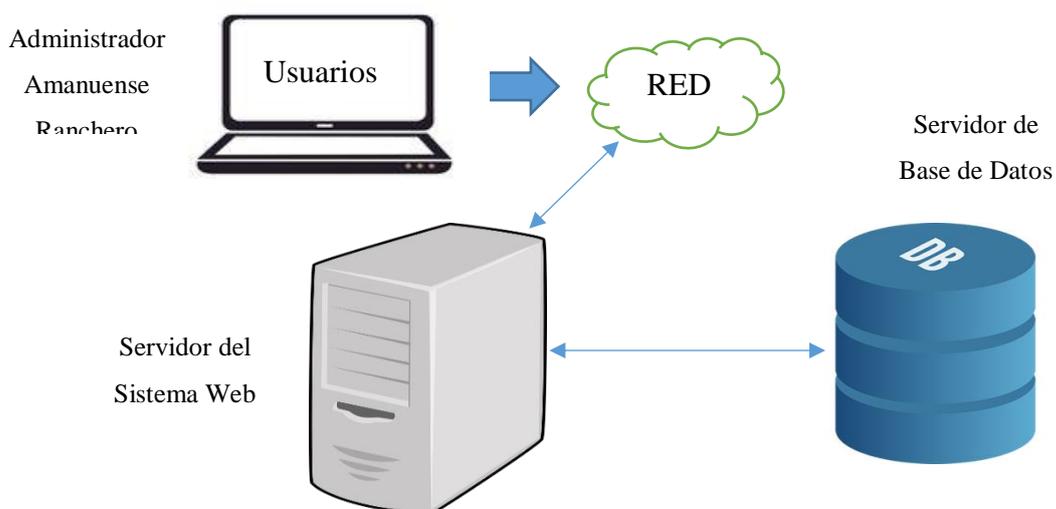


Gráfico N° 53. Arquitectura del sistema web

Fuente: El Autor

La gráfica muestra la interacción de los usuarios con la red en primera instancia, que posteriormente ingresará al sistema web mostrando la información que así lo requiera y generar el requerimiento respectivo para que este pueda ser almacenado en la base de datos, después de este proceso se muestra al usuario el requerimiento solicitado haciendo en un proceso de dos vías.

## 1.4 IMPLEMENTACIÓN DEL SISTEMA

Después de la etapa de análisis y diseño del sistema web, inicia el proceso de implementación que está relacionado directamente con la definición de los diferentes subsistemas que determinan a detalle las funcionalidades específicas del sistema haciendo que su desarrollo se lo realice de manera ágil, así como también el descubrimiento de las fallas existentes en las pruebas de aplicación.

### 1.4.1 Definición de los subsistemas de implementación

Los subsistemas de implementación vienen delimitados por las características de diseño agrupados de acuerdo a la relación existente entre las clases resumidos en un empaquetamiento que determinan tareas comunes para la ejecución de los procesos.

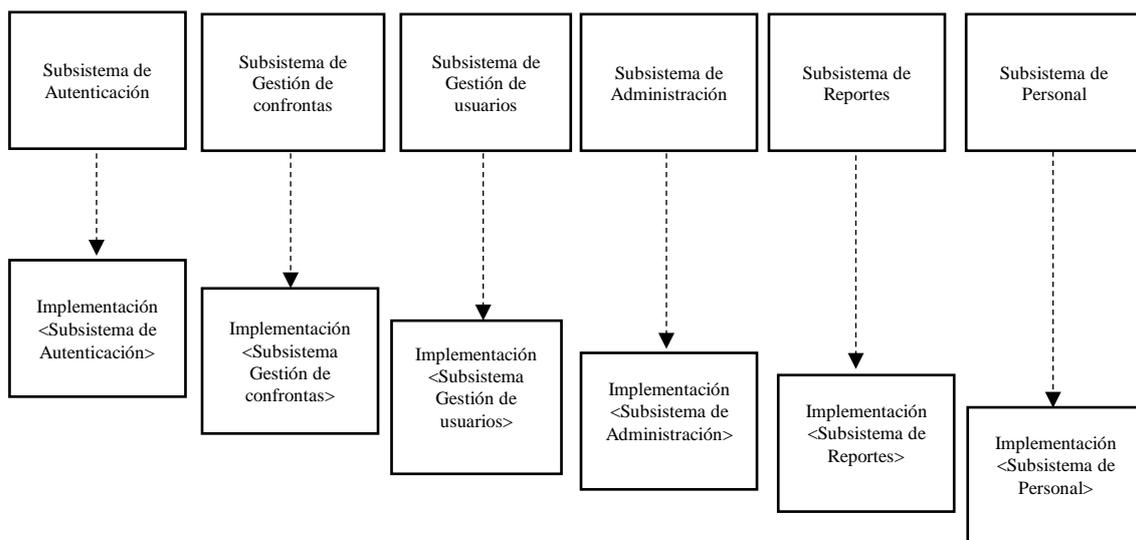


Gráfico N° 54. Subsistemas del sistema web

Fuente: El Autor

### 1.4.2 Desarrollo de la aplicación implementación

El desarrollo del sistema web representa la puesta en marcha de todos los procesos de análisis, diseño y las herramientas descritas anteriormente. Dando así un sistema web interactivo con el usuario capaz de controlar en tiempos diarios las comandas que requiere el personal que integra la Caballería N° 11 “Galápagos” de la ciudad de Riobamba. Además, aprobará a los encargados o administradores tener un reporte de la alimentación para determinar acciones o decisiones pertinentes para el mejoramiento de dicho servicio.

El sistema web de forma inicial muestra la pantalla principal que representa la opción de registro de datos para el ingreso.



Gráfico N° 55. Página principal del sitio web

Fuente: El Autor

El contenido restante del sistema web se detalla en el manual de Usuario (ANEXO 1).

### 1.4.3 Implementación del sistema web

La implementación del sistema web se lo realizó mediante una red interna cuya configuración residió en un computador personal que hizo las funciones de servidor. Los procedimientos para la mencionada configuración se señalan en el manual de instalación. (ANEXO 2).

## 1.5 PRUEBAS DEL SISTEMA WEB

Las pruebas realizadas para la comprobación de que el sistema web funciona correctamente y soportan todas las necesidades de los usuarios fueron las siguientes:

**Funcionalidad:** Cumple con cada uno de los requerimientos y necesidades que el usuario así lo demande.

Así por ejemplo, cuando el usuario amanuense requiere la realización de una nueva confronta, ingresa al sistema y con la opción confronta inmediatamente se desplegarán opciones para el cumplimiento de dicha actividad.



Unidad	Desayunos			Almuerzos			Meriendas			Fecha	Acciones
	OFI	VOL	CON	OFI	VOL	CON	OFI	VOL	CON		
EC-11	1	59	20	2	59	18	2	59	20	2016-09-27	[Acciones]
EC-11	1	62	28	2	61	28	2	60	28	2016-10-06	[Acciones]
EC-11	2	62	28	2	62	28	2	62	28	2017-05-11	[Acciones]
EC-11	2	58	28	1	56	28	2	60	28	2017-10-23	[Acciones]
EC-11	1	61	28	1	61	28	2	61	28	2017-10-24	[Acciones]
EC-11	0	5	28	2	62	28	0	8	28	2017-12-01	[Acciones]

Gráfico N° 56. Prueba de funcionalidad

Fuente: El Autor

**Estabilidad:** El sistema contiene un sistema de control de fallos por medio de mensajes cuando los datos se encuentran mal tipificados, por ejemplo:

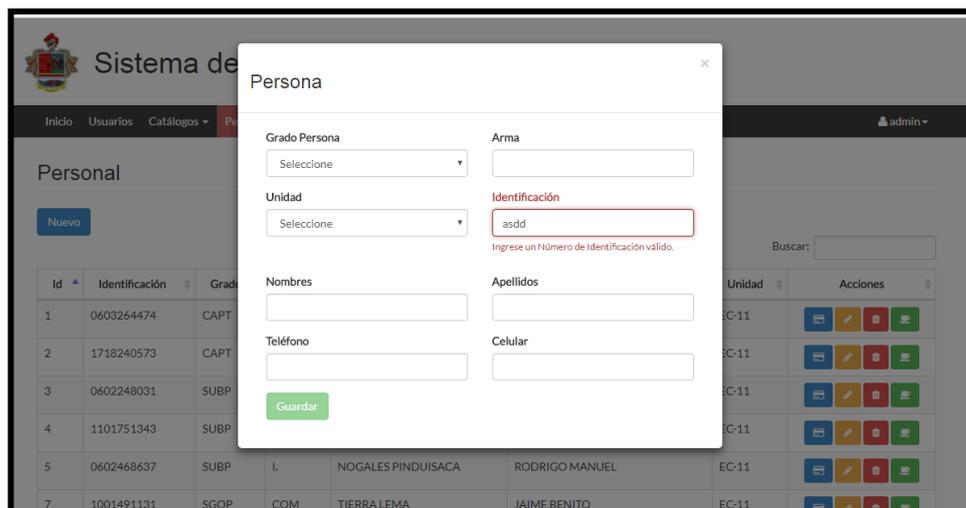


Gráfico N° 57. Prueba de estabilidad

Fuente: El Autor

**Tiempo:** Existen varios procesos que toman un mínimo de segundo en ser procesados, como por ejemplo el inicio de sesión, los procesos internos, las consultas y las distintas cargas según las tareas requeridas, sin embargo, el sistema se relacionará directamente con el ancho de banda de internet que se posea.

**Escalabilidad:** el sistema es adaptable a los cambios que se requieran, muestra de un ejemplo es la incorporación del código de barras, aunque esta es una opción que posee el sistema no está habilitada actualmente sino quedará como una idea posterior para que la Caballería N°11 Galápagos pueda implementarlo cuando crea pertinente.



Gráfico N° 58. Prueba de escalabilidad

Fuente: El Autor

**Seguridad de Acceso:** Representa aquella prueba que delimita y autentica los datos otorgados para los usuarios emitidos por medio de una cuenta y una clave, si la información no es correcta el sistema web arrojará un error que limitará el acceso al mismo.

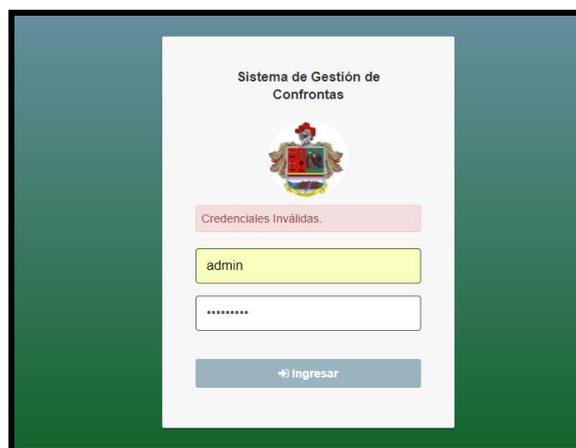


Gráfico N° 59. Prueba de seguridad de acceso

Fuente: El Autor

## **1.6 CAPACITACIÓN AL PERSONAL**

La fase de capacitación, es uno de los aspectos a tratar más importantes para el correcto manejo del sistema web, en este proceso se detallaron todos los aspectos requeridos para el uso de la herramienta, para lo cual, se requirió que el personal que asista a este curso tenga conocimientos básicos de informática.

El personal total que recibió la capacitación constó de 13 amanuenses, 1 administrador y 2 rancheros.

## **1.7 MANTENIMIENTO DEL SISTEMA WEB**

El mantenimiento del sistema web, es vital para contar con una herramienta actualizada y libre de fallos por lo cual se requiere un encargado con conocimientos técnicos en el área de sistemas que pueda dar soporte continuo cuando así se requiera y evitar retrasos innecesarios, por ello el proponente de la investigación está dispuesto a cumplir con esta tarea.

Un ejemplo del mantenimiento requerido es el realizar un respaldo o copias de seguridad de forma mensual con la finalidad de salvaguardar la información almacenada en caso de fallos que pudieran suscitarse.

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

## **1.8 CONCLUSIONES**

- El uso de herramientas de software libre favorece a la realización de sistemas o proyectos web por su bajo costo y la amplia disponibilidad de documentación.
- El empleo de Netbeans en el entorno de desarrollo facilitó la Programación PHP Orientado a Objetos ya que es una herramienta que facilita la implementación de sistemas web por su facilidad de uso, es multiplataforma, multilenguaje y presenta bajos costos. Así también WampServer como servidor integrado de páginas web y MySql para el almacenamiento y respaldo de los datos.
  - El diseño del sistema web facilita el acceso de los usuarios al sistema sin utilizar programas específicos sino simplemente con el uso de un navegador web que respalda las necesidades y requerimientos que posee la Caballería N°11 Galápagos con relación al control de las comandas.
  - La implementación del sistema web de registro para el proceso de registro y control del servicio de alimentación en la Brigada De Caballería Blindada N°11 “Galápagos tiene un funcionamiento óptimo y se está empelando por los usuarios pertinentes.

## **1.9 RECOMENDACIONES**

- El sistema web siempre debe estar actualizado y hacer uso de las herramientas de software libre para ahorrar costos y facilita la implementación de futuras aplicaciones.
- Antes de utilizar cualquier herramienta web para la elaboración de aplicaciones se recomienda la verificación de todas las ventajas y desventajas que estas posee para así generar proyectos eficientes.

- El proceso de diseño es aquel que determina si el grado de cumplimiento de los requerimientos solicitados por los usuarios pueden llegar a cumplirse por lo es indispensable cumplir con este proceso en cualquier tipo de proyecto web
- Un sistema web es un mecanismo de control, por lo que se podría indagar en otras áreas que requieran este servicio dentro de la institución.

## BIBLIOGRAFÍA

- Alvarez, M. A. (2 de Enero de 2014). *Qué es MVC*. Obtenido de <https://desarrolloweb.com/articulos/que-es-mvc.html>
- Avila, J. (18 de Mayo de 2016). *Objetos: Ventajas y Desventajas*. Obtenido de <https://objetosweb.wordpress.com/2016/05/18/ventajas-y-desventajas-de-la-programacion-orientada-a-objetos/>
- Baez, S. (20 de Octubre de 2012). *Sistemas Web. ¿Para qué sirven?* Recuperado el 04 de Octubre de 2017, de <http://www.knowdo.org/knowledge/39-sistemas-web>
- Cardenas, D. (1 de Julio de 2016). *Ventajas y Desventajas de Programar en PHP*. Obtenido de <http://www.programandolaweb.info/posts/php/6/Ventajas-y-desventajas-en-el-uso-de-php.html>
- Cobo, A. (2005). *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. 6: Ediciones Diaz de Santos.
- Comando Conjunto de las Fuerzas Armadas del Ecuador. (2016). *Misión, Visión, Valores Institucionales*. Recuperado el 15 de Octubre de 2017, de <https://www.ccffaa.mil.ec/mision-vision-valores-institucionales/>
- Comunidad NetBeans. 2014. *Información NetBeans IDE 6.1*. Obtenido de [https://netbeans.org/community/releases/61/index\\_es.html](https://netbeans.org/community/releases/61/index_es.html)
- Conociendo. 2015. Obtenido de <https://programacionmovilandroid.wordpress.com/netbeans/>
- Cowburn, P. (1 de Diciembre de 2009). *PHP MANUAL*. Recuperado el 2017, de <http://php.net/manual/es/language.operators.php>
- Daniel. (21 de Mayo de 2012). *Bienvenidos a MySQL*. Obtenido de <https://mysqldaniel.wordpress.com/>

- Delgado Heber, S. (30 de Septiembre de 2014). *Concepto, características, ventajas y desventajas de MySQL y Workbench*. Obtenido de <https://prezi.com/923yydsinkww/concepto-caracteristicas-ventajas-y-desventajas-de-mysql-y-workbench/>
- Eder, C. (26 de Mayo de 2016). *Herramienta de Desarrollo del Software*. Obtenido de <https://okhosting.com/blog/herramientas-de-desarrollo-de-software/>
- Escobar Saucedo, A. (27 de Noviembre de 2015). *NeatBeans*. Obtenido de <http://laprogramacionentuidiaria.blogspot.com/2015/11/netbeans.html>
- Fumás, E. (11 de Junio de 2014). *Apache HTTP Server: ¿Qué es, cómo funciona y para qué sirve?* Recuperado el 23 de Octubre de 2017, de <http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>
- Garcia, F. (25 de Enero de 2013). *Entorno de Desarrollo Integrado (IDE)*. Obtenido de <https://fergarciaac.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>
- Gonzalez, A. (21 de Julio de 2008). *Programacion Orientada a Objetos*. Recuperado el 2017, de <https://algonzalezpoo.wordpress.com/caracteristicas-de-poo/>
- Hidalgo, H. (04 de Marzo de 2014). *Fases para el desarrollo de un proyecto Web*. Recuperado el 7 de Octubre de 2017, de <http://disenowebakus.net/fases-para-el-desarrollo-de-un-proyecto-web.php>
- Lagunez, C., & Abarca, N. (1 de Septiembre de 2016). *Programacion Orientada a Objetos PPT*. Obtenido de <https://cesarlagg.files.wordpress.com/2016/09/programacion-estructurada1.pptx>
- Lara, E. (13 de Julio de 2010). *Protocolo HTTP y Servidores Web*. Recuperado el 23 de Octubre de 2017, de

- <http://elara.site.ac.upc.edu/documentacion/INTERNET%20-%20UD8%20-%20Protocolo%20HTTP%20y%20servidores%20WEB.pdf>
- Lecaros, M. (17 de Febrero de 2010). *Rasmus Lerdorf: Ingeniero estrella y padre del PHP*. Obtenido de <http://www.maestrosdelweb.com/rasmus-lerdorf-php/>
- Ledesma, A. 2013. *¿Que es WampServer*. Obtenido de [https://prezi.com/h0j-\\_95vhkj2/que-es-wampserver/](https://prezi.com/h0j-_95vhkj2/que-es-wampserver/)
- MySQL. (13 de Abril de 2010). *Estructuras de tabla y de índice*. Obtenido de <http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/table-and-index.html>
- Pairuna, L. (11 de Mayo de 2017). *¿Qué es y para que sirve un sitio web?* Recuperado el 26 de Octubre de 2017, de <http://www.codedimension.com.ar/noticias-sobre-tecnologia/noticias/-que-es-y-para-que-sirve-un-sitio-web-/1>
- Pavón, J. (2012). *Servidores Web – Apache*. Recuperado el 24 de Octubre de 2017, de <https://www.fdi.ucm.es/profesor/jpavon/web/31-ServidoresWeb-Apache.pdf>
- Peñafiel, M. (13 de Abril de 2013). *Aplicaciones Web*. Recuperado el 6 de Octubre de 2017, de <https://www.scribd.com/doc/136052164/APLICACIONES-WEB-pdf>
- Red Gráfica Latinoamérica. (2013). *Elementos del diseño web, estilos y tendencias*. Recuperado el 5 de Octubre de 2017, de <http://redgrafica.com/Elementos-del-diseno-web-estilos-y>
- UNET. (13 de Diciembre de 2013). *Programación Orientada a Objetos(POO)*. Recuperado el 2017, de <https://compu2poo.wordpress.com/>
- UNEWEB. (7 de Marzo de 2016). *MySql Procedimientos almacenados y funciones*. Obtenido de <https://tecnologiaenvivo.com/mysql-procedimientos-almacenados-y-funciones/>

Universidad de Alicante. (2017). *Modelo vista controlador (MVC)*. Obtenido de <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

Visual Studio. (2010). *Procedimientos de seguridad básicos para aplicaciones Web*. Recuperado el 6 de Octubre de 2017, de [https://msdn.microsoft.com/es-es/library/zdh19h94\(v=vs.100\).aspx#cpconbestsecuritypracticesforwebapplicationsanchor3](https://msdn.microsoft.com/es-es/library/zdh19h94(v=vs.100).aspx#cpconbestsecuritypracticesforwebapplicationsanchor3)

Word, P. 2017. *Aplicaciones Móviles*. Obtenido de <https://appmundomovil.wordpress.com/2-3-entorno-de-desarrollo/>

# **ANEXOS**

# **ANEXO I**

## **MANUAL DEL USUARIO**

# MANUAL DE USUARIO DEL SISTEMA DE GESTIÓN DE CONFRONTAS Y CONTROL DEL SERVICIO DE ALIMENTACIÓN DE LA 11BCB “GALÁPAGOS”

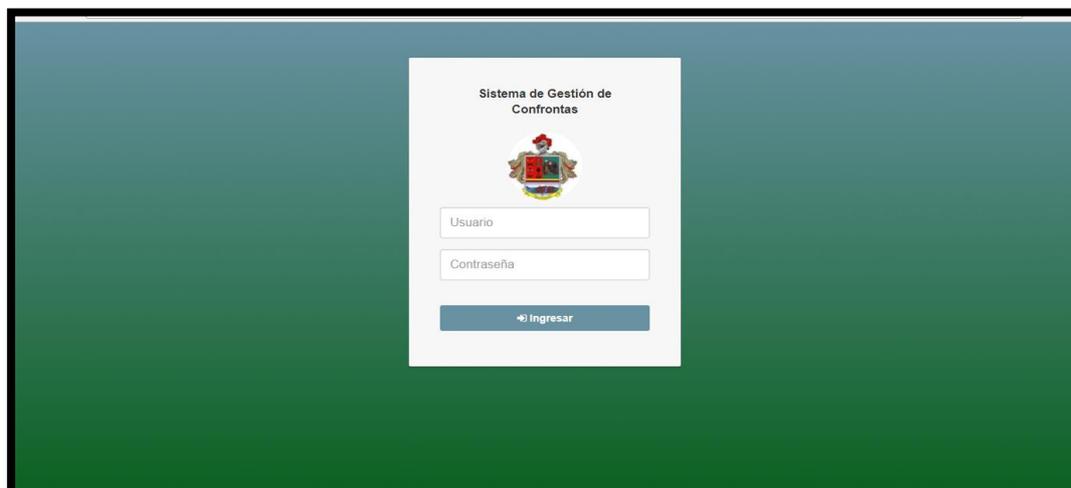
El sistema de gestión de confrontas y control del servicio de alimentación tiene como finalidad la modernización de los procesos realizados manualmente en la actualidad para lo cual es importante enfocarse en el desarrollo y modernización de sistemas informáticos que nos permitan digitalizar estas actividades de manera ágil y eficiente.

## JUSTIFICACIÓN

Al implementar el presente proyecto se pretende digitalizar el registro de las confrontas así como obtener un reporte detallado y de esta manera establecer un control del consumo del servicio de alimentación por parte del sr. Oficial ranchero, de la misma manera obtener un detalle de lo consumido por parte de los usuarios quienes podrán verificar el consumo y registro generado por el amanuense de su unidad durante el mes, de esta manera establecer una mejor administración y un control óptimo y moderno de este servicio.

## ESTRUCTURA DEL SISTEMA

### 1.- página de inicio



El presente sistema cuenta con tres tipos de usuarios quienes tendrán acceso al presente sistema y de acuerdo a las diferentes funciones que en cada uno de los usuarios cumple y los diferentes procesos que cada tipo de usuario nos brinda.

1.- usuario administrador

2.- usuario amanuense

3.- usuario ranchero

Para ingresar a cada uno de estos usuarios tenemos que autenticar nuestro usuario y contraseña asignado por el administrador del sistema.



The image shows a login interface for the 'Sistema de Gestión de Confrontas'. At the top, the system name is displayed above a national coat of arms. Below this, there are two input fields: the first contains the username 'admin' and the second contains a masked password '.....'. A green button with a right-pointing arrow and the text 'Ingresar' is positioned below the password field.

## USUARIO ADMINISTRADOR



El usuario administrador posee las siguientes opciones:

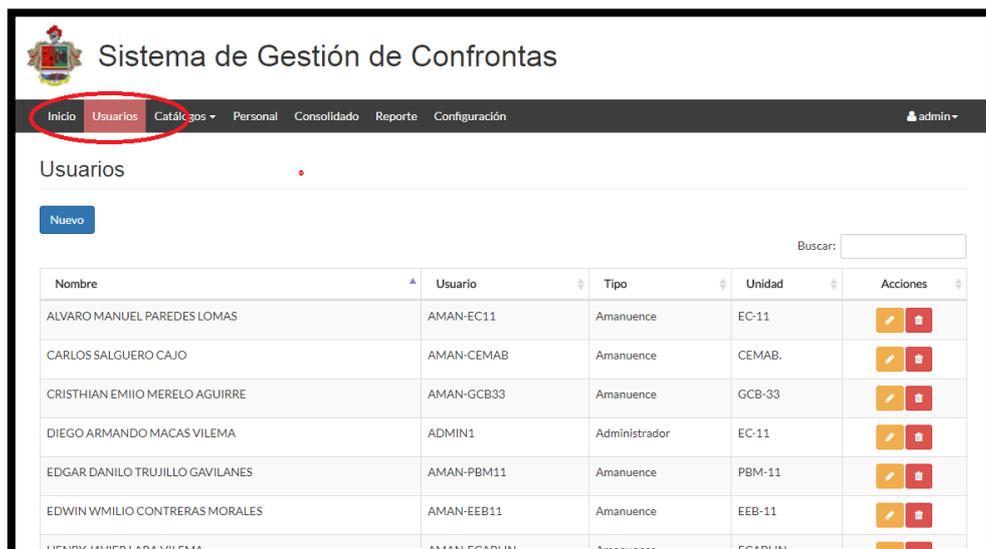
## INICIO

Da la bienvenida al sistema de gestión de confrontas, así como nos muestra el catálogo de opciones que tenemos para interactuar y navegar en el usuario.



## OPCIÓN USUARIO

Esta opción nos permite administrar los usuarios que manipularan el presente sistema, de entre sus principales funciones son: crear, editar, eliminar usuarios, cambio de claves entre otros.



The screenshot shows the 'Usuarios' management page. The navigation bar is highlighted with a red circle, showing 'Inicio', 'Usuarios', 'Catálogos', 'Personal', 'Consolidado', 'Reporte', and 'Configuración'. The page title is 'Usuarios'. There is a 'Nuevo' button and a search box labeled 'Buscar:'. Below is a table with columns: Nombre, Usuario, Tipo, Unidad, and Acciones. The table lists several users with their respective details and action icons (edit and delete).

Nombre	Usuario	Tipo	Unidad	Acciones
ALVARO MANUEL PAREDES LOMAS	AMAN-EC11	Amanuense	EC-11	 
CARLOS SALGUERO CAJO	AMAN-CEMAB	Amanuense	CEMAB.	 
CRISTHIAN EMIO MERELO AGUIRRE	AMAN-GCB33	Amanuense	GCB-33	 
DIEGO ARMANDO MACAS VILEMA	ADMIN1	Administrador	EC-11	 
EDGAR DANILO TRUJILLO GAVILANES	AMAN-PBM11	Amanuense	PBM-11	 
EDWIN WMILIO CONTRERAS MORALES	AMAN-EEB11	Amanuense	EEB-11	 
HENRY JAVIER LARA VILEMA	AMAN-ECARUN	Amanuense	ECARUN	 

Para el desarrollo del presente proyecto este sistema contará con un usuario administrador, un usuario amanuense por cada una de las unidades que tengan bajo su

responsabilidad el registro de las confrontas, y un usuario ranchero quien realizará como misión fundamental el control del registro y creación de extra confrontas de ser necesario.

## OPCIÓN CATALOGO

Esta opción nos brinda un menú desplegable con las opciones:

- Unidades
- Tipo de personal
- Tipo de novedades
- Grados del personal
- Armas/servicios/especialidades

Opciones disponibles en un menú desplegable al momento de crear nuevos usuarios y otras funciones en el sistema.



ID	Nombre	Abreviatura	Conscriptos	Hora Almuerzo	Acciones
1	CEM-11	CEM-11	0	12:00 - 14:00	[+][-][x]
3	GCB-31	GCB-31	54	12:00 - 14:00	[+][-][x]
4	ECABLIN	ECABLIN	25	12:00 - 14:00	[+][-][x]
5	GCB-32	GCB-32	0	12:00 - 14:00	[+][-][x]
7	GCB-33	GCB-33	0	12:00 - 14:00	[+][-][x]
9	GAAP-11	GAAP-11	0	12:00 - 14:00	[+][-][x]
	GAAP-11	GAAP-11	0	12:00 - 14:00	[+][-][x]

## OPCIÓN PERSONAL

Esta opción nos brinda la facilidad de visualizar a todo el personal que ha sido registrado en cada una de las unidades por sus respectivos amanuenses, de la misma manera el administrado tiene la facilidad de registrar más usuarios en las diferentes

unidades, editar, eliminar crear un código de barras para el usuario y visualizar el consumo individual de los usuarios de cada una de las unidades.

Id	Identificación	Grado	Arma	Apellidos	Nombres	Unidad	Acciones
1	0603264474	CAPT	COM.	INCA CHUNATA	MARCO ELEICIO	EC-11	[Edit] [Delete] [Generate]
2	1718240573	CAPT	COM.	SANCHEZ CARDENAS	VICTOR ORLANDO	EC-11	[Generate Tarjeta] [Edit] [Delete]
3	0602248031	SUBP	COM	MERINO CHAVEZ	JUAN ANIBAL	EC-11	[Edit] [Delete] [Generate]
4	1101751343	SUBP	COM.	JIMENEZ RIOS	PABLO ADALBERTO	EC-11	[Edit] [Delete] [Generate]
5	0602468637	SUBP	I.	NOGALES PINDUISACA	RODRIGO MANUEL	EC-11	[Edit] [Delete] [Generate]
6	0602536377	SGOP	COM	CHAVEZ	LUIS ANGEL	EC-11	[Edit] [Delete] [Generate]
				TIERRA LEMA	JAIME BENITO	EC-11	[Edit] [Delete] [Generate]

## USUARIO CONSOLIDADO

Esta opción nos permite visualizar el consolidado de las unidades que ya generaron las confrontas así como tener un control de las unidades que no realizaron el registro de la confronta que está controlado por un horario establecido para esta actividad.

Unidad	Desayunos				Almuerzos				Meriendas				Acciones
	OFI	VOL	CON	Total	OFI	VOL	CON	Total	OFI	VOL	CON	Total	
GCB-32	12	127	0	139	17	127	0	144	12	127	0	139	[Edit] [Delete] [Generate]
EEB-11	1	67	0	68	1	67	0	68	1	67	0	68	[Edit] [Delete] [Generate]
CAL-11	8	133	0	141	8	135	0	143	8	133	0	141	[Edit] [Delete] [Generate]
GCB-31	3	98	0	101	17	120	0	137	3	96	0	99	[Edit] [Delete] [Generate]
ERS-11	0	93	0	93	3	106	0	109	0	96	0	96	[Edit] [Delete] [Generate]
EPM-11	2	37	0	39	2	37	0	39	2	37	0	39	[Edit] [Delete] [Generate]
GAAA 4/5	0	6	0	6	2	22	0	24	1	9	0	10	[Edit] [Delete] [Generate]
GAAP-11	1	67	45	113	16	121	45	182	1	71	45	117	[Edit] [Delete] [Generate]
EC-11	2	33	5	40	2	58	5	65	2	49	5	56	[Edit] [Delete] [Generate]
Total	29	661	50	740	68	793	50	911	30	685	50	765	

Dentro de las opciones que brinda esta podremos ver el resumen de la confronta creada para cada una de las unidades.

Unidad	Desayunos				Almuerzos				Meriendas			
	OFI	VOL	CON	Total	OFI	VOL	CON	Total	OFI	VOL	CON	Total
GCB-32	12	127	0	139	17	127	0	144	12	127	0	139
EEB-11												
CAL-11												
GCB-31												
ERS-11												
EPM-11												
GAAA 4/5												

Otra opción de esta herramienta es el verificar el listado de toda la unidad con el detalle del registro de la confronta en cada una de las unidades.

Identificación	Grado	Arma	Nombres	DES	ALM	MER
1801842962	TCRN	EM.	DARWIN RODOLFO MONTOYA FREIRE	✓	✓	✓
1304310301	MAYO	C.B	WILFRIDO VIRGILIO ARIAS PICO	✗	✓	✗
1712865573	CAPT	C.B	FREDDY XAVIER GALLO COBOS	✗	✓	✗
1718827320	TNTE	C.B	CARLOS ALBERTO GARZON GONZALEZ	✗	✓	✗
0604075457	TNTE	COM	SILVANA EDITH BARRAGAN GUILLEN	✗	✓	✗
1717990665	TNTE	C.B	MILTON PATRICIO ARG?ELLO SOLA	✓	✓	✓
1720572294	TNTE	C.B	OLIVER STALIN C?RDENAS ALB?N	✓	✓	✓
1718923574	SUBT	C.B	GINSON GUILLERMO PES?NTEZ OCHOA	✗	✓	✗
2100396569	SUBT	C.B	PABLO ANDRES PINTO VALLEJO	✓	✓	✓
1722632161	SUBT	COM	HUGO XAVIER JACOME LOPEZ	✓	✓	✓

El ultimo icono nos permite ver los extra confrontas creados por el usuario ranchero que será quien en caso de no estar registrado procederá a generar los extra-confrontas.

Inicio Usuarios Catálogos Personal Consolidado Reporte Configuración

localhost/comedor/Confronta/verExtraconfronta/15&valor=5 - Google Chrome

localhost/comedor/Confronta/verExtraconfronta/15&valor=5

Confronta

Día Consolidado: 2017-11-24

**EXTRACONFRONTA DE RANCHO DE GCB-32**  
Para el día Viernes 24 de Noviembre del 2017

Unidad	Desayunos	Almuerzos	Meriendas	Total
GCB-32	0	0	0	0

Unidad: GCB-32, EEB-11, CAL-11, GCB-31, ERS-11, EPM-11, GAAA 4/5

### Confronta general

Sistema de Gestión de Confrontas

Inicio Usuarios Catálogos Personal Consolidado Reporte Configuración

Confrontas

Día Consolidado: 2018-01-01 [Generar](#)

Unidad	Desayunos				Almuerzos				Meriendas				Acciones
	OFI	VOL	CON	Total	OFI	VOL	CON	Total	OFI	VOL	CON	Total	
EC-11	2	58	28	88	2	56	28	86	2	61	28	91	<a href="#">Ver</a> <a href="#">Imprimir</a>
GCB-31	0	0	54	54	0	2	54	56	0	1	54	55	<a href="#">Ver</a> <a href="#">Imprimir</a>
ECABLIN	0	1	0	1	0	2	0	2	0	0	0	0	<a href="#">Ver</a> <a href="#">Imprimir</a>
<b>Total</b>	<b>2</b>	<b>59</b>	<b>82</b>	<b>143</b>	<b>2</b>	<b>60</b>	<b>82</b>	<b>144</b>	<b>2</b>	<b>62</b>	<b>82</b>	<b>146</b>	

### Resumen confronta unidad

localhost/comedor/Confronta/verGeneral/11 - Google Chrome

localhost/comedor/Confronta/verGeneral/11

**CONFRONTA DE RANCHO DE EC-11**  
Para el día Lunes 01 de Enero del 2018

Unidad	Desayunos				Almuerzos				Meriendas			
	OFI	VOL	CON	Total	OFI	VOL	CON	Total	OFI	VOL	CON	Total
EC-11	2	58	28	88	2	56	28	86	2	61	28	91

## Listado del personal quienes fueron registrados.

Identificación	Grado	Arma	Nombres	DES	ALM	MER
0603264474	CAPT	COM.	INCA CHUNATA MARCO ELEICIO	✓	✓	✓
1718240573	CAPT	COM.	SANCHEZ CARDENAS VICTOR ORLANDO	✓	✓	✓
0602248031	SUBP	COM.	MERINO CHAVEZ JUAN ANIBAL	✓	✓	✓
1101751343	SUBP	COM.	JIMENEZ RIOS PABLO ADALBERTO	✓	✓	✓
0602468637	SUBP	I.	NOGALES PINDUISACA RODRIGO MANUEL	✓	✓	✓
0602536377	SGOP	COM	CHAVEZ LUIS ANGEL	✓	✓	✓
1001491131	SGOP	COM	TIERRA LEMA JAIME BENITO	✓	✓	✓
0602485971	SGOP	COM	SEGURA RODRIGUEZ JORGE SANDRO	✓	✓	✓
1802648566	SGOP	COM.	IBARRA FALCONI WILLIAM ROBERTO	✓	✓	✓
1802803575	SGOP	COM	RUIZ GARCES WILLIAM FERNANDO	✓	✓	✓

## OPCIÓN REPORTE

Esta función nos permitirá generar el consolidado del mes de cada una de las unidades así como el valor a pagar en cada una de ellas y el detalle por días del consumo del servicio de alimentación.

Sistema de Gestión de Confrontas															
Inicio Usuarios Catálogos Personal Consolidado <b>Reporte</b> Configuración admin															
Confrontas															
Mes Consolidado															
2018-01															Generar
Imprimir															
CONFRONTA CONSOLIDADA DEL MES DE ENERO del 2018															
Unidad	Desayunos					Almuerzos					Meriendas				
	OFI	VOL	CON	Total	Valor	OFI	VOL	CON	Total	Valor	OFI	VOL	CON	Total	Valor
GCB-31	0	0	54	54	32.40	0	2	54	56	84.00	0	1	54	55	49.50
ECABLIN	0	1	0	1	0.60	0	2	0	2	3.00	0	0	0	0	0.00
EC-11	4	107	56	167	100.20	3	107	56	166	249.00	4	116	56	176	158.40
<b>Total</b>	<b>4</b>	<b>108</b>	<b>110</b>	<b>222</b>	<b>133.20</b>	<b>3</b>	<b>111</b>	<b>110</b>	<b>224</b>	<b>336.00</b>	<b>4</b>	<b>117</b>	<b>110</b>	<b>231</b>	<b>207.90</b>

Sistema de Gestión de Confrontas																
Inicio Usuarios Catálogos Personal Consolidado <b>Reporte</b> Configuración admin																
Confronta																
Mes Consolidado																
2017-11															Generar	
Imprimir																
CONFRONTA DE RANCHO DE EC-11																
Para Noviembre del 2017																
Unidad	C	Desayunos					Almuerzos					Meriendas				
		OFI	VOL	CON	Total	Valor	OFI	VOL	CON	Total	Valor	OFI	VOL	CON	Total	Valor
GCB-31		0	0	54	54	32.40	0	2	54	56	84.00	0	1	54	55	49.50
GCB-32		0	1	0	1	0.60	0	2	0	2	3.00	0	0	0	0	0.00
GAAP-11		4	107	56	167	100.20	3	107	56	166	249.00	4	116	56	176	158.40
GAAA 4/5		0	0	0	0	0.00	0	0	0	0.00	0	0	0	0	0.00	
CAL-11		0	0	0	0	0.00	0	0	0	0.00	0	0	0	0	0.00	
ERS-11		0	0	0	0	0.00	0	0	0	0.00	0	0	0	0	0.00	
EC-11		4	107	56	167	100.20	3	107	56	166	249.00	4	116	56	176	158.40
<b>Total</b>		<b>4</b>	<b>108</b>	<b>110</b>	<b>222</b>	<b>133.20</b>	<b>3</b>	<b>111</b>	<b>110</b>	<b>224</b>	<b>336.00</b>	<b>4</b>	<b>117</b>	<b>110</b>	<b>231</b>	<b>207.90</b>

## OPCIÓN CONFIGURACIÓN

Esta opción nos brinda la facilidad de establecer los parámetros de configuración del sistema tales como:

- Horario de envío de las confrontas
- Costo del servicio

Sistema de Gestión de Confrontas

Inicio Usuarios Catálogos Personal Consolidado Reporte Configuración admin

Parametros de Configuración

Hora Max. de envío Confronta	<input type="text" value="17:05"/>
Precio Desayuno	<input type="text" value="0.60"/>
Precio Almuerzo	<input type="text" value="1.50"/>
Precio Merienda	<input type="text" value="0.90"/>

Guardar

Copyright ©2016

## USUARIO AMANUENSE

### OPCIÓN INICIO

Nos da la bienvenida al sistema de gestión de confrontas, así como nos muestra el menú de herramientas y opciones que tiene el usuario, este tipo de usuarios para el desarrollo del presente sistema será asignado a cada uno de los amanuenses encargados del registro de las confrontas en las unidades.



## OPCIÓN PERSONAL

Esta opción nos muestra el listado del personal registrado en el sistema dentro de una determinada unidad, de este listado tiene control total el usuario amanuense que se encuentre encargado de la creación de las confrontas diarias de esta unidad.

Debemos tomar en cuenta que en la brigada existen 13 unidades y cada una de ellas tienen un usuario de tipo amanuense, quien cumplirá las funciones de administrador y encargado de la creación diaria de las confrontas en cada una de las unidades.

Id	Identificación	Grado	Arma	Apellidos	Nombres	Unidad	Acciones
1	0603264474	CAPT	COM.	INCA CHUNATA	MARCO ELEICIO	EC-11	[Edit] [Delete] [Add]
2	1718240573	CAPT	COM.	SANCHEZ CARDENAS	VICTOR ORLANDO	EC-11	[Edit] [Delete] [Add]
3	0602248031	SUBP	COM	MERINO CHAVEZ	JUAN ANIBAL	EC-11	[Edit] [Delete] [Add]
4	1101751343	SUBP	COM.	JIMENEZ RIOS	PABLO ADALBERTO	EC-11	[Edit] [Delete] [Add]
5	0602468637	SUBP	I.	NOGALES PINDUISACA	RODRIGO MANUEL	EC-11	[Edit] [Delete] [Add]
6	0602536377	SGOP	COM	CHAVEZ	LUIS ANGEL	EC-11	[Edit] [Delete] [Add]
7	1001491131	SGOP	COM	TIERRA LEMA	JAIME BENITO	EC-11	[Edit] [Delete] [Add]

Dentro de esta opción tenemos 4 herramientas que corresponde a la creación de un código de barras que servirá para la implementación de la segunda parte de este sistema.

La edición editar y eliminar que nos permitirá la edición de algún error en la creación de los usuarios en la unidad así como de la eliminación.

Y la última opción nos permite visualizar el consumo mensual que tiene cada uno de los usuarios y poder cancelar.

## OPCIÓN NOVEDAD

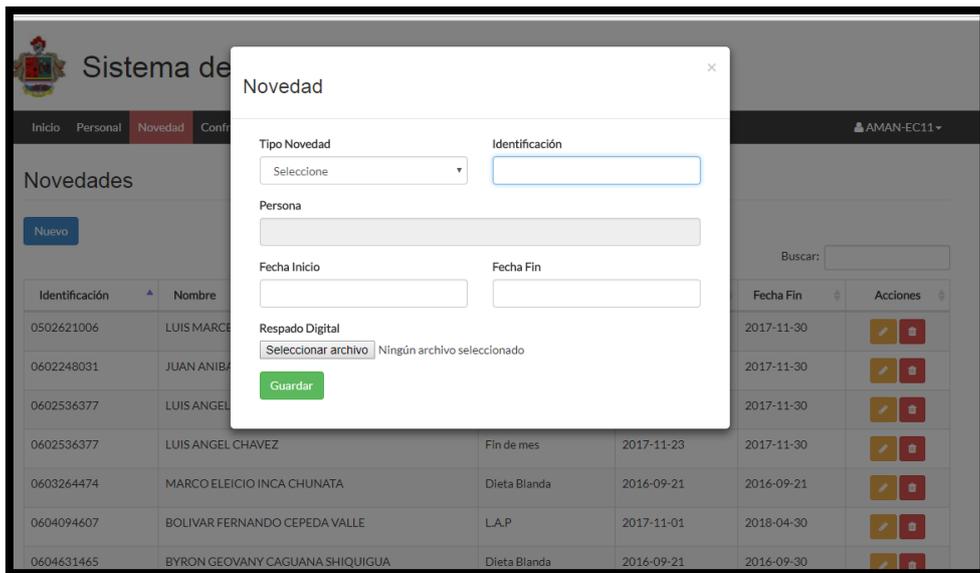
Esta herramienta nos permite registrar de una forma adecuada las novedades presentadas antes de crear la confronta, teniendo en cuenta que este paso se debe realizar antes de crear las confrontas.



Identificación	Nombre	Tipo Novedad	Fecha Inicio	Fecha Fin	Acciones
0502621006	LUIS MARCELO SUAREZ BUNSHI	L.A.P	2017-11-01	2017-11-30	 
0602248031	JUAN ANIBAL MERINO CHAVEZ	Antenas	2017-11-24	2017-11-30	 
0602536377	LUIS ANGEL CHAVEZ	Dieta Blanda	2017-11-24	2017-11-30	 
0602536377	LUIS ANGEL CHAVEZ	Fin de mes	2017-11-23	2017-11-30	 
0603264474	MARCO ELEICIO INCA CHUNATA	Dieta Blanda	2016-09-21	2016-09-21	 
0604094607	BOLIVAR FERNANDO CEPEDA VALLE	L.A.P	2017-11-01	2018-04-30	 
0604631465	BYRON GEOVANY CAGUANA SHIQUIGUA	Dieta Blanda	2016-09-21	2016-09-30	 

## NOVEDAD NUEVA

Nos permite registrar las novedades que se han presentado en cuanto al movimiento del personal teniendo en cuenta para la creación de la misma la fecha de inicio y fecha de finalización de la novedad así como adjuntar la documentación de descargo.



## OPCIÓN CONFRONTA

En esta herramienta se encuentran almacenadas o registradas las confrontas creadas por fechas de la unidad.

The image shows the 'Confrontas' section of the web application. The table displays the following data:

Unidad	Desayunos			Almuerzos			Meriendas			Fecha	Acciones
	OFI	VOL	CON	OFI	VOL	CON	OFI	VOL	CON		
EC-11	1	59	20	2	59	18	2	59	20	2016-09-27	[Edit] [Delete] [Add]
EC-11	1	62	28	2	61	28	2	60	28	2016-10-06	[Edit] [Delete] [Add]
EC-11	2	62	28	2	62	28	2	62	28	2017-05-11	[Edit] [Delete] [Add]
EC-11	2	58	28	1	56	28	2	60	28	2017-10-23	[Edit] [Delete] [Add]
EC-11	1	61	28	1	61	28	2	61	28	2017-10-24	[Edit] [Delete] [Add]
EC-11	0	5	28	2	62	28	0	8	28	2017-12-01	[Edit] [Delete] [Add]

## Nueva Confronta

Esta herramienta nos permitirá crear una nueva confronta previo al registro de las novedades y seleccionar la fecha a la que corresponde, de la misma manera está contemplado horarios en este herramienta.



Esta acción se realiza activando con un clic en el visto el servicio que se acceder y de no ser así una x.

Identificación	Grado	Arma	Nombres	DES	ALM	MER
0603264474	CAPT	COM.	INCA CHUNATA MARCO ELEICIO	✓	✓	✓
1718240573	CAPT	COM.	SANCHEZ CARDENAS VICTOR ORLANDO	✓	✓	✓
0602248031	SUBP	COM.	MERINO CHAVEZ JUAN ANIBAL	✓	✓	✓
1101751343	SUBP	COM.	JIMENEZ RIOS PABLO ADALBERTO	✓	✓	✓
0602468637	SUBP	I.	NOGALES PINDUISACA RODRIGO MANUEL	✓	✓	✓
0602536377	SGOP	COM.	CHAVEZ LUIS ANGEL			Fin de mes
1001491131	SGOP	COM.	TIERRA LEMA JAIME BENITO	✓	✓	✓
0602485971	SGOP	COM.	SEGURA RODRIGUEZ JORGE SANDRO	✓	✓	✓
1802648566	SGOP	COM.	IBARRA FALCONI WILLIAM ROBERTO			Disponibilidad
1802803575	SGOP	COM.	RUIZ GARCES WILLIAM FERNANDO	✓	✓	✓

Dentro de las confrontas creadas tenemos cuatro acciones que se pueden realizar editar, eliminar, ver el resumen de la unidad y ver el consolidado de todos los usuarios de esa unidad.

Unidad	Desayunos			Almuerzos			Meriendas			Total		
	OFI	VOL	CON	OFI	VOL	CON	OFI	VOL	CON			
EC-11	1	59	20	80	2	59	18	79	2	59	20	81

Identificación	Grado	Arma	Nombres	DES	ALM	MER
0603264474	CAPT	COM.	INCA CHUNATA MARCO ELEICIO	✓	✓	✓
0603264474	CAPT	COM.	INCA CHUNATA MARCO ELEICIO	✗	✓	✓
0602248031	SUBP	COM.	MERINO CHAVEZ JUAN ANIBAL	✓	✓	✗
1101751343	SUBP	COM.	JIMENEZ RIOS PABLO ADALBERTO	✓	✗	✓
0602468637	SUBP	I.	NOGALES PINDUISACA RODRIGO MANUEL	✓	✓	✓
0602536377	SGOP	COM.	CHAVEZ LUIS ANGEL	✓	✓	✓
1001491131	SGOP	COM.	TIERRA LEMA JAIME BENITO	✓	✓	✗
0602485971	SGOP	COM.	SEGURA RODRIGUEZ JORGE SANDRO	✓	✓	✗
1802648566	SGOP	COM.	IBARRA FALCONI WILLIAM ROBERTO	✓	✓	✓
1802803575	SGOP	COM.	RUIZ GARCES WILLIAM FERNANDO	✓	✓	✓

## OPCIÓN UNIDAD

En esta herramienta consta el numérico de conscriptos que dispone la unidad, de la misma manera parámetros de horarios del consumo de alimentación de esa unidad, cabe destacar que los horarios para acceder al servicio de alimentación hacen referencia más a los almuerzos.



## OPCIÓN CONSOLIDADO

Esta herramienta nos permite visualizar el reporte consolidado del día, creado por el usuario amanuense de la misma manera contamos con un menú que nos permite ver el resumen, ver el listado o detalle del registro de la confronta del día y el reporte de extra-confrontas creados ese día por el usuario rancharo.

Unidad	Desayunos				Almuerzos				Meriendas				Acciones
	OFI	VOL	CON	Total	OFI	VOL	CON	Total	OFI	VOL	CON	Total	
EC-11	2	33	5	40	2	58	5	65	2	49	5	56	
<b>Total</b>	<b>2</b>	<b>33</b>	<b>5</b>	<b>40</b>	<b>2</b>	<b>58</b>	<b>5</b>	<b>65</b>	<b>2</b>	<b>49</b>	<b>5</b>	<b>56</b>	

## Resumen

localhost/comedor/Confronta/verGeneral/24 Imprimir

**CONFRONTA DE RANCHO DE EC-11**  
Para el día Viernes 24 de Noviembre del 2017

Unidad	Desayunos				Almuerzos				Meriendas			
	OFI	VOL	CON	Total	OFI	VOL	CON	Total	OFI	VOL	CON	Total
EC-11	2	33	5	40	2	58	5	65	2	49	5	56

## Ver listado

**Confronta de Rancho de EC-11**  
Para el día Viernes 24 de Noviembre del 2017 Imprimir

Identificación	Grado	Arma	Nombres	DES	ALM	MER
0603264474	CAPT	COM.	INCA CHUNATA MARCO ELEICIO	✓	✓	✓
1718240573	CAPT	COM.	SANCHEZ CARDENAS VICTOR ORLANDO	✓	✓	✓
0602248031	SUBP	COM.	MERINO CHAVEZ JUAN ANIBAL	✓	✓	✓
1101751343	SUBP	COM.	JIMENEZ RIOS PABLO ADALBERTO	✓	✓	✓
0602468637	SUBP	I.	NOGALES PINDUISACA RODRIGO MANUEL	✓	✓	✓
0602936377	SGOP	COM.	CHAVEZ LUIS ANGEL			
1001491131	SGOP	COM.	TIERRA LEMA JAIME BENITO	✗	✓	✗
0602465971	SGOP	COM.	SEGURA RODRIGUEZ JORGE SANDRO	✗	✓	✗
1802648566	SGOP	COM.	IBARRA FALCONI WILLIAM ROBERTO	✗	✓	✗
1802803575	SGOP	COM.	RUIZ GARCES WILLIAM FERNANDO	✗	✓	✓

Fin de mes

## Ver extra confronta

localhost/comedor/Confronta/verExtraconfronta/24&valor=13 Imprimir

**EXTRACONFRONTA DE RANCHO DE EC-11**  
Para el día Viernes 24 de Noviembre del 2017

Unidad	Desayunos	Almuerzos	Meriendas	Total
EC-11	0	0	0	0

## OPCIÓN CONSUMOS

Esta herramienta nos permitirá obtener un reporte del consumo del servicio de alimentación en entre dos fechas determinadas.

Esta opción cuenta con dos opciones la primera de manera individual.

### Consumo Individual

Identificación: 0603264474 Persona: INCA CHUNATA MARCO ELEICIO

Desde: 2017-11-01 Hasta: 2017-11-30

Tabla de Consumo del Servicio de Confronta EC-11

Nombre	INCA CHUNATA MARCO ELEICIO	Identificación	0603264474
--------	----------------------------	----------------	------------

Mes Noviembre del 2017

	Cantidad	Total
Desayuno	2	1.20
Almuerzo	3	4.50
Merienda	3	2.70
<b>Total</b>		<b>8.4</b>

Extra Confronta

Mes	Servicio	Cantidad	Total
Noviembre 2017	Almuerzo	1	1.50

Y la opción consolidada del mes donde reflejara el listado de los usuarios y el valor a cancelar.

Tabla de Consumo del Servicio de Confronta EC-11  
Correspondiente al mes de Noviembre del 2017

Identificación	Nombres	Grado	Arma	Desayuno		Almuerzo		Merienda		Total
				Cant.	Total	Cant.	Total	Cant.	Total	
0603264474	CAPT	COM.	INCA CHUNATA MARCO ELEICIO	2	1.20	3	4.50	3	2.70	6.4
1718240573	CAPT	COM.	SANCHEZ CARDENAS VICTOR ORLANDO	3	1.80	3	4.50	2	1.80	8.1
0602248031	SUBP	COM.	MERINO CHAVEZ JUAN ANIBAL	2	1.20	3	4.50	2	1.80	7.5
1101751343	SUBP	COM.	JIMENEZ RIOS PABLO ADALBERTO	3	1.80	3	4.50	3	2.70	9
0602468637	SUBP	I.	NOGALES PINDUISACA RODRIGO MANUEL	3	1.80	3	4.50	3	2.70	9
0602336377	SGOP	COM.	CHAVEZ LUIS ANGEL	1	0.60	1	1.50	1	0.90	3
1001491131	SGOP	COM.	TIERRA LEMA JAJME BENITO	2	1.20	3	4.50	2	1.80	7.5
0602485971	SGOP	COM.	SEGURA RODRIGUEZ JORGE SANDRO	2	1.20	3	4.50	2	1.80	7.5
1802648566	SGOP	COM.	IBARRA FALCONI WILLIAM ROBERTO	1	0.60	2	3.00	1	0.90	4.5
1802803575	SGOP	COM.	RUIZ GARCES WILLIAM FERNANDO	2	1.20	3	4.50	3	2.70	8.4
0602462343	SGOP	COM.	QUISHPE QUISHPE MIGUEL OSWALDO	2	1.20	3	4.50	3	2.70	8.4
1712548021	SGOS	COM.	VALVERDE PACHECO LUIS REINALDO	2	1.20	3	4.50	3	2.70	8.4
1002520201	SGOS	COM.	CRIOLO BARAHONA EDUARDO GUILLERMO	2	1.20	3	4.50	3	2.70	8.4
1802987709	SGOS	COM.	SALGUERO ZUMBA ANGEL FERNANDO	2	1.20	3	4.50	2	1.80	7.5
0916365208	SGOS	COM.	BASTIDAS GONZALEZ WILLIAM WLADIMIR	2	1.20	3	4.50	2	1.80	7.5

## **USUARIO RANCHERO**

Este usuario esta creado con la finalidad de establecer un parámetro de control así como igual este tipo de usuario tendrá las ventajas de visualizar algunos parámetros necesarios para el control, posterior a la creación de las confrontas por parte de los usuarios amanuenses en cada una de las unidades, de la misma manera podrá visualizar quienes ya cumplieron con la creación de las confrontas de las unidades y quienes faltan.

De entre otras las detallaremos las funciones que dispone este tipo de usuarios:

### **Herramienta inicio**

Como en los otros tipos de usuarios nos da la bienvenida al sistema.



### **Herramienta personal**

Esta herramienta es la que nos permitirá el control del personal que acceda al servicio de alimentación, haya que una vez creada la confronta por parte de los usuarios amanuenses de cada una de las unidades con esta herramienta se determinara si cada uno de los usuarios están o no registrados en la confronta diaria.

Mediante un buscador podremos identificar al usuario y de no constar dentro de la confronta diaria.

### Herramienta extra confronta

Esta herramienta nos permitirá una vez verificado que un usuario no consta con en la confronta diaria, proceder a crear una extra-confronta a ese usuario y esto será almacenado y reflejara en el reporte mensual de este usuario.

Identificación	Nombre	Unidad	Servicio	Fecha	Acciones
0502621006	LUIS MARCELO SUAREZ BUNSHI	EC-11	Desayuno	2017-11-23	Editar Eliminar
0502621006	LUIS MARCELO SUAREZ BUNSHI	EC-11	Merienda	2017-11-23	Editar Eliminar
0603264474	MARCO ELEICIO INCA CHUNATA	EC-11	Almuerzo	2017-11-23	Editar Eliminar
0603264474	MARCO ELEICIO INCA CHUNATA	EC-11	Almuerzo	2016-10-06	Editar Eliminar
0603264474	MARCO ELEICIO INCA CHUNATA	EC-11	Almuerzo	2016-09-06	Editar Eliminar
0603264474	MARCO ELEICIO INCA CHUNATA	EC-11	Merienda	2016-10-06	Editar Eliminar
0604631465	BYRON GEOVANY CAGUANA SHIQUIGUA	EC-11	Desayuno	2016-10-06	Editar Eliminar
0916365208	WILLIAM WLADIMIR BASTIDAS GONZALEZ	EC-11	Desayuno	2017-11-23	Editar Eliminar

### Herramienta consolidado

Esta herramienta permitirá al este usuario visualizar la creación de las confrontas de las unidades ya que este usuario es encargado de realizar las adquisición con anterioridad para la preparación de los alimentos.

Unidad	Desayunos				Almuerzos				Meriendas				Acciones
	OFI	VOL	CON	Total	OFI	VOL	CON	Total	OFI	VOL	CON	Total	
GCB-32	12	127	0	139	17	127	0	144	12	127	0	139	Editar Eliminar
EEB-11	1	67	0	68	1	67	0	68	1	67	0	68	Editar Eliminar
CAL-11	8	133	0	141	8	135	0	143	8	133	0	141	Editar Eliminar
GCB-31	3	98	0	101	17	120	0	137	3	96	0	99	Editar Eliminar
ERS-11	0	93	0	93	3	106	0	109	0	96	0	96	Editar Eliminar
EPM-11	2	37	0	39	2	37	0	39	2	37	0	39	Editar Eliminar
GAAA 4/5	0	6	0	6	2	22	0	24	1	9	0	10	Editar Eliminar
GAAP-11	1	67	45	113	16	121	45	182	1	71	45	117	Editar Eliminar
EC-11	2	33	5	40	2	58	5	65	2	49	5	56	Editar Eliminar

## Herramienta reporte

Esta herramienta permite a este usuario visualizar y tener un control automático en referencia a las confrontas creadas diariamente y mensualmente, esta actividad esta automatizada posterior la creación de las confrontas en cada una de las unidades.

Inicio Personal Extra Confronta Consolidado **Reporte** ▲ rancharo ▼

### Confrontas

Mes Consolidado:  Generar Imprimir

CONFRONTA CONSOLIDADA DEL MES DE NOVIEMBRE del 2017															
Unidad	Desayunos				Almuerzos					Meriendas					
	OFI	VOL	CON	Total	Valor	OFI	VOL	CON	Total	Valor	OFI	VOL	CON	Total	Valor
GCB-31	3	98	0	101	60.60	17	120	0	137	205.50	3	96	0	99	89.10
GCB-32	12	127	0	139	83.40	17	127	0	144	216.00	12	127	0	139	125.10
GAAP-11	1	67	45	113	67.80	16	121	45	182	273.00	1	71	45	117	105.30
GAAA 4/5	0	6	0	6	3.60	2	22	0	24	36.00	1	9	0	10	9.00
CAL-11	8	133	0	141	84.60	8	135	0	143	214.50	8	133	0	141	126.90
ERS-11	0	93	0	93	55.80	3	106	0	109	163.50	0	96	0	96	86.40
EC-11	5	142	38	185	111.00	6	175	38	219	328.50	5	163	38	206	185.40
EEB-11	1	67	0	68	40.80	1	67	0	68	102.00	1	67	0	68	61.20
EPM-11	2	37	0	39	23.40	2	37	0	39	58.50	2	37	0	39	35.10
<b>Total</b>	<b>32</b>	<b>770</b>	<b>83</b>	<b>885</b>	<b>531.00</b>	<b>72</b>	<b>910</b>	<b>83</b>	<b>1065</b>	<b>1,597.50</b>	<b>33</b>	<b>799</b>	<b>83</b>	<b>915</b>	<b>823.50</b>

EXTRA CONFRONTA CONSOLIDADA DEL MES DE NOVIEMBRE del 2017						
Unidad	Desayunos		Almuerzos		Meriendas	
	Cantidad	Total	Cantidad	Total	Cantidad	Total
EC-11	3	1.80	1	1.50	1	0.90
<b>Total</b>	<b>3</b>	<b>1.80</b>	<b>1</b>	<b>1.50</b>	<b>1</b>	<b>0.90</b>

# **ANEXO II**

## **MANUAL DE INSTALACIÓN DEL SISTEMA WEB**

## **MYSQL WORKBENCH**

Permite el diseño de diagramas de Entidad-Relación para bases de datos MySQL, en esta herramienta se puede elaborar y visualizar: tablas, vistas, procedimientos almacenados y claves foráneas de dichas base.

Permite el análisis de la ingeniería inversa como directa, que exporta e importa la base de datos, para crear y guardar una copia de seguridad con MySQL

### **Ventajas y Desventajas de Workbench**

#### **Ventajas**

- Brinda libertad a los usuarios.
- Puede ser usado, copiado, estudiado, modificado y redistribuido.
- Ahorros multimillonarios en la adquisición de licencias
- Tiende a ser muy eficiente (porque mucha gente lo optimiza, mejora).

#### **Desventajas**

- El software libre y el software no-comercial son en realidad incompatibles con el software comercial.
- El software libre crea riesgos legales.
- El software libre no tiene garantía proveniente del autor.
- Disminuye el índice de software "pirata"

#### **¿Cómo Conseguir la Aplicación?**

Existen versiones para Window, Linux y Mac. Hay que tener en cuenta que para los dos últimos sistemas las versiones son Betas. Los links de descarga son los siguientes:

Windows y Linux:

<http://dev.mysql.com/downloads/workbench/5.1.html>

Mac:

<http://dev.mysql.com/workbench/?p=182>

### Instalación de MYSQL Workbench

Para poder realizar este proceso es importante tener el instalador de MySQL WorkBench.

- Ejecutamos el programa



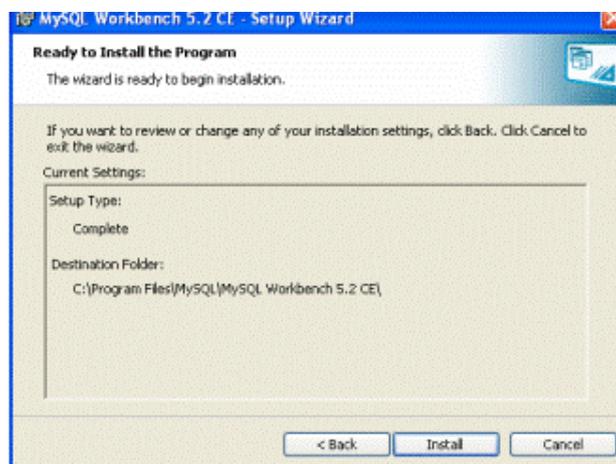
- Muestra una ventana de bienvenida. Para continuar con la instalación se da click en el botón Next.



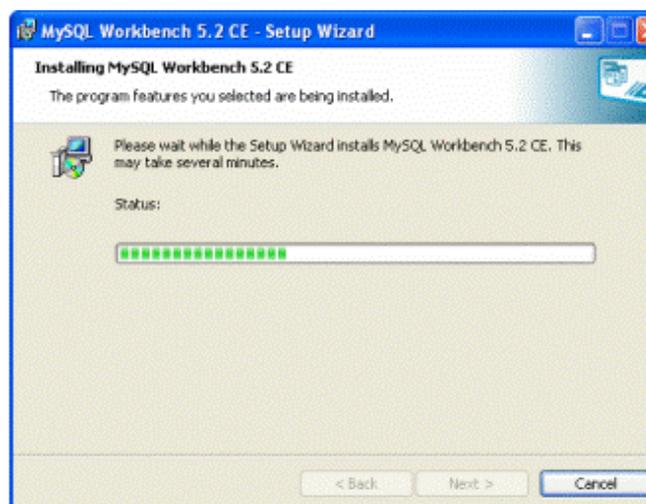
- El siguiente paso muestra la ruta de donde se instala el programa



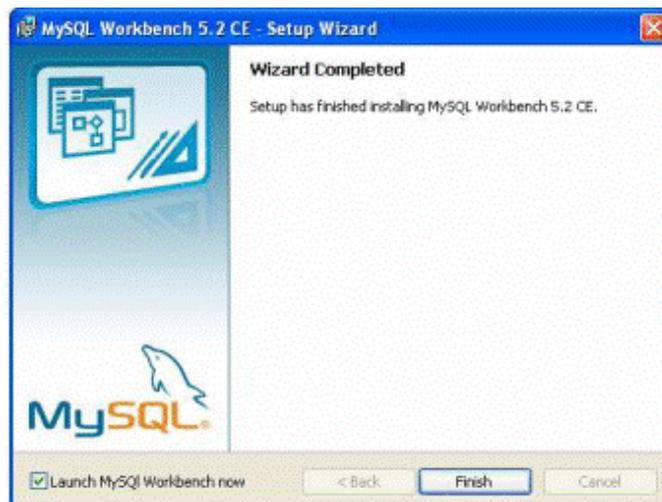
- Instalación del programa se procede presionando el botón "Install"



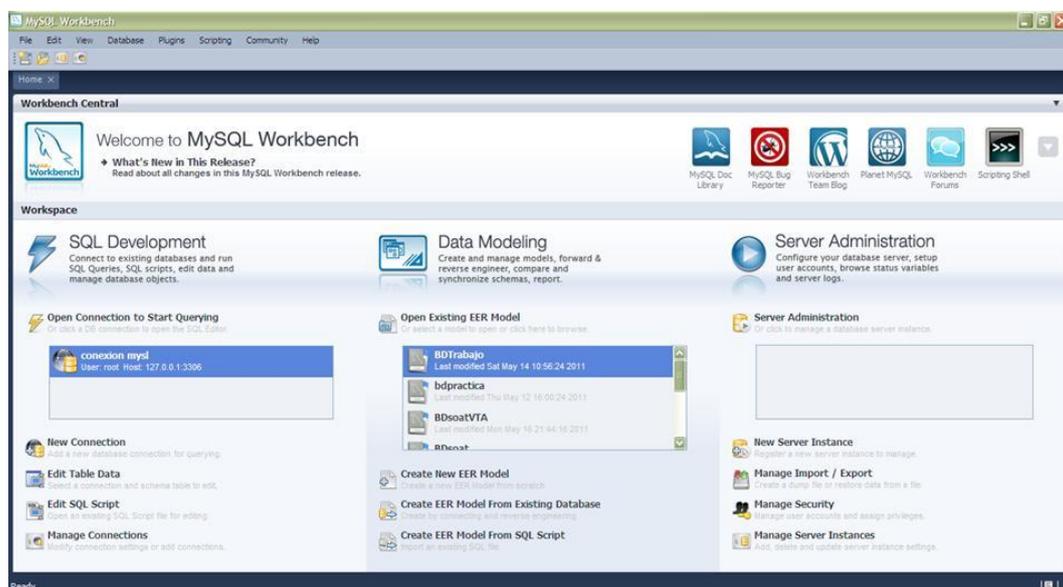
- Procesando instalación, y es importante esperar unos minutos,



- Una vez que la Instalación haya finalizada, hacer Click en el botón "Finish"



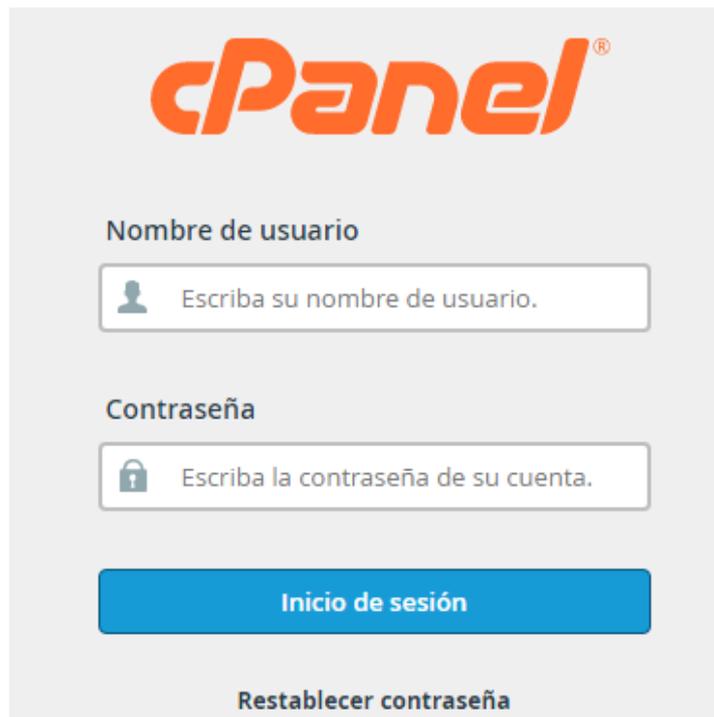
- Ventana de bienvenida



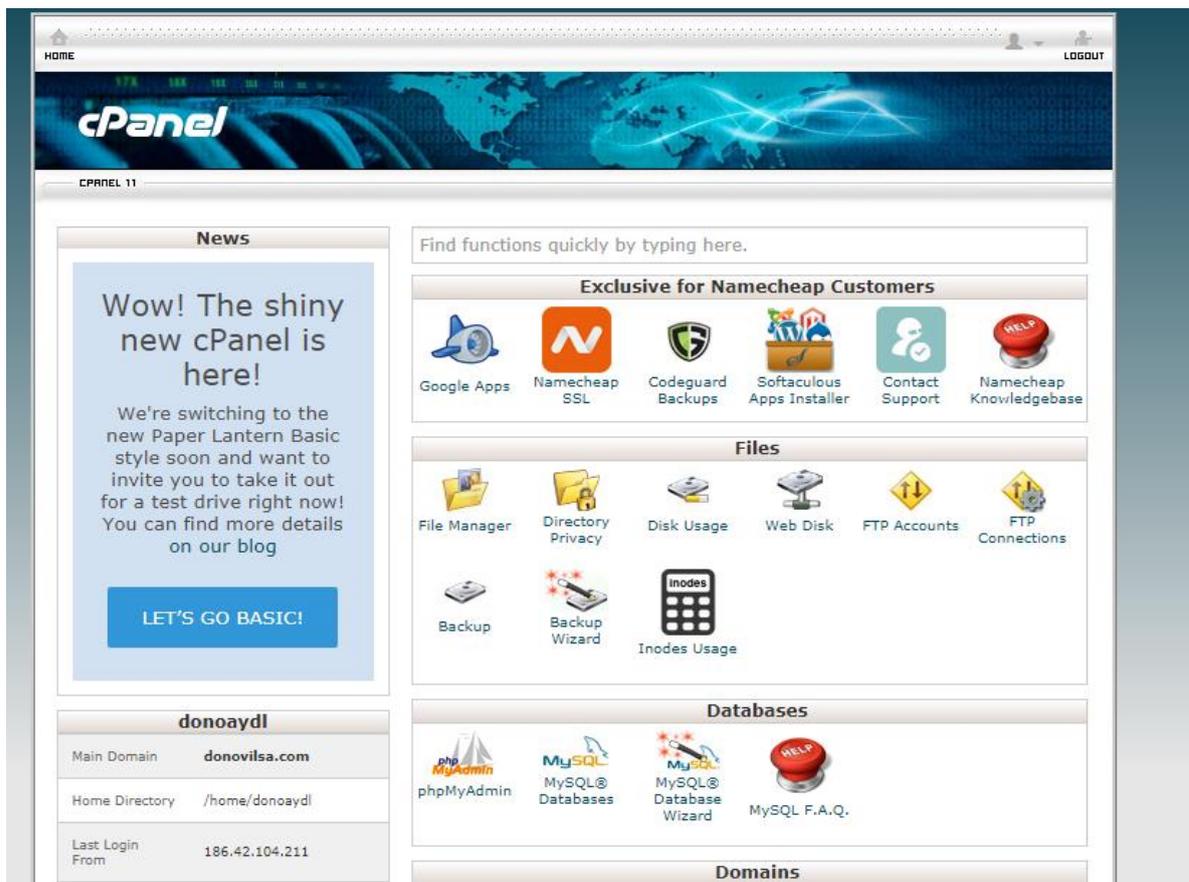
Para la publicación del sitio se debe ingresar a la siguiente dirección:

<http://donovilsa.com/cpanel>

Una vez que se haya ingresado a la dirección especificada, se presentará un formulario donde se debe ingresar el Nombre de Usuario y Contraseña establecida para el alojamiento del sitio moviliaria donovilsa

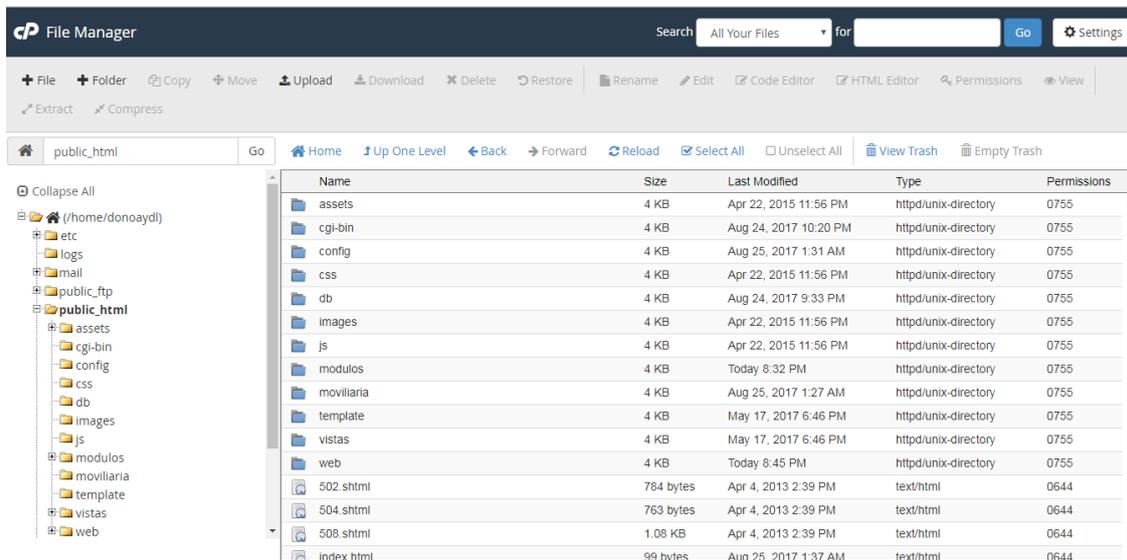


Con los datos ingresados correctamente se presentará una pantalla de inicio



A continuación se debe ingresar a la opción "*File Manager*".

Al presionar sobre la opción “*File Manager*” se presentará un formulario, aquí se puede subir y bajar archivos a la página a través del navegador web. Se puede borrar directorios enteros, archivos, etc., aunque es un poco complicado y lento de utilizar. En el caso de presentarse un proceso lento se recomienda utilizar FTP (FileZilla).



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

Select the file you want to upload to “/home/donoaydl”.

Maximum file size allowed for upload: 18.82 GB

Overwrite existing files

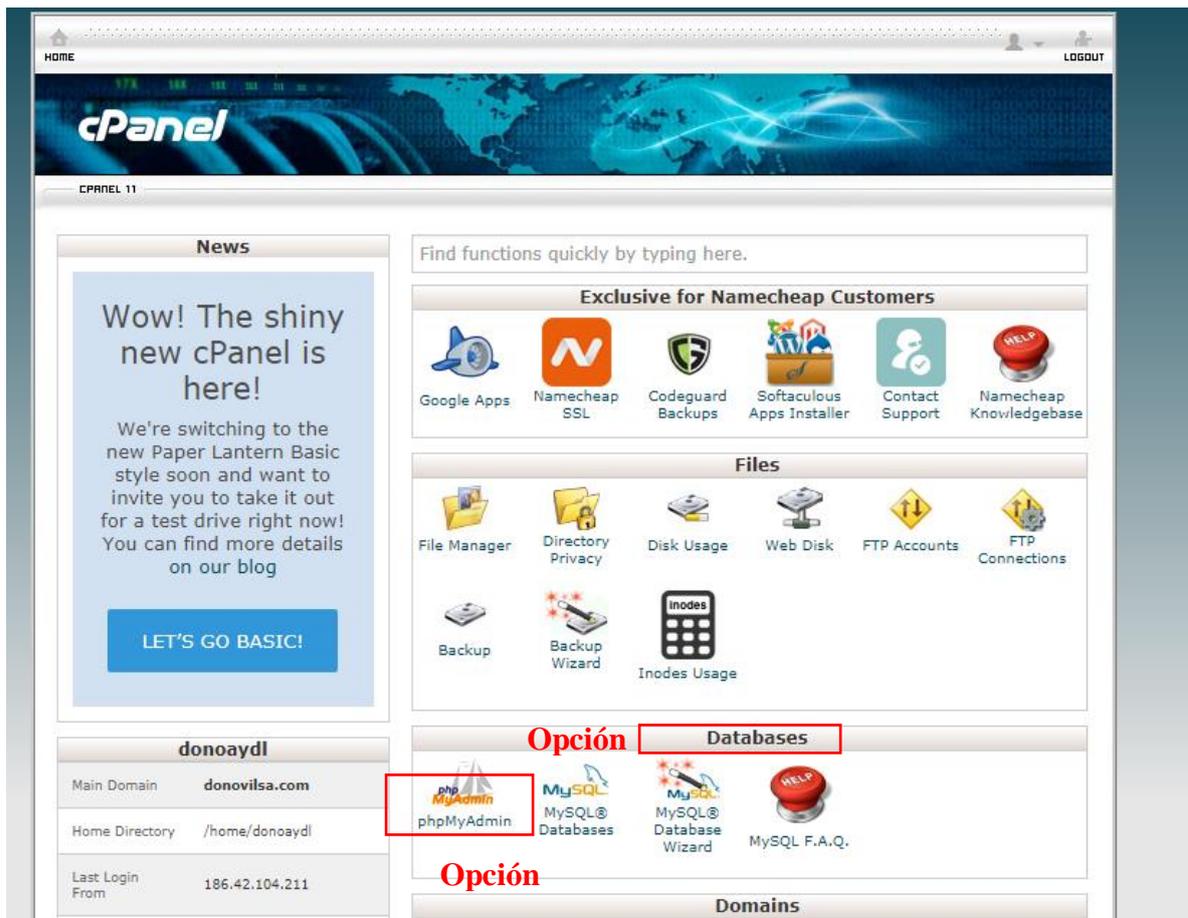
Drop files here to start uploading

or

[Select File](#)

## RESTAURACIÓN DE LA BASE DE DATOS

Para realizar la restauración de la base de datos del sitio web, se debe ingresar a la opción “*Databases*” y presionar en la opción “*phpMyAdmin*”



Al presionar sobre la opción “*phpMyAdmin*” se presentará el formulario donde se debe seleccionar el archivo que se requiere restaurar con la opción “*Seleccionar archivo*”.

Una vez que el archivo sea el correcto, se debe presionar sobre la opción “*Importar*” y presionar sobre el botón “*Continuar*” que se encuentra en la parte inferior de la pantalla.

Con este proceso se habrá realizado la restauración de la base de datos de forma correcta y podrá visualizarse la misma en la parte izquierda del formulario de phpMyAdmin.

Bases de datos SQL Estado actual Exportar **Importar** Configuración Variables Juegos de caracteres Motores

**Archivo a importar:**

El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido. Un archivo comprimido tiene que terminar en **[formato].[compresión]**. Por ejemplo: **.sql.zip**

Buscar en su ordenador: **Seleccionar archivo** bolsa.sql (Máximo: 50MB)

Conjunto de caracteres del archivo: utf-8

**Importación parcial:**

Permitir la interrupción de una importación en caso que el script detecte que se ha acercado al límite de tiempo PHP. (Esto podría ser un buen método para importar archivos grandes; sin embargo, puede dañar las transacciones.)

Número de filas a omitir, iniciando de la primera fila: 0

**Formato:**

SQL

**Opciones específicas al formato:**

Modalidad SQL compatible: NONE

No utilizar AUTO\_INCREMENT con el valor 0

**Continuar**

phpMyAdmin

Tablales recientes ...

donoydl\_moviliana

- acuerdo
- lote
- lote\_infraestructura
- lote\_multa
- lotizacion
- manzana
- multa
- obras\_infraestructura
- pago
- tipo\_pago
- tipo\_usuario
- transaccion
- usuario
- information\_schema

Configuraciones generales

Cotejamiento de la conexión al servidor: utf8\_general\_ci

Configuraciones de apariencia

Idioma - Language: Español - Spanish

Tema: pmahomme

Tamaño de fuente: 82%

Más configuraciones

Servidor de base de datos

- Servidor: Localhost via UNIX socket
- Tipo de servidor: MariaDB
- Versión del servidor: 10.1.24-MariaDB-ell-ive - MariaDB Server
- Versión del protocolo: 10
- Usuario: donoydl@localhost
- Conjunto de caracteres del servidor: UTF-8 Unicode (utf8)

Servidor web

- cpsrvd 11.60.0.46
- Versión del cliente de base de datos: libmysql - 5.1.73
- extensión PHP: mysql

phpMyAdmin

- Acerca de esta versión: 4.0.10.14
- Documentación
- Wiki
- Página oficial de phpMyAdmin
- Contribuir
- Obtener soporte
- Lista de cambios

# **ANEXO III**

# **ENTREVISTA**

**ENTREVISTA PARA LA REALIZACIÓN DEL SISTEMA WEB DE GESTIÓN DE CONFRONTAS Y CONTROL DEL SERVICIO DE ALIMENTACIÓN EN LA BRIGADA DE CABALLERÍA BLINDADA N° 11 “GALÁPAGOS”.**

**NOMBRE DEL ENTREVISTADO:** \_\_\_\_\_

**CARGO QUE DESEMPEÑA:** \_\_\_\_\_

1. CUÁL ES EL PROCESO QUE SE REALIZA ACTUALMENTE PARA EL REGISTRO Y CONTROL DEL SERVICIO DE ALIMENTACIÓN DENTRO DE LA 11-BCB “GALÁPAGOS”.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

2. COMO SE REALIZA EL REGISTRO Y CONTROL DEL SERVICIO DE ALIMENTACIÓN EN LA 11-BCB “GALÁPAGOS”.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

3. CONOCE USTED LOS BENEFICIOS DE UN SISTEMA INFORMÁTICO?

SI ( )

NO ( )

4. Ha utilizado usted alguna vez un sistema informático?

SI ( )

NO ( )

**Cual:** \_\_\_\_\_

5. Si se le facilitara el registro y control del servicio de alimentación respecto a las confrontas mediante la implementación de un sistema Informático, usted estaría dispuesto a utilizarlo?

SI ( )

NO ( )

**ANEXO IV**  
**ANTEPROYECTO**

**PROYECTO DE TESIS**  
**INSTITUTO TECNOLÓGICO SUPERIOR**  
**“SAN GABRIEL”**



DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB, PARA EL PROCESO DE REGISTRO Y CONTROL DEL SERVICIO DE ALIMENTACIÓN EN LA BRIGADA DE CABALLERÍA BLINDADA N°11 “GALÁPAGOS” DESARROLLADO EN PHP ORIENTADO A OBJETOS CON MOTOR DE BASE DE DATOS MYSQL.

**PREVIA A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN  
INFORMÁTICA**

**MENCIÓN: ANÁLISIS EN SISTEMAS**

**PRESENTADO POR: DIEGO A. MACAS V.**

ANALIZADO Y APROBADO POR LA COMISION TECNICA DEL “ITSGA”.

## Contenido

ASPECTOS GENERALES .....	192
1.1 TÍTULO DEL PROYECTO DE TESIS .....	192
1.2 PROPONENTE .....	192
1.3 ASESOR DOCENTE .....	192
1.4 LUGAR DE REALIZACIÓN: .....	192
1.5 TIEMPO DE DURACIÓN:.....	192
1.6 FECHA ESTIMADA DE INICIACIÓN:.....	192
FORMULACIÓN GENERAL DEL PROYECTO DE TESIS .....	193
2.1 ANTECEDENTES .....	193
2.2 JUSTIFICACIÓN DEL PROYECTO DE TESIS. ....	193
2.3 OBJETIVO .....	194
2.3.1 OBJETIVO GENERAL: .....	194
2.3.2 OBJETIVOS ESPECÍFICOS: .....	194
2.4 MARCO TEÓRICO TENTATIVO .....	195
2.4.1 LENGUAJE PROGRAMACIÓN PHP. ....	195
2.4.2 PROGRAMACIÓN ORIENTADA A OBJETOS.....	210
2.4.3 MODELO VISTA CONTROLADOR .....	215
2.4.4 BASES DE DATOS. ....	218
2.5 BIBLIOGRAFÍA .....	221
EJECUCIÓN DEL PROYECTO DE INVESTIGACIÓN TESIS. ....	222
3.1 PLAN GENERAL DE TRABAJO .....	222
3.1.1 ANÁLISIS PRELIMINAR DEL PROBLEMA. ....	222
3.1.2 DESARROLLO DEL PROYECTO.....	225
3.1.3 DISEÑO CONCEPTUAL Y LÓGICO DEL SISTEMA. ....	226
3.2 HIPÓTESIS .....	227
3.3 VARIABLES.....	227
3.3.1 VARIABLE INDEPEMDIENTE.....	227
3.3.2 VARIABLE DEPENDIENTE.....	227
3.3.3 OPERACIONALIZACIÓN DE LAS VARIABLES. ....	227
3.4 CRONOGRAMA TENTATIVO.....	228
3.5 RECURSOS.....	228
3.5.1 RECURSOS HARDWARE .....	228

3.5.2 RECURSOS SOFTWARE.....	229
3.5.3 RECURSOS MATERIALES .....	229
3.6 MÉTODOS Y TÉCNICAS. ....	230
3.6.1 MÉTODOS.....	230
3.6.2 TÉCNICAS.....	230
3.7 PRESUPUESTO.....	230
3.8 FUENTE DE FINANCIAMIENTO .....	231
3.9 FECHA DE APROBACIÓN.....	231

### ÍNDICE DE FIGURAS

Figura 1 PHP .....	195
Figura 2 Variables .....	200
Figura 3 Variables Predefinidas .....	200
Figura 4 Variables Variables .....	201
Figura 5 Constantes .....	201
Figura 6 Precedencia de los operadores .....	203
Figura 7 Sintaxis IF .....	204
Figura 8 Sintaxis IF ELSE.....	204
Figura 9 Sentencia ELSEIF .....	205
Figura 10 Sintaxis de Estructuras de Control .....	205
Figura 11 Sintaxis IF .....	206
Figura 12 Sintaxis DO-WHILE.....	206
Figura 13 Snntencia FOR .....	207
Figura 14 Sintaxis FOREACH .....	208
Figura 15 Sintaxis SWITCH.....	209
Figura 15 Programación Orientada a Objetos .....	216
Figura 17 Función del MVC.....	217
Figura 18 Implementación del MVC.....	218

### ÍNDICE DE TABLAS

Tabla 1 Operacionalización de variables.....	228
Tabla 2 Cronograma Tentativo.....	228
Tabla 3 Recursos Harware.....	229
Tabla 4 Recursos Software .....	229
Tabla 5 Recursos Materiales.....	230
Tabla 6 Presupuesto.....	230

## **ASPECTOS GENERALES**

### **1.1 TÍTULO DEL PROYECTO DE TESIS**

Diseño e implementación de un sistema web, para el proceso de registro y control del servicio de alimentación en la Brigada de Caballería Blindada N°11 “GALÁPAGOS” desarrollado en PHP Orientado a Objetos con motor de base de datos MYSQL.

### **1.2 PROPONENTE**

**NOMBRE:** DIEGO ARMANDO MACAS VILEMA.

**ESPECIALIDAD:** TECNÓLOGO EN INFORMÁTICA, MENCIÓN ANALISTA DE SISTEMAS.

**ÁREA:** SISTEMAS.

### **1.3 ASESOR DOCENTE**

### **1.4 LUGAR DE REALIZACIÓN:**

BRIGADA DE CABALLERÍA BLINDADA N°11 “GALÁPAGOS”.

### **1.5 TIEMPO DE DURACIÓN:**

SIETE MESES

### **1.6 FECHA ESTIMADA DE INICIACIÓN:**

OCTUBRE 2016

## **FORMULACIÓN GENERAL DEL PROYECTO DE TESIS**

### **2.1 ANTECEDENTES**

En la actualidad el registro y control del servicio de alimentación dentro de la Brigada de Caballería Blindada N° 11 “Galápagos”, se lo realiza a través del Departamento de Recursos Humanos (B-1), en conjunto con las oficinas de Recursos Humanos (P-1) de cada una de las unidades que tiene bajo su responsabilidad el registro, control y manejo en el cobro del servicio de alimentación del personal de señores Oficiales, Voluntarios y Conscriptos bajo su responsabilidad.

Este proceso se lo realiza de forma manual, en base a la Directiva N° 04 del 2010 para la administración de la asignación del rancho del personal militar de las Fuerzas Armadas, Instructivo N° FT-DDIE-2015-004-O para la administración del servicio de alimentación en las unidades militares de la fuerza terrestre, publicada en Orden General N° 03 de fecha 06-ENE-2016, mediante la recopilación de datos proporcionados por el AMANUENSE DE PERSONAL (persona encargada de la recopilación de información referente a la confronta diaria) de las unidades y documentos emitidos por medio de tablas de Excel, enfrentando problemas de dificultad, lentitud, desorganización e inconsistencias de datos así como inconformidad por parte del personal, los cuales imposibilitan la optimización del tiempo al momento de realizar la liquidación mensual de este servicio.

Así como el control de este servicio no es el adecuado y en la actualidad se lo realiza por medio de tickets emitidos por el Sr. Oficial rancho acorde al registro de las confrontas diarias, exponiéndose a problemas como pérdida de los mismos así como duplicación de estos al ser impresiones comunes y no tener seguridades.

### **2.2 JUSTIFICACIÓN DEL PROYECTO DE TESIS.**

Debido a la gran exigencia y demanda en la actualización, modernización de los sistemas así como los procesos realizados manualmente en la actualidad es importante enfocarse en el desarrollo y modernización de sistemas informáticos que nos permitan digitalizar estas actividades de manera ágil y eficiente, de lo cual surge la necesidad de implementar un sistema web interno que nos permita digitalizar el sistema de registro del servicio de alimentación dentro de la Brigada de Caballería Blindada N° 11 “GALAPAGOS”. Para el desarrollo y programación del presente proyecto se utilizara el lenguaje de programación PHP, el modelo vista controlador, la programación orientada a objetos y las herramientas que sean necesarias durante el desarrollo del

presente sistema, de la misma manera utilizaremos el motor de base de datos My-SQL mismo nos permitirá levantar la base de datos soporte de este sistema.

Al implementar el presente proyecto se pretende desarrollar un sistema que permitirá la digitalización de esta actividad, la cual contará con una interfaz amigable con el usuario, estableciendo políticas de acceso, brindando un detalle de la información en parámetros como consumo cada uno de los usuarios o de las unidades de forma total o parcial partiendo de la directiva vigente que regula el consumo del rancho dentro de las unidades militares, permitiendo un control más óptimo y detallado al momento de realizar el cobro de este servicio, lo cual establecerá una mejor administración así como un control óptimo y moderno de este servicio.

### **3 OBJETIVO**

#### **3.1 OBJETIVO GENERAL:**

Diseñar e implementar un sistema web, para el proceso de registro y control del servicio de alimentación en la Brigada de Caballería Blindada N°11 “GALÁPAGOS” desarrollado en PHP Orientado a Objetos con motor de base de datos My-SQL.

#### **3.2 OBJETIVOS ESPECÍFICOS:**

Establecer y determinar las herramientas disponibles y necesarias para el desarrollo e implementación del presente sistema.

Analizar las características, ventajas, desventajas y todas las herramientas que dispone para la creación del presente sistema.

Diseñar un sistema web interno para el registro y control del servicio de alimentación dentro de la Brigada de Caballería Blindad N° 11 “GALÁPAGOS”.

Implementar el sistema web interno para el registro y control de las confrontas del servicio de alimentación dentro de la Brigada de Caballería Blindad N° 11 “GALÁPAGOS”.

## 4 MARCO TEÓRICO TENTATIVO

### LENGUAJE PROGRAMACIÓN PHP.

#### INTRODUCCIÓN<sup>47</sup>



Figura 2 PHP

**Fuente:** <http://php.net/manual/es/security.intro.php>

PHP es un potente lenguaje, y su intérprete, bien como módulo del servidor web o bien como binario CGI, puede acceder a ficheros, ejecutar comandos o abrir conexiones de red desde el servidor. Estas propiedades hacen que, por omisión, sea inseguro todo lo que se ejecute en un servidor web. PHP está diseñado específicamente para ser un lenguaje más seguro para escribir aplicaciones CGI que Perl o C. Partiendo de un correcto ajuste de opciones de configuración para tiempo de ejecución y en tiempo de compilación, y el uso de prácticas de programación apropiadas, pueden proporcionarle la combinación de libertad y de seguridad que necesita.

Dado que hay muchas vías para ejecutar PHP, existen muchas opciones de configuración para controlar su comportamiento. Al haber una extensa selección de opciones se garantiza poder usar PHP para un gran número de propósitos, pero a la vez significa que existen combinaciones que conllevan una configuración menos segura.

La flexibilidad de configuración de PHP rivaliza igualmente con la flexibilidad de su código. PHP puede ser usado para construir completas aplicaciones de servidor, con toda la potencia de un usuario de consola, o se puede usar sólo desde el lado del servidor implicando un menor riesgo dentro de un entorno controlado. El cómo construir ese entorno, y cómo de seguro es, depende del desarrollador PHP.

---

<sup>47</sup> <http://php.net/manual/es/security.intro.php>

## **IMPORTANCIA**

Este lenguaje es de gran importancia, ya que a pesar de ser un lenguaje de interpretación del lado del servidor y que necesita de un local-server para su ejecución en el ordenador es muy sencillo de aprender, la simplicidad y la enorme capacidad son las dos cualidades que caracterizan este lenguaje a pesar de su relativamente corto tiempo de existencia se ha convertido en el más popular y usado en estos últimos años, este lenguaje de programación tiene la facilidad de interactuar con el HTML.  
<http://es.slideshare.net/Gledsi/php-y-mysql>

## **CARACTERÍSTICAS**

Dentro de las principales características que tiene este lenguaje de programación mencionaremos que es un lenguaje de programación orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos, se caracteriza por su potencia, versatilidad, robustez y modularidad, es considerado un lenguaje fácil de aprender de programación interpretado de código abierto, el código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegadores, tiene la capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL, es libre, por lo que se presenta como una alternativa de fácil acceso para todos así como permite aplicar técnicas de programación orientada a objetos.

## **VENTAJAS Y DESVENTAJAS<sup>48</sup>**

### **VENTAJAS:**

Es un lenguaje multiplataforma.

Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.

---

<sup>48</sup> <http://klarimartinezbenjumea.blogspot.com/2011/04/ventajas-y-desventajas.html>

El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.

Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.

Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).

Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.

Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

Permite aplicar técnicas de programación orientada a objetos.

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable.

### **DESVENTAJAS:**

Como es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado.

Debido a que es un lenguaje interpretado, un script en PHP suele funcionar considerablemente más lento que su equivalente en un lenguaje de bajo nivel.

En las versiones previas a la 7, las variables no son tipificadas, lo cual dificulta a los diferentes IDEs ofrecer asistencias para el tipificado del código.

### **ESTRUCTURAS DEL CÓDIGO FUENTE EN PHP**

#### **ASPECTOS SOBRE LA SINTAXIS DEL LENGUAJE.**

La sintaxis del lenguaje PHP es muy similar a la de otros lenguajes conocidos como C o Perl, algunos aspectos iniciales sobre la sintaxis que deben ser considerados al momento de la escritura del código son:

El uso de mayúsculas y minúsculas es diferente, a diferencia de otros lenguajes similares, PHP permite que las sentencias estén escritas con cualquier combinación de mayúsculas o minúsculas a excepción de las variables.

Se pueden colocar todos los espacios en blanco y cambios de línea que se deseen con el objeto de hacer más legible el código fuente. (el intérprete PHP los ignora).

Las instrucciones deben terminar con ";" sin embargo PHP admite cierta flexibilidad al respecto, ya que el terminador ; no sería necesario en la última sentencia del script antes del terminador ;>, por ejemplo las estructuras condicionales o bucles.

Las instrucciones se pueden partir en tantas líneas como se desee a efectos, de nuevo, de mejorar la legibilidad, esa división de las instrucciones no afectan su ejecución.

En una misma línea se puede colocar, sí así se desea, varias instrucciones diferentes separadas por ";"

## ELEMENTOS BÁSICOS DEL LENGUAJE PHP

### TIPOS DE DATOS

PHP admite ocho tipos de datos primitivos:

#### CUATRO DE TIPOS ESCALARES QUE SON:

**BOOLEAN** .- Este es el tipo más simple, un boolean expresa un valor que indica verdad. Puede ser TRUE (verdadero) o FALSE (falso).

**INTEGER**.- Un número entero (o integer) es un número del conjunto  $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ , los integer pueden especificarse mediante notación decimal (base 10), hexadecimal (base 16), octal (base 8) o binaria (base 2), opcionalmente precedidos por un signo (- o +).

**FLOAT**.- (número de punto flotante, también conocido como double), los números de punto flotante (también conocidos como "de coma flotante" en español, y "floats" en inglés) pueden ser especificados usando cualquiera de las siguientes sintaxis:

**STRING**.- Un string, o cadena, es una serie de caracteres donde cada carácter es lo mismo que un byte. Esto significa que PHP solo admite un conjunto de 256 caracteres, y de ahí que no ofrezca soporte nativo para Unicode. Véanse los detalles del tipo string.

**Nota:** Un string puede llegar a alcanzar hasta 2 GB de tamaño (2147483647 bytes máximo).

**ARRAY**.- Un array en PHP es en realidad un mapa ordenado es decir un tipo de datos que asocia *valores* con *claves*.

Este tipo se optimiza para varios usos diferentes; se puede emplear como un array, lista (vector), tabla asociativa (tabla hash - una implementación de un mapa), diccionario, colección, pila, cola, y posiblemente más. Ya que los valores de un array pueden ser otros arrays, también son posibles árboles y arrays multidimensionales.

**OBJECT.-** Para crear un nuevo object, utilice la sentencia *new* para instanciar una clase:

**CALLABLE.-** (Llamadas de retorno, o retrollamadas) se pueden indicar con la declaración de tipo callable a partir de PHP 5.4., algunas funciones como `call_user_func()` o `usort()` aceptan como parámetro funciones de llamada de retorno definidas por el usuario, las funciones de llamadas de retorno no sólo pueden ser funciones simples, sino también métodos de un object, incluyendo métodos de clase estáticos.

## **FINALMENTE DOS TIPOS ESPECIALES:<sup>49</sup>**

**RESOURCE.-** Un valor tipo resource es una variable especial, que contiene una referencia a un recurso externo. Los recursos son creados y usados por funciones especiales. Vea el apéndice para un listado de todas estas funciones y los tipos resource correspondientes.

**NULL.-** El valor especial NULL representa una variable sin valor. NULL es el único valor posible del tipo null.

Una variable es considerada null si:

Se le ha asignado la constante NULL.

No se le ha asignado un valor todavía.

Se ha destruido con `unset()`.

### 4.2.1.6.2 VARIABLES

En PHP las variables se representan con un signo de dólar seguido por el nombre de la variable. El nombre de la variable es sensible a minúsculas y mayúsculas.

Los nombres de variables siguen las mismas reglas que otras etiquetas en PHP. Un nombre de variable válido tiene que empezar con una letra o un carácter de subrayado (underscore), seguido de cualquier número de letras, números y caracteres de subrayado. Como expresión regular se podría expresar como: `'[a-zA-Z_\x7f-\xff][a-zA-Z0-9_\x7f-\xff]*'`

---

<sup>49</sup> <http://php.net/manual/es/language.types.php>

```
<?php
$var = 'Roberto';
$Var = 'Juan';
echo "$var, $Var";      // imprime "Roberto, Juan"

$4site = 'aun no';     // inválido; comienza con un número
$_4site = 'aun no';   // válido; comienza con un carácter de subrayado
$täyte = 'mansikka';  // válido; 'ä' es ASCII (Extendido) 228
?>
```

**Figura 3 Variables**

**Fuente:** <http://php.net/manual/es/language.types.php>

### **VARIABLES PREDIFINIDAS<sup>50</sup>**

PHP proporciona una gran cantidad de variables predefinidas a cualquier script que se ejecute. Muchas de éstas, sin embargo, no pueden ser completamente documentadas ya que dependen del servidor que esté corriendo, la versión y configuración de dicho servidor, y otros factores. Algunas de estas variables no estarán disponibles cuando se ejecute PHP desde la línea de comandos. Para obtener una lista de estas variables, por favor vea la sección sobre Variables Predefinidas Reservadas.

### **AMBITO DE LAS VARIABLES**

El ámbito de una variable es el contexto dentro del que la variable está definida. La mayor parte de las variables PHP sólo tienen un ámbito simple. Este ámbito simple también abarca los ficheros incluidos y los requeridos. Por ejemplo:

```
<?php
$a = 1;
include 'b.inc';
?>
```

**Figura 4 Variables Predefinidas**

**Fuente:** <http://php.net/manual/es/language.variables.php>

### **VARIABLES VARIABLES**

A veces es conveniente tener nombres de variables-variables. Dicho de otro modo, son nombres de variables que se pueden definir y usar dinámicamente. Una variable normal se establece con una sentencia como:

---

<sup>50</sup> <http://php.net/manual/es/language.variables.php>

```
<?php
$a = 'hola';
?>
```

Figura 5 Variables Variables

**Fuente:** <http://php.net/manual/es/language.variables.php>

#### 4.2.1.6.3 CONSTANTES <sup>51</sup>

Una constante es un identificador (nombre) para expresar un valor simple, como el nombre sugiere, este valor no puede variar durante la ejecución del script. (A excepción de las constantes mágicas, que en realidad no son constantes).

Una constante es sensible a mayúsculas por defecto, los identificadores de constantes siempre suelen declararse en mayúsculas, el nombre de una constante sigue las mismas reglas que cualquier otra etiqueta de PHP. Un nombre de constante válido empieza por una letra o sub-guión, seguido por cualquier número o letras, números o sub-guiones.

#### SINTAXIS

Se puede definir una constante usando la función `define()` o con la palabra reservada `const` fuera de la definición de una clase a partir PHP 5.3.0. Mientras que `define()` permite definir una constante con una expresión arbitraria, la palabra reservada `const` tiene restricciones que se resumen en el siguiente párrafo. Una vez que la constante está definida, no puede ser cambiada o redefinida.

```
<?php
// Funciona a partir de PHP 5.3.0
const CONSTANTE = 'Hola Mundo';

echo CONSTANTE;

// Funciona a partir de PHP 5.6.0
const OTRA_CONSTANTE = CONSTANTE.' Adiós Mundo';
echo OTRA_CONSTANTE;

const ANIMALES = array('perro', 'gato', 'pájaro');
echo ANIMALES[1]; // muestra "gato"

// Funciona a partir de PHP 7
define('ANIMALES', array(
    'perro',
    'gato',
    'pájaro'
));
echo ANIMALES[1]; // muestra "gato"
?>
```

Figura 6 Constantes

---

<sup>51</sup> <http://php.net/manual/es/language.constants.php>

**Fuente:** <http://php.net/manual/es/language.constants.php>

#### 4.2.1.6.4 OPERADORES<sup>52</sup>

Un operador es algo que toma uno más valores (o expresiones, en la jerga de programación) y produce otro valor (de modo que la construcción en si misma se convierte en una expresión).

Los operadores se pueden agrupar de acuerdo con el número de valores que toman. Los operadores unarios toman sólo un valor, por ejemplo! (el operador lógico de negación) o ++ (el operador de incremento).

Los operadores binarios toman dos valores, como los familiares operadores aritméticos + (suma) y - (resta), y la mayoría de los operadores de PHP entran en esta categoría. Finalmente, hay sólo un operador ternario, ? :, el cual toma tres valores; usualmente a este se le refiere simplemente como "el operador ternario" (aunque podría tal vez llamarse más correctamente como el operador condicional).

Una lista completa de operadores de PHP sigue en la sección Precedencia de Operadores. La sección también explica la precedencia y asociatividad de los operadores, las cuales gobiernan exactamente cómo son evaluadas expresiones que contienen varios diferentes operadores.

#### **PRECEDENCIA DE LOS OPERADORES**

La precedencia de un operador indica qué tan "estrechamente" se unen dos expresiones juntas. Por ejemplo, en la expresión  $1 + 5 * 3$ , la respuesta es 16 y no 18 porque el operador de multiplicación ("\*") tiene una precedencia mayor que el operador de adición ("+"). Los paréntesis pueden ser usados para forzar la precedencia, si es necesario. Por ejemplo:  $(1 + 5) * 3$  se evalúa como 18.

El uso de paréntesis, incluso cuando no es estrictamente necesario, a menudo puede aumentar la legibilidad del código haciendo grupos explícitamente en lugar de confiar en la precedencia y asociatividad implícitas del operador.

La siguiente tabla enumera los operadores en orden de precedencia, con los de más alta precedencia al inicio. Los operadores en la misma línea tienen igual precedencia, en cuyo caso la asociatividad decide el agrupamiento.

---

<sup>52</sup> <http://php.net/manual/es/language.operators.php>

Precedencia de operadores		
Asociatividad	Operadores	Información adicional
no asociativo	<i>clone new</i>	<a href="#">clone and new</a>
izquierda	[	<a href="#">array()</a>
derecha	**	<a href="#">aritmética</a>
derecha	++ -- ~ (int) (float) (string) (array) (object) (bool) @	<a href="#">tipos e incremento/decremento</a>
no asociativo	<i>instanceof</i>	<a href="#">tipos</a>
derecha	!	<a href="#">lógico</a>
izquierda	* / %	<a href="#">aritmética</a>
izquierda	+ - .	<a href="#">aritmética y string</a>
izquierda	<< >>	<a href="#">bit a bit</a>
no asociativo	< <= > >=	<a href="#">comparación</a>
no asociativo	== != === !== =< > <=>	<a href="#">comparación</a>
izquierda	&	<a href="#">bit a bit y referencias</a>
izquierda	^	<a href="#">bit a bit</a>
izquierda		<a href="#">bit a bit</a>
izquierda	&&	<a href="#">lógico</a>
izquierda		<a href="#">lógico</a>
derecha	??	<a href="#">comparación</a>
izquierda	? :	<a href="#">ternario</a>
derecha	= += -= *= **= /= .= %= &=  = ^= <<= >>=	<a href="#">asignación</a>
izquierda	<i>and</i>	<a href="#">lógico</a>
izquierda	<i>xor</i>	<a href="#">lógico</a>
izquierda	<i>or</i>	<a href="#">lógico</a>

**Figura 7 Precedencia de los operadores**

**Fuente:** <http://php.net/manual/es/language.operators.php>

#### 4.2.1.6.5 SENTENCIAS DE CONTROL<sup>53</sup>

Todo script PHP está construido según una serie de sentencias. Una sentencia puede ser una asignación, una llamada de función, un ciclo, una sentencia condicional o incluso una sentencia que no hace nada (una sentencia vacía).

Las sentencias generalmente finalizan con un punto y coma adicionalmente, las sentencias pueden agruparse en un conjunto de sentencias, encapsulándolas entre corchetes. Un grupo de sentencias es una sentencia por sí misma también.

<sup>53</sup> <http://php.net/manual/es/language.control-structures.php>

## SENTENCIA IF

El constructor *if* es una de las características más importantes de muchos lenguajes, incluido PHP. Permite la ejecución condicional de fragmentos de código. PHP dispone de una estructura *if* que es similar a la de C:

```
if (expr)  
    sentencia
```

Esta expresión es evaluada a su valor booleano, si la expresión se evalúa como TRUE, PHP ejecutará la sentencia y si se evalúa como FALSE la ignorará.

```
<?php  
if ($a > $b) {  
    echo "a es mayor que b";  
    $b = $a;  
}  
?>
```

Figura 8 Sintaxis IF

**Fuente:** <http://php.net/manual/es/language.control-structures.php>

## SENTENCIA ELSE

Con frecuencia se desea ejecutar una sentencia si una determinada condición se cumple y una sentencia diferente si la condición no se cumple esto es para lo que sirve *else*.

El *else* extiende una sentencia *if* para ejecutar una sentencia en caso que la expresión en la sentencia *if* se evalúe como FALSE. Por ejemplo, el siguiente código deberá mostrar a es mayor que b si \$a es mayor que \$b y a NO es mayor que b en el caso contrario:

```
<?php  
if ($a > $b) {  
    echo "a es mayor que b";  
} else {  
    echo "a NO es mayor que b";  
}  
?>
```

Figura 9 Sintaxis IF ELSE

**Fuente:** <http://php.net/manual/es/language.control-structures.php>

## SENTENCIA ELSEIF/ELSE IF

*elseif*, como su nombre lo sugiere, es una combinación de *if* y *else*. Del mismo modo que *else*, extiende una sentencia *if* para ejecutar una sentencia diferente en caso que la

expresión *if* original se evalúe como FALSE. Sin embargo, a diferencia de *else*, esa expresión alternativa sólo se ejecutará si la expresión condicional del *elseif* se evalúa como TRUE.

Por ejemplo, el siguiente código debe mostrar a es mayor que b, a es igual que b o a es menor que b:

```
<?php
if ($a > $b) {
    echo "a es mayor que b";
} elseif ($a == $b) {
    echo "a es igual que b";
} else {
    echo "a es menor que b";
}
?>
```

Figura 10 Sentencia ELSEIF

**Fuente:** <http://php.net/manual/es/language.control-structures.php>

### SINTAXIS ALTERNATIVA DE ESTRUCTURAS DE CONTROL

PHP ofrece una sintaxis alternativa para algunas de sus estructuras de control; a saber: *if*, *while*, *for*, *foreach*, y *switch*. En cada caso, la forma básica de la sintaxis alternativa es cambiar el corchete de apertura por dos puntos (:) y el corchete de cierre por *endif*;, *endwhile*;, *endfor*;, *endforeach*;, o *endswitch*;, respectivamente.

```
<?php if ($a == 5): ?>
A es igual a 5
<?php endif; ?>
```

Figura 11 Sintaxis de Estructuras de Control

**Fuente:** <http://php.net/manual/es/language.control-structures.php>

### SENTENCIA WHILE

Los bucles *while* son el tipo más sencillo de bucle en PHP. Se comportan igual que su contrapartida en C. La forma básica de una sentencia *while* es:

```
while (expr)
    sentencia
```

El significado de una sentencia *while* es simple. Le dice a PHP que ejecute las sentencias anidadas, tanto como la expresión *while* se evalúe como TRUE. El valor de la expresión es verificado cada vez al inicio del bucle, por lo que incluso si este valor cambia durante la ejecución de las sentencias anidadas, la ejecución no se detendrá hasta el final de la iteración (cada vez que PHP ejecuta las sentencias contenidas en el

bucle es una iteración). A veces, si la expresión *while* se evalúa como FALSE desde el principio, las sentencias anidadas no se ejecutarán ni siquiera una vez.

Al igual que con la sentencia *if*, se pueden agrupar varias instrucciones dentro del mismo bucle *while* rodeando un grupo de sentencias con corchetes, o utilizando la sintaxis alternativa:

```
while (expr):  
    sentencias  
  
    ...  
endwhile;
```

```
<?php  
/* ejemplo 1 */  
  
$i = 1;  
while ($i <= 10) {  
    echo $i++; /* el valor presentado sería  
               $i antes del incremento  
               (post-incremento) */  
}  
  
/* ejemplo 2 */  
  
$i = 1;  
while ($i <= 10):  
    echo $i;  
    $i++;  
endwhile;  
?>
```

Figura 12 Sintaxis IF

**Fuente:** <http://php.net/manual/es/language.control-structures.php>

### SENTENCIA *DO-WHILE*

Los bucles *do-while* son muy similares a los bucles *while*, excepto que la expresión verdadera es verificada al final de cada iteración en lugar que al principio. La diferencia principal con los bucles *while* es que está garantizado que corra la primera iteración de un bucle *do-while* (la expresión verdadera sólo es verificada al final de la iteración), mientras que no necesariamente va a correr con un bucle *while* regular (la expresión verdadera es verificada al principio de cada iteración, si se evalúa como FALSE justo desde el comienzo, la ejecución del bucle terminaría inmediatamente).

```
<?php  
$i = 0;  
do {  
    echo $i;  
} while ($i > 0);  
?>
```

Figura 13 Sintaxis DO-WHILE

**Fuente:** <http://php.net/manual/es/language.control-structures.php>

## SENTENCIA *FOR*

Los bucles *for* son los más complejos en PHP. Se comportan como sus homólogos en C.

La sintaxis de un bucle *for* es:

```
for (expr1; expr2; expr3)
    sentencia
```

La primera expresión (*expr1*) es evaluada (ejecutada) una vez incondicionalmente al comienzo del bucle.

En el comienzo de cada iteración, se evalúa *expr2*. Si se evalúa como TRUE, el bucle continúa y se ejecutan la/s y sentencia/s anidada/s. Si se evalúa como FALSE, finaliza la ejecución del bucle.

Al final de cada iteración, se evalúa (ejecuta) *expr3*.

Cada una de las expresiones puede estar vacía o contener múltiples expresiones separadas por comas. En *expr2*, todas las expresiones separadas por una coma son evaluadas, pero el resultado se toma de la última parte. Que *expr2* esté vacía significa que el bucle debería ser corrido indefinidamente (PHP implícitamente lo considera como TRUE, como en C). Esto puede no ser tan inútil como se pudiera pensar, ya que muchas veces se debe terminar el bucle usando una sentencia condicional *break* en lugar de utilizar la expresión verdadera del *for*.

```
<?php
/* ejemplo 1 */

for ($i = 1; $i <= 10; $i++) {
    echo $i;
}

/* ejemplo 2 */

for ($i = 1; ; $i++) {
    if ($i > 10) {
        break;
    }
    echo $i;
}

/* ejemplo 3 */

$i = 1;
for ( ; ; ) {
    if ($i > 10) {
        break;
    }
    echo $i;
    $i++;
}

/* ejemplo 4 */

for ($i = 1, $j = 0; $i <= 10; $j += $i, print $i, $i++);
?>
```

Figura 14 Sentencia FOR

**Fuente:** <http://php.net/manual/es/language.control-structures.php>

## SENTENCIA FOR EACH

El constructor *foreach* proporciona un modo sencillo de iterar sobre arrays. *foreach* funciona sólo sobre arrays y objetos, y emitirá un error al intentar usarlo con una variable de un tipo diferente de datos o una variable no inicializada. Existen dos sintaxis:

```
foreach (expresión_array as $valor)
```

```
    sentencias
```

```
foreach (expresión_array as $clave => $valor)
```

```
    sentencias
```

La primera forma recorre el array dado por *expresión\_array*. En cada iteración, el valor del elemento actual se asigna a *\$valor* y el puntero interno del array avanza una posición (así en la próxima iteración se estará observando el siguiente elemento).

La segunda forma además asigna la clave del elemento actual a la variable *\$clave* en cada iteración.

```
<?php
$array = array(1, 2, 3, 4);
foreach ($array as $valor) {
    $valor = $valor * 2;
}
// $array ahora es array(2, 4, 6, 8)
unset($valor); // rompe la referencia con el último elemento
?>
```

Figura 15 Sintaxis FOREACH

**Fuente:** <http://php.net/manual/es/language.control-structures.php>

## SENTENCIA SWITCH

La sentencia *switch* es similar a una serie de sentencias IF en la misma expresión. En muchas ocasiones, es posible que se quiera comparar la misma variable (o expresión) con muchos valores diferentes, y ejecutar una parte de código distinta dependiendo de qué valor es igual, para esto es exactamente la expresión *switch*.

```

<?php
if ($i == 0) {
    echo "i es igual a 0";
} elseif ($i == 1) {
    echo "i es igual a 1";
} elseif ($i == 2) {
    echo "i es igual a 2";
}

switch ($i) {
    case 0:
        echo "i es igual a 0";
        break;
    case 1:
        echo "i es igual a 1";
        break;
    case 2:
        echo "i es igual a 2";
        break;
}
?>

```

Figura 16 Sintaxis SWITCH

**Fuente:** <http://php.net/manual/es/language.control-structures.php>

### SENTENCIA DECLARE

El constructor *declare* es usado para fijar directivas de ejecución para un bloque de código. La sintaxis de *declare* es similar a la sintaxis de otros constructores de control de flujo:

declare (directive)

statement

La sección directive permite que el comportamiento de declare sea configurado. Actualmente, solamente están reconocidas tres directivas: ticks (véase abajo para más información sobre la directiva ticks), encoding (véase abajo para más información sobre la directiva encoding) y strict\_types (para más información, véase la sección strict de la página de Argumentos de funciones)

### SENTENCIA RETURN

Devuelve el control del programa al módulo que lo invoca la ejecución vuelve a la siguiente expresión después del módulo que lo invoca.

Si se llama desde una función, la sentencia *return* inmediatamente termina la ejecución de la función actual, y devuelve su argumento como el valor de la llamada a la función. *return* también termina la ejecución de una sentencia eval() o un archivo de script.

Cabe señalar que dado que *return* es un constructor del lenguaje y no una función, los paréntesis que rodean sus argumentos no son necesarios. Es común no utilizarlos, y en realidad se debería hacer así a fin de que PHP tenga menos trabajo que hacer en este caso.

## **SENTENCIA REQUIRE**

Require es idéntico a include excepto que en caso de fallo producirá un error fatal de nivel E\_COMPILE\_ERROR. En otras palabras, éste detiene el script mientras que include sólo emitirá una advertencia (E\_WARNING) lo cual permite continuar el script.

## **SENTENCIA INCLUDE**

La sentencia *include* incluye y evalúa el archivo especificado, los archivos son incluidos con base en la ruta de acceso dada o, si ninguna es dada, el `include_path` especificado. Si el archivo no se encuentra en el `include_path`, *include* finalmente verificará en el propio directorio del script que hace el llamado y en el directorio de trabajo actual, antes de fallar. El constructor *include* emitirá una advertencia si no puede encontrar un archivo, éste es un comportamiento diferente al de `require`, el cual emitirá un error fatal..

## **SENTENCIA REQUIRE\_ONCE**

La sentencia `require_once` es idéntica a `require` excepto que PHP verificará si el archivo ya ha sido incluido y si es así, no se incluye (`require`) de nuevo.

Ver la documentación de `include_once` para información sobre el comportamiento de `_once`, y como difiere de sus hermanos no `_once`.

## **SENTENCIA INCLUDE\_ONCE**

La sentencia *include\_once* incluye y evalúa el fichero especificado durante la ejecución del script. Tiene un comportamiento similar al de la sentencia `include`, siendo la única diferencia de que si el código del fichero ya ha sido incluido, no se volverá a incluir, e `include_once` devolverá TRUE. Como su nombre indica, el fichero será incluido solamente una vez.

*include\_once* se puede utilizar en casos donde el mismo fichero podría ser incluido y evaluado más de una vez durante una ejecución particular de un script, así que en este caso, puede ser de ayuda para evitar problemas como la redefinición de funciones, reasignación de valores de variables, etc.

### **2.4.2 PROGRAMACIÓN ORIENTADA A OBJETOS<sup>54</sup>**

La programación orientada a objetos, ha tomado las mejores ideas de la programación estructurada y los ha combinado con varios conceptos nuevos y potentes que incitan a contemplar las tareas de programación desde un nuevo punto de vista. La programación

---

<sup>54</sup> [http://pdi.topografia.upm.es/m.manso/docencia/Informatica\\_plan92/Curso-2002-2003/poo.pdf](http://pdi.topografia.upm.es/m.manso/docencia/Informatica_plan92/Curso-2002-2003/poo.pdf)

orientada a objetos, permite descomponer más fácilmente un problema en subgrupos de partes relacionadas del problema.

Entonces, utilizando el lenguaje se pueden traducir estos subgrupos a unidades auto contenidas llamadas objetos, el término Programación Orientada a Objetos (POO), hoy en día ampliamente utilizado, es difícil de definir, ya que no es un concepto nuevo, sino que ha sido el desarrollo de técnicas de programación desde principios de la década de los setenta, aunque sea en la década de los noventa cuando ha aumentado su difusión, uso y popularidad. No obstante, se puede definir POO como una técnica o estilo de programación que utiliza objetos como bloque esencial de construcción, un objeto es una unidad que contiene datos y las funciones que operan sobre esos datos. A los elementos de un objeto se les conoce como miembros; las funciones que operan sobre los objetos se denominan métodos y los datos se denominan miembros datos.

#### 2.4.2.1 ORIGEN DE LA PROGRAMACIÓN ORIENTADA A OBJETOS<sup>55</sup>

##### **PRIMERA ETAPA LENGUAJES ENSAMBLADORES.**

La unidad de programación es la instrucción, compuesta de un operador y los operandos. El nivel de abstracción que se aplica es muy bajo.

##### **SEGUNDA ETAPA LENGUAJES DE PROGRAMACIÓN:**

Los objetos y operaciones del mundo real se podían modelar mediante datos y estructuras de control separadamente. En esta etapa el diseño del software se enfoca sobre la representación del detalle procedimental y en función del lenguaje elegido. Conceptos como: refinamiento progresivo, modularidad procedimientos y programación estructurada son conceptos básicos que se utilizan en esta etapa. Existe mayor abstracción de datos.

##### **TERCERA ETAPA.**

Se introducen en esta etapa los conceptos de abstracción y ocultación de la información. Programación Orientada a objetos. 2

##### **CUARTA ETAPA.**

A partir de los años setenta se trabaja sobre una nueva clase de lenguajes de simulación y sobre la construcción de prototipo tal como Simula-70 y basado en parte de éste, el Smalltalk. En estos lenguajes, la abstracción de datos tiene una gran importancia y los problemas del mundo real se representan mediante objetos de datos a los cuales se les añade el correspondiente conjunto de operaciones asociados a ellos. Términos como

---

<sup>55</sup> <http://dis.um.es/~jfernand/0506/dai/poo.pdf>

Abstracción de datos, objeto, encapsulación entre otros, son conceptos básicos sobre la que se fundamenta la POO.

#### CONCEPTOS DE LA PROGRAMACION ORIENTADA A OBJETOS.<sup>56</sup>

La POO representa una metodología de programación que se basa en las siguientes características:

- 1) Los diseñadores definen nuevas clases (o tipos) de objetos.
- 2) Los objetos poseen una serie de operaciones asociadas a ellos.
- 3) Las operaciones tienden a ser genéricas, es decir, operan sobre múltiples tipos de datos.
- 4) Las clases o tipos de objetos comparten componentes comunes mediante mecanismos de herencia.

**OBJETO:** Una estructura de datos y conjunto de procedimientos que operan sobre dicha estructura. Una definición más completa de objeto es: una entidad de programa que consiste en datos y todos aquellos procedimientos que pueden manipular aquellos datos; el acceso a los datos de un objeto es solamente a través de estos procedimientos, únicamente estos procedimientos pueden manipular, referenciar y/o modificar estos datos. Para poder describir todos los objetos de un programa, conviene agrupar éstos en clases.

**CLASE:** Podemos considerar una clase como una colección de objetos que poseen características y operaciones comunes. Una clase contiene toda la información necesaria para crear nuevos objetos.

**ENCAPSULACIÓN:** Es una técnica que permite localizar y ocultar los detalles de un objeto, la encapsulación previene que un objeto sea manipulado por operaciones distintas de las definidas. La encapsulación es como una caja negra que esconde los datos y solamente permite acceder a ellos de forma controlada. Las principales razones técnicas para la utilización de la encapsulación son:

- 1) Mantener a salvo los detalles de representación, si solamente nos interesa el comportamiento del objeto.
- 2) Modificar y ajustar la representación a mejores soluciones algorítmicas o a nuevas tecnologías de software. Programación Orientada a objetos.

---

<sup>56</sup> <http://dis.um.es/~jfernand/0506/dai/poo.pdf>

**ABSTRACCIÓN:** En el sentido más general, una abstracción es una representación concisa de una idea o de un objeto complicado, en un sentido más específico, la abstracción localiza y oculta los detalles de un modelo o diseño para generar y manipular objetos.

Una abstracción tiene un significado más general que la encapsulación, pudiendo hablar de abstracción de datos en lugar de encapsulación de datos. Como resumen de los 3 conceptos expuestos anteriormente podemos decir que:

- 1) Los objetos son encapsulaciones de abstracciones en la POO.
- 2) La unidad de encapsulación en la POO es el objeto. Una clase es un tipo: Un objeto es una instancia de ese tipo.

Además, la clase es un concepto estático: una clase es un elemento reconocible en el texto del programa.

Un objeto es un concepto puramente dinámico, el cual pertenece, no al texto del programa, sino a la memoria de la computadora, donde los objetos ocupan un espacio en tiempo de ejecución una vez que haya sido creado, la programación orientada a objetos, ha tomado las mejores ideas de la programación estructurada y los ha combinado con varios conceptos nuevos y potentes que incitan a contemplar las tareas de programación desde un nuevo punto de vista. La programación orientada a objetos, permite descomponer más fácilmente un problema en subgrupos de partes relacionadas del problema entonces, utilizando el lenguaje se pueden traducir estos subgrupos a unidades auto contenidas llamadas objetos.

**OBJETOS:** Un objeto es una entidad lógica que contiene datos y un código que manipula estos datos; el enlazado de código y de datos, de esta manera suele denominarse encapsulación. Cuando se define un objeto, se está creando implícitamente un nuevo tipo de datos.

**POLIMORFISMO:** Significa que un nombre se puede utilizar para especificar una clase genérica de acciones.

**HERENCIA:** La herencia es un proceso mediante el cual un objeto puede adquirir las propiedades de otro objeto.

#### CARACTERÍSTICAS DEL LENGUAJE ORIENTADOS A OBJETOS.

Además de las características que mencionamos anteriormente como esenciales de los lenguajes de programación orientados a objetos, es deseable que éstos cumplan también con las siguientes: **TIPIFICACIÓN FUERTE.**- Esto es, que durante la fase de diseño e implementación se declare que tipo de datos soportara cada variable.

**MANEJO DE EXCEPCIONES.-** Dentro de la misma definición del lenguaje se deberá establecer la forma de detectar y manipular excepciones que puedan surgir durante la ejecución de un programa.

**PASO DE MENSAJES.-** Es conveniente que el lenguaje soporte paso de mensajes entre módulos de manera bidireccional.

**GENERALIDAD.-** Se refiere principalmente a que las clases se definan lo más generalizadas posible para que sean fácilmente reusables. Para generar este tipo de clases, normalmente se definen parámetros formales que son instanciados por parámetros reales.

**MULTITAREA.-** Es conveniente que el lenguaje permita la creación de procesos que se ejecuten de forma simultánea independientemente del sistema operativo.

**PERSISTENCIA.-** Los objetos deben poder permanecer, si así se desea, después de la ejecución de un programa.

**DATOS COMPARTIDOS.-** Los objetos pueden necesitar referirse a la misma localidad de memoria (memoria compartida) o bien comunicarse mediante mensajes.

**VENTAJAS DE LA TECNOLOGÍA ORIENTADA A OBJETOS.**

**FLEXIBILIDAD.-** Si partimos del hecho que mediante la definición de clases establecemos módulos independientes, a partir de los cuales podemos definir nuevas clases, entonces podemos pensar en estos módulos como bloques con los cuales podemos construir diferentes programas.

**REUSABILIDAD.-** Una vez que hemos definido a la entidad persona para utilizarla en una aplicación de negocios, por mencionar un ejemplo, y deseamos construir a continuación una aplicación, digamos de deportes, en donde requerimos definir a la misma entidad persona, no es deseable volver a escribir la definición para la entidad persona, por medio de la reusabilidad podemos utilizar una clase definida previamente en las aplicaciones que nos sea conveniente. Es claro que la flexibilidad con la que se definió la clase va a ser fundamental para su reutilización.

**MANTENIBILIDAD.-** Las clases que conforman una aplicación, vistas como módulos independientes entre sí, son fáciles de mantener sin afectar a los demás componentes de la aplicación.

**EXTENSIBILIDAD.-** Gracias a la modularidad y a la herencia una aplicación diseñada bajo el paradigma de la orientación a objetos puede ser fácilmente extensible para cubrir necesidades de crecimiento de la aplicación.

**DESVENTAJAS DE LA TECNOLOGÍA ORIENTADA A OBJETOS.**

A pesar de que las ventajas de la programación orientada a objetos superan a las limitaciones de la misma, podemos encontrar algunas características no deseables en ésta.

**LIMITACIONES PARA EL PROGRAMADOR.-** No obstante que la tecnología orientada a objetos no es nueva, un gran porcentaje de programadores no están familiarizados con los conceptos de dicha tecnología en otras palabras, la lógica de la programación estructurada sigue siendo predominante en la mayoría de los desarrolladores de software, después de haber revisado de forma breve los principios de la programación orientada a objetos, nos es claro que en ésta se requiere una lógica de pensamiento totalmente diferente a la lógica comúnmente utilizada para la programación estructurada.

**TAMAÑO EXCESIVO EN LAS APLICACIONES RESULTANTES.-** La gran mayoría de los equipos de cómputo cuentan con capacidades tanto de almacenamiento como de memoria lo suficientemente buena como para ejecutar la mayoría de las aplicaciones que puedan desarrollarse con la tecnología orientada a objetos, sin embargo existen casos en los que lo anterior no se cumple.

Una de las desventajas de la programación orientada a objetos es que cuando se heredan clases a partir de clases existentes se heredan de forma implícita todos los miembros de dicha clase aun cuando no todos se necesiten, lo que produce aplicaciones muy grandes que no siempre encajan en los sistemas con los que se disponga.

**VELOCIDAD DE EJECUCIÓN.-** Esto tiene que ver, en cierto modo, con el punto anterior, una aplicación innecesariamente pesada en muchas ocasiones es más lenta de ejecutar que una aplicación conformada únicamente por los módulos necesarios.

## MODELO VISTA CONTROLADOR

### 2.4.3.1 INTRODUCCIÓN

El patrón MVC (Model, View, Controller) o Modelo, Vista Controlador, es un tipo de diseño que separa en capas bien definidas el desarrollo de una aplicación, esas partes son tres, el Modelo encargado de la lógica del negocio y la persistencia de los datos, las Vistas son las responsables de mostrar al usuario el resultado que obtienen del modelo a través del controlador, el Controlador encargado es el encargado de gestionar las peticiones del usuario, procesarlas invocando al modelo y mostrarlas al usuario a través de las vistas.

El modelo vista controlador es un patrón de arquitectura de software que nos aporta separando los datos de nuestra aplicación de la lógica y de la interfaz de usuario

obteniendo así tres componentes relacionados entre sí. Si bien la POO nos ayuda a clasificar nuestra aplicación, el MVC nos ayudará a separarlo por componentes (que desde ahora le llamaremos capas porque me parece un concepto más adecuado).

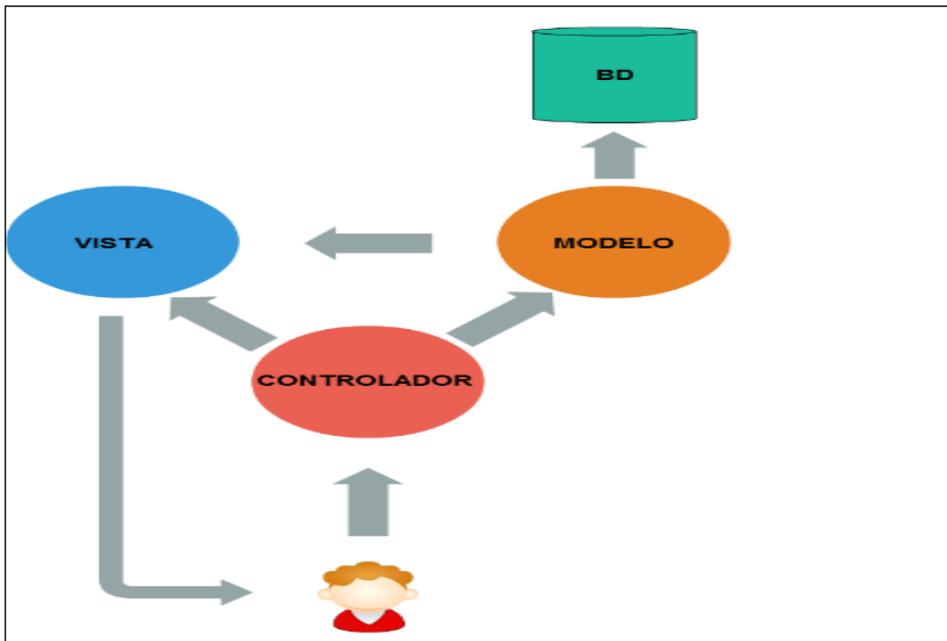


Figura 17 Programación Orientada a Objetos

**Fuente:** <http://dis.um.es/~jfernand/0506/dai/poo.pdf>

#### 2.4.3.2 CONCEPTOS DEL MVC

##### **MODELO**

Es todo lo referente a la gestión de la información y la interacción con los datos de nuestra aplicación (comúnmente bases de datos) este modelo realizará acceso a dicha información como también podrá realizar actualizaciones y depuraciones de los datos gestionados, toda petición de acceso a la información siempre pasará por esta capa.

##### **CONTROLADOR**

Este es el puente entre la vista y el modelo ya que el usuario solicitará información mediante la vista y este pasará por el controlador para posteriormente realizar la petición al modelo, habitualmente es llamado la capa de lógica del negocio.

##### **VISTA**

Esta capa nos mostrará la información formateada y ordenada, es el resultado de todo lo que el modelo interaccione con los datos, este lo muestra mediante la interfaz de usuario, habitualmente llamado la capa de presentación.

#### 2.4.3.4 CARACTERÍSTICAS

Debido a que es un patrón de diseño de software verdaderamente probado que convierte una aplicación en un paquete modular fácil de mantener y mejora la rapidez del desarrollo.

La separación de las tareas de tu aplicación en modelos, vistas y controladores hace que su aplicación sea muy ligera de entender.

Añadir y agregar fácilmente cosas nuevas a código viejo, se hace muy sencillo.

El diseño modular también permite a los desarrolladores y los diseñadores trabajar simultáneamente, incluyendo la capacidad de hacer prototipos.

La separación también permite a los desarrolladores hacer cambios en una parte de la aplicación sin afectar a los demás.

#### FUNCIONAMIENTO DEL MVC.

El funcionamiento básico del patrón MVC, puede resumirse en:

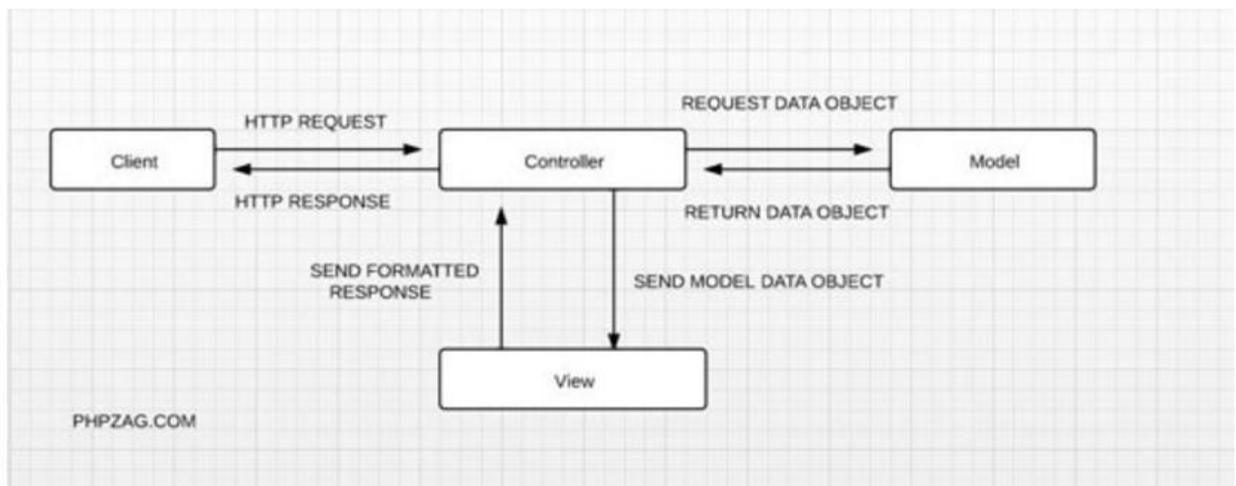


Figura 18 Función del MVC

**Fuente:** <http://dis.um.es/~jfernand/0506/dai/poo.pdf>

El usuario realiza una petición.

El controlador captura la petición.

Hace la llamada al modelo correspondiente.

El modelo será el encargado de interactuar con la base de datos.

El controlador recibe la información y la envía a la vista.

La vista muestra la información.

#### 2.4.3.6 IMPLEMENTACIÓN DEL MVC.

Para implementar el MVC es imprescindible crear una estructura de ficheros parecida a esta:



Figura 19 Implementación del MVC

**Fuente:** <http://dis.um.es/~jfernand/0506/dai/poo.pdf>

#### 2.4.4 BASES DE DATOS.

##### 2.4.4.1 INTRODUCCIÓN

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite, es un sistema de archivos electrónico.

Las bases de datos tradicionales se organizan por campos, registros y archivos.

Un campo es una pieza única de información;

Un registro es un sistema completo de campos;

Un archivo es una colección de registros.

Una base de datos es una herramienta para recopilar y organizar información, en las bases de datos, se puede almacenar información sobre personas, productos, pedidos, o cualquier otra cosa. Muchas bases de datos empiezan siendo una lista en un programa de procesamiento de texto o en una hoja de cálculo, a medida que crece la lista, empiezan a aparecer repeticiones e inconsistencias en los datos.

Cada vez resulta más complicado comprender los datos presentados en la lista y existen pocos métodos para buscar o recuperar subconjuntos de datos para revisarlos, cuando empiezan a observarse estos problemas, es aconsejable transferir la información a una base de datos creada mediante un sistema de administración de bases de datos (DBMS).

Una base de datos informatizada es un contenedor de objetos, puede contener más de una tabla. Por ejemplo, un sistema de seguimiento de inventario que utiliza tres tablas no es un conjunto de tres bases de datos, sino una sola base de datos que contiene tres tablas. Excepto si se ha diseñado específicamente para utilizar datos o código de otro origen, una base de datos almacena sus tablas en un solo archivo, junto con otros objetos, como formularios, informes, macros y módulos.

##### 2.4.4.2 CLASIFICACIÓN DE LA BASE DE DATOS

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se esté manejando, o la utilidad de la misma.

## **BASES DE DATOS ESTÁTICAS**

Estas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

## **BASES DE DATOS DINÁMICAS**

Estas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de un supermercado, una farmacia, un videoclub o una empresa.

### **2.4.4.3 MODELOS DE BASES DE DATOS**

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos. Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos , y conceptos matemáticos.

## **BASES DE DATOS JERÁRQUICAS**

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica en este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos, el nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

## **BASE DE DATOS DE RED**

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa

administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

### **BASES DE DATOS TRANSACCIONALES**

Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, estas bases son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad, datos de producción e industrial, es importante entender que su fin único es recolectar y recuperar los datos a la mayor velocidad posible, por lo tanto la redundancia y duplicación de información no es un problema como con las demás bases de datos, por lo general para poderlas aprovechar al máximo permiten algún tipo de conectividad a bases de datos relacionales.

### **BASES DE DATOS RELACIONALES**

Éste es el modelo utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en 1970, de los laboratorios Ibm en San José, no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos.

## 2.5 BIBLIOGRAFÍA

Abrahamsson, P. S. (2002). *"Agile software development methods Review and analysis"*. Recuperado el 15 de 06 de 2016, de [http://www.cyta.com.ar/ta0502/b\\_v5n2a1.htm](http://www.cyta.com.ar/ta0502/b_v5n2a1.htm)

ANGELES, J. A. (2005). *ORIGEN DE LA PROGRAMACION ORIENTADA A OBJETOS*. Recuperado el 25 de 11 de 2016, de [http://pdi.topografia.upm.es/m.manso/docencia/Informatica\\_plan92/Curso-2002-2003/poo.pdf](http://pdi.topografia.upm.es/m.manso/docencia/Informatica_plan92/Curso-2002-2003/poo.pdf)

ANGELES, J. A. (s.f.). *INTRODUCCION A LA PROGRAMACION ORIENTADA A OBJETOS*. Recuperado el 25 de 10 de 2016, de [http://pdi.topografia.upm.es/m.manso/docencia/Informatica\\_plan92/Curso-2002-2003/poo.pdf](http://pdi.topografia.upm.es/m.manso/docencia/Informatica_plan92/Curso-2002-2003/poo.pdf)

Cowburn, P. (12 de 2009). *MANUAL PHP*. Recuperado el 12 de 2016, de <https://secure.php.net/manual/es/index.php>

DANIEL PERZ, ROCIO ROCHA. (2005). *PHP Y MySQL TECNOLOGIAS PARA EL DESARROLLO APLICACIONES WEB*. ESPAÑA: DIAZ DE SANTOS.

fondo, P. A. (04 de septiembre de 2013). *estudio del lenguaje y desarrollo de aplicaciones (2a. ed.)*. Recuperado el 12 de 06 de 2016, de [http://www.consultorjava.com/wp/wp-content/uploads/2015/09/herramienta\\_desarrollo\\_netbeans.pdf](http://www.consultorjava.com/wp/wp-content/uploads/2015/09/herramienta_desarrollo_netbeans.pdf)

FRANCISCO DURAN, FRANCISCO GUTIERRES, ERNESTO PIMENTEL. (2007). *GOOGLE ACADEMICO*. Recuperado el 15 de 08 de 2016, de GOOGLE ACADEMICO:

<https://books.google.com.ec/books?hl=es&lr=&id=3EQdUbkOVGIC&oi=fnd&pg=PP1&dq=libro+programacion+orientada+a+objetos&ots=03hVuDzzPO&sig=HZ7yuKarcAYwR0fRdc->

[3U1SreYU#v=onepage&q=libro%20programacion%20orientada%20a%20objetos&f=false](https://books.google.com.ec/books?hl=es&lr=&id=3EQdUbkOVGIC&oi=fnd&pg=PP1&dq=libro+programacion+orientada+a+objetos&ots=03hVuDzzPO&sig=HZ7yuKarcAYwR0fRdc-3U1SreYU#v=onepage&q=libro%20programacion%20orientada%20a%20objetos&f=false)

<http://php.net/manual/es/index.php>. (s.f.). Obtenido de <http://php.net/manual/es/index.php>

MORENO, F. (1999-2000). *INTRODUCCION A LA OOP VERCION 1.0.0*. Recuperado el 15 de 08 de 2016, de *INTRODUCCION A LA OOP VERCION 1.0.0*: <file:///C:/Users/MARY/Desktop/DOC.%20ITSGA%20TESIS/ANTEPROYECTO/Introduccion%20a%20la%20Programacion%20Orientada%20a%20Objetos.pdf>

## EJECUCIÓN DEL PROYECTO DE INVESTIGACIÓN TESIS.

### 3.1 PLAN GENERAL DE TRABAJO

#### 3.1.1 ANÁLISIS PRELIMINAR DEL PROBLEMA.

La Brigada de Caballería Blindada N° 11 “Galápagos”, a través del departamento de recursos humanos (B-1), en conjunto con las oficinas de recursos humanos (P-1) de cada una de las unidades tiene bajo su responsabilidad el registro, control y manejo de las confrontas del servicio de alimentación del personal de señores Oficiales, Voluntarios y Conscriptos bajo su responsabilidad.

Este proceso se lo realiza de forma manual por medio de tablas realizadas en Excel, las mismas que sirven para el registro de la confronta diaria presentando inconsistencias el manejo de las confrontas en cuanto estas sirven para el control por parte del Sr. Oficial ranchero y se realiza por medio de tickets, mismos son entregados a la persona encargada antes de pasar a consumir los alimentos esta actividad conlleva dificultades como la perdida, duplicación de tickets presentando un problema en el control de esta actividad.

Para lo cual es importante enfocarse en el desarrollo y modernización por medio de sistemas informáticos que nos permitan digitalizar estas actividades de manera ágil y eficiente, que nos permitirá digitalizar este proceso de una manera web, centralizado y administrado por el amaneuse de personal de la Brigada, facilitando y resolviendo problemas actuales y modernización de esta actividad.

#### 3.1.1.2 DEFINICIÓN DEL PROBLEMA.

En la actualidad el registro y control del servicio de alimentación dentro de la Brigada de Caballería Blindada N° 11 “Galápagos”, se lo realiza a través del Departamento de Recursos Humanos (B-1), en conjunto con las oficinas de Recursos Humanos (P-1) de cada una de las unidades que tiene bajo su responsabilidad el registro, control y manejo en el cobro del servicio de alimentación del personal de señores Oficiales, Voluntarios y Conscriptos bajo su responsabilidad.

Este proceso se lo realiza de forma manual, en base a la Directiva N° 04 del 2010 para la administración de la asignación del rancho del personal militar de las Fuerzas Armadas, Instructivo N° FT-DDIE-2015-004-O para la administración del servicio de alimentación en las unidades militares de la fuerza terrestre, publicada en Orden

General N° 03 de fecha 06-ENE-2016, mediante la recopilación de datos proporcionados por el AMANUENSE DE PERSONAL (persona encargada de la recopilación de información referente a la confronta diaria) de las unidades y documentos emitidos por medio de tablas de Excel, enfrentando problemas de dificultad, lentitud, desorganización e inconsistencias de datos así como inconformidad por parte del personal, los cuales imposibilitan la optimización del tiempo al momento de realizar la liquidación mensual de este servicio.

El registro de las confrontas diarias, realizadas diariamente en coordinación con los amanuenses de las Unidades quienes tienen a su responsabilidad esta actividad, y luego de la elaboración de los tickets son entregados al jefe de semana quien se encarga de hacer llegar a cada uno de los usuarios que están registrados en la confronta, los mismos que una vez que se acercan al comedor central entregan el ticket al personal de control siendo este medio de control el que permite o valida el registro de la confronta , exponiéndose a problemas como pérdida de los mismos así como duplicación de estos al ser impresiones comunes y no tener seguridades.

Para lo cual surge la necesidad de implementar un sistema moderno y digital que nos permitirá la digitalización de esta actividad, la cual contará con una interfaz amigable con el usuario, estableciendo políticas de acceso, brindando un detalle de la información en parámetros como consumo cada uno de los usuarios o de las unidades de forma total o parcial partiendo de la directiva vigente que regula el consumo del rancho dentro de las unidades militares, permitiendo un control más óptimo y detallado al momento de realizar el cobro de este servicio, lo cual establecerá una mejor administración así como un control óptimo y moderno de este servicio.

### 3.1.1.3 ESTUDIO DE FACTIBILIDAD.

#### **TÉCNICA.**

Este proyecto tiene la factibilidad técnica de realizarse de una manera web el recurso primordial y herramienta fundamental es la intranet institucional que será el requisito que nos permitirá el control de esta actividad desde diferentes lugares previo a la validación del usuario y seguridades que brindara el presente proyecto.

Para la implementación proyecto se contará con un computador por usuario administrador los mismos cumplen las funciones de amanuenses de las oficinas de Recursos Humanos en cada una de las unidades de la 11-BCB.

Dentro de las herramientas para el desarrollo del presente proyecto contaremos con el lenguaje de programación PHP que es un lenguaje de programación interpretado del

lado del servidor que se caracteriza por su potencia, versatilidad, robustez y modularidad, así como el gestor de base de datos MySQL que es un sistema muy conocido y ampliamente usado por su simplicidad y notable rendimiento así como un alto grado de estabilidad y un rápido desarrollo y está disponible para múltiples plataformas y es posible realizar todos los ejercicios sobre un servidor instalado localmente o, a través de Internet, sobre un servidor remoto.

### **OPERATIVA.**

El presente proyecto contará con el recurso humano que cumple las funciones de amanuenses que cumplirán la función usuarios administradores encargados del registro en cada una de las unidades, de la misma manera se cuenta con el administrador del sistema que será el encargado de la administración, generar reportes y el manejo en sí del presente sistema, quienes ya son parte de la institución y a quienes beneficiará al realizarlo de una forma moderna y digitalizada.

De la misma manera para el desarrollo del presente sistema se contará con la ayuda de personal profesional especializado dentro del área de sistemas que brindará la asesoría necesaria para el desarrollo e implementación.

### **ECONÓMICA**

La parte económica del desarrollo del presente sistema así como la implementación del mismo estará financiada en dos partes que son:

La programación, diseño del sistema por parte del interesado en ejecutar presente proyecto.

La implementación y pruebas del sistema estará financiada por la institución beneficiaria del proyecto.

### **LEGAL**

El desarrollo del presenta proyecto estará normado bajo los siguientes parámetros, mismos que son los que regulan cobro del servicio de alimentación en las unidades y repartos militares:

Directiva N° 04 del 2010 para la administración de la asignación del rancho del personal militar de las Fuerzas Armadas.

Instructivo N° FT-DDIE-2015-004-O para la administración del servicio de alimentación en las unidades militares de la fuerza terrestre, publicada en Orden General N° 03 de fecha 06-ENE-2016.

Los derechos de autoría serán únicos y reservados por parte del programador y diseñador del presente sistema.

El presente sistema utilizará la intranet disponible en la 11-BCB “Galápagos”, precautelando así la confidencialidad de la información de carácter militar.

### 3.1.2 DESARROLLO DEL PROYECTO.

#### 3.1.2.1 PLANIFICACIÓN DEL PROYECTO.

**FASE DE INVESTIGACIÓN BIBLIOGRÁFICA:** Recopilaremos información sobresaliente y actual, se hará un estudio detallado del objeto de estudio y se preparará el entorno de trabajo para el desarrollo del proyecto de tesis empezando por la parte investigativa.

**FASE DE SELECCIÓN DE LA INFORMACIÓN:** Seleccionaremos la información más relevante y la prepararemos para la redacción y organización de contenidos del trabajo escrito de tesis es decir confeccionaremos el marco teórico de referencia.

**FASE DE ANÁLISIS Y DISEÑO:** En esta fase empezaremos a confeccionar la base de datos y a preparar el entorno de desarrollo de la aplicación. Empezaremos a generar ideas de la parte gráfica, diseño, estructura, etc.

### **REQUERIMIENTOS FUNCIONALES**

Persona

Grado de persona

Tipo de persona

Usuario

Tipo de usuario

Unidad

Novedad

Tipo de novedad

Parámetros

Confronta

Confronta general

### **REQUERIMIENTOS NO FUNCIONALES**

Rendimiento

Disponibilidad

Seguridad

Portabilidad

Mantenibilidad

Estabilidad

Interfaces

Usabilidad

**FASE DE DESARROLLO DE SOFTWARE:** Realizaremos el proceso de desarrollo típico de software. Desarrollo de la aplicación web tanto la capa de aplicación como la de acceso a datos, se hará los cambios que se deban hacer en la base de datos antes de empezar la programación pura.

**FASE DE FINALIZACIÓN DEL PROYECTO DE TESIS:** Se realizara las pruebas finales al software, así como la implementación del sistema dentro de la institución beneficiaria y la redacción de documentos pendientes, preparación y corrección de cambios finales de la tesis escrita de grado, impresión de documentos, defensa de presentación de trabajos escritos.

3.1.3 DISEÑO CONCEPTUAL Y LÓGICO DEL SISTEMA.

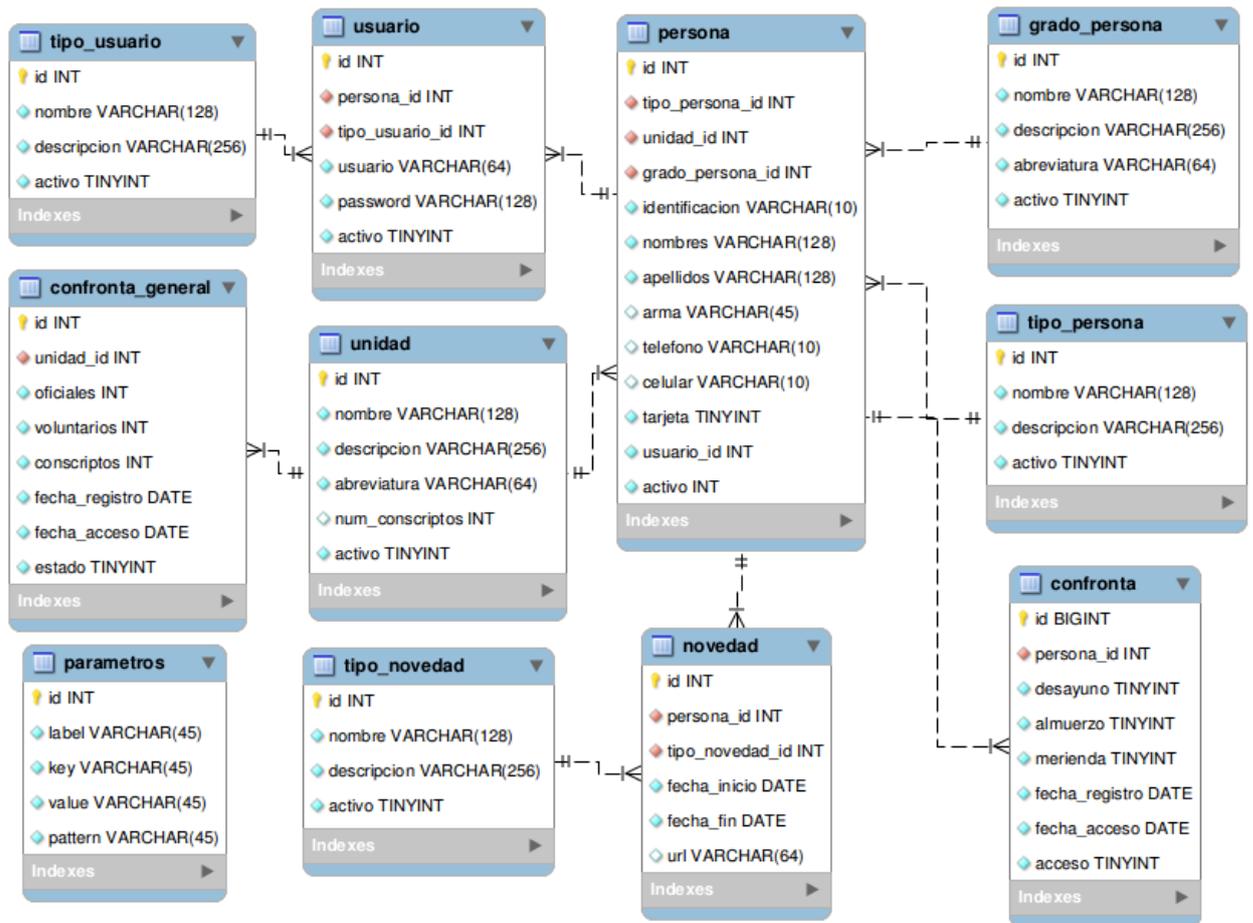


Figura 20 Diseño Conceptual

Fuente: Autor

### 3.2 HIPÓTESIS

El diseño e implementación de un sistema web mejorará el proceso de registro y control del servicio de alimentación.

### 3.3 VARIABLES.

#### 3.3.1 VARIABLE INDEPENDIENTE.

Diseño e implementación de un sistema web.

#### 3.3.2 VARIABLE DEPENDIENTE.

El proceso de registro y control del sistema de alimentación

#### 3.3.3 OPERACIONALIZACIÓN DE LAS VARIABLES.

VARIABLES	CONCEPTUALIZACIÓN	INDICADORES	TÉCNICAS E INSTRUMENTOS
Diseño e implementación	Elaborar un sistema informático que nos ayude	My.SQL. Boopstrap	Entrevista Encuesta

n de un sistema web.	a modernizar el proceso de registro y control, de una manera web que funcionara en la 11-BCB "GALAPAGOS"	NetBeans	
El proceso de registro y control del sistema de alimentación.	Actividad realizada con la finalidad de llevar una administración adecuada del servicio de alimentación.	Confronta diaria Confronta general Reportes mensuales	Entrevista Encuesta

Tabla 1 Operacionalización de variables

**Fuente: Autor**

### 3.4 CRONOGRAMA TENTATIVO

	<b>Nombre Tarea</b>	<b>Duración</b>	<b>Comienzo</b>	<b>Fin</b>	<b>Predecesoras</b>
<b>1</b>	Elaboración del proyecto	30 días	01/11/2016	30/11/2016	
<b>2</b>	Presentación y aprobación del proyecto	31 días	01/12/2016	30/12/2016	
<b>3</b>	Desarrollo e implementación	90 días	01/01/2017	30/03/2017	
<b>4</b>	Elaboración del informe	31 días	01/04/2017	30/04/2017	
<b>5</b>	Presentación y aprobación	31 días	03/05/2017	30/05/2017	

Tabla 2 Cronograma Tentativo

**Fuente: Autor**

### 3.5 RECURSOS

#### 3.5.1 RECURSOS HARDWARE

<b>ORD</b>	<b>DETALLE</b>	<b>CANT</b>	<b>DESCRIPCIÓN</b>
01	Maquina Desarrollo de la tesis	01	Laptop Toshiba I3

02	Impresora	01	Impresión documentación
03	Intranet 11 B.C.B	01	Punto por usuario administrador
04	Pc usuarios administradores	01	PC escritorio por usuario administrador.
05	PC escritorio/servidor	01	PC escritorio
			Procesador Intel ® Core i7 sexta generación.
			Disco duro de 1 Tbt.
			Memoria Ram de 8 Gb
			Motherboard Gigabyte
06	Cable UTP	200 Mts	Cable UTP categoría 7
07	Conectores	10	Rj-45
08	Regulador de voltaje	01	FORZA AVR-1211VA 110 V US
09	Cable gemelo 110 V	30 Mts	Energía eléctrica conexiones.
10	Ponchadora	01	Ponchar cables y conectores.
11	Prueba continuidad internet	01	Verificar conectividad

Tabla 3 Recursos Harware

**Fuente: Autor**

### 3.5.2 RECURSOS SOFTWARE.

ORD	DESCRIPCIÓN	CANT	DETALLE
01	Sistema operativo Windows 8.1	01	Instalación PC.
02	Microsoft Office 2013	01	Instalación PC.
03	Motor de Base de Datos PHP My-SQL	01	Desarrollo sistema
04	Wamp Server 64	01	Desarrollo sistema
05	Proyect 2010	01	Elaboración documentación
06	Visio 2010	01	Elaboración documentación

Tabla 4 Recursos Software

**Fuente: Autor**

### 3.5.3 RECURSOS MATERIALES

<b>ORD</b>	<b>DESCRIPCIÓN</b>	<b>CANT</b>	<b>DETALLE</b>
01	Papel bon formato A-4	01 resma	Impresión documentación
02	Tinta impresión	01	Recarga impresora
03	Cable gemelo 110 V	30 Mts	Energía eléctrica conexiones.

Tabla 5 Recursos Materiales

**Fuente: Autor**

### 3.6 MÉTODOS Y TÉCNICAS.

#### 3.6.1 MÉTODOS

Para el desarrollo de la investigación se aplicara el método científico, que permitirá establecer una secuencia ordenada de acciones que nos llevarán a establecer las conclusiones sobre el uso eficiente de las tecnologías que permitirá el desarrollo ágil del presente sistema.

#### 3.6.2 TÉCNICAS

También se empleará la observación y análisis por parte de los investigadores para hacer deducciones e inducciones sobre el tema de tesis.

Para la aplicación se realizarán entrevistas y encuestas a las personas que utilizarán el sistema y que actualmente realizan los procesos de forma manual, para determinar las necesidades que se deben cubrir.

### 3.7 PRESUPUESTO

<b>ORD</b>	<b>DETALLE</b>	<b>CANTIDAD</b>	<b>PRECIO</b>
01	Libros	02	100
02	PC/ Servidor	01	850
03	Cable internet	100 mts	35
04	Conectores	20	10
05	Asesoría	01	500
06	Hojas	02	10
07	Tinta impresora	01	50
08	Gastos varios		50
<b>TOTAL</b>			<b>\$1100,00</b>

Tabla 6 Presupuesto

**Fuente: Autor**

### 3.8 FUENTE DE FINANCIAMIENTO

El financiamiento en el desarrollo del presente sistema así como la implementación del mismo estará financiado en dos partes que son la parte de programación, diseño del sistema por parte del interesado en ejecutar el sistema, y la segunda parte que consta en la implementación y pruebas del sistema estará financiada por la institución beneficiaria del proyecto.

### 3.9 FECHA DE APROBACIÓN

# **ANEXO V**

# **FOTOGRAFÍAS**

