



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS**

Tema:

APLICACIÓN WEB DE CATÁLOGOS PRIVADOS UTILIZANDO LA
ARQUITECTURA MVC PARA LA EMPRESA TURBOMEKANICS S.A. DE LA
CIUDAD DE AMBATO.

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo la
obtención del título de Ingeniero en Sistemas Computacionales e Informáticos.

SUBLÍNEA DE INVESTIGACIÓN: Aplicación Web.

AUTOR: Edwin Rubén Jiménez Ruiz.

TUTOR: Ing. Mg. Franklin Oswaldo Mayorga Mayorga,

Ambato - Ecuador

Agosto/2017

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo de investigación sobre el tema: “APLICACIÓN WEB DE CATÁLOGOS PRIVADOS UTILIZANDO LA ARQUITECTURA MVC PARA LA EMPRESA TURBOMEKANICS S.A. DE LA CIUDAD DE AMBATO.”, del señor Edwin Rubén Jiménez Ruiz., estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el Art. 16 del Capítulo II, del Reglamento de Graduación para obtener el título terminal de tercer nivel de la Universidad Técnica de Ambato.

Ambato, agosto de 2017

EL TUTOR



Ing. Mg. Franklin Oswaldo Mayorga Mayorga,

AUTORÍA

El presente trabajo de investigación titulado: “APLICACIÓN WEB DE CATÁLOGOS PRIVADOS UTILIZANDO LA ARQUITECTURA MVC PARA LA EMPRESA TURBOMEKANICS S.A. DE LA CIUDAD DE AMBATO.”, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto de 2017



Sr. Edwin Rubén Jiménez Ruiz.

CC: 1804409652

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato, agosto de 2017



Sr. Edwin Rubén Jiménez Ruiz.

CC: 1804409652

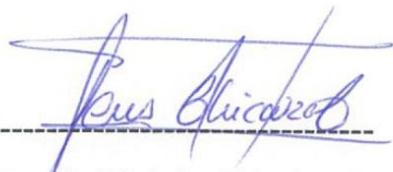
APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Dennis Chicaiza e Ing. Hernán Naranjo, revisó y aprobó el Informe Final del trabajo de graduación titulado “APLICACIÓN WEB DE CATÁLOGOS PRIVADOS UTILIZANDO LA ARQUITECTURA MVC PARA LA EMPRESA TURBOMEKANICS S.A. DE LA CIUDAD DE AMBATO.”, presentado por el señor Edwin Rubén Jiménez Ruiz. de acuerdo al Art. 17 del Reglamento de Graduación para obtener el título Terminal de tercer nivel de la Universidad Técnica de Ambato.

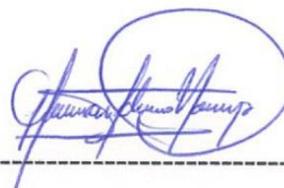


Ing. Mg. Elsa Pilar Urrutia Urrutia

PRESIDENTE DEL TRIBUNAL



Ing. Dennis Vinicio Chicaiza Castillo
DOCENTE CALIFICADOR



Ing. Hernán Fabricio Naranjo Ávalos
DOCENTE CALIFICADOR

DEDICATORIA

Dedico el presente proyecto a mis padres Gladys Ruiz y Roberto Jiménez por saber guiarme en todo momento de la vida por el camino del bien, a mi esposa Diana Paredes quien ha sabido ser mi mano derecha y consejera para la realización de este proyecto, a mis hijas Angy y Viki quienes son el pilar de mi vida y las personas por las cuales saldré adelante, finalmente a mis maestros que supieron brindarme grandes conocimientos en mi etapa académica.

Edwin Rubén Jiménez Ruiz.

AGRADECIMIENTO

Agradezco a mis padres quienes me supieron encaminar por el camino del bien sabiéndome brindar todo su apoyo, a toda mi familia, a mi esposa e hijas por estar a mi lado alentándome para cumplir este gran objetivo. A la empresa Turbomekanics S.A. por abrirme sus puertas y brindarme la información necesaria para la realización de este proyecto de investigación.

A la facultad de Ingeniería en Sistemas, Electrónica e Industrial por haberme brindado la oportunidad de adquirir conocimientos y formarme profesionalmente así como también a todos los docentes que la conforman quienes supieron transmitir sus saberes a mi persona.

A mi tutor de tesis Ing. Franklin Mayorga por haberme guiado de manera profesional sin ningún interés. A todos ustedes de corazón muchísimas gracias.

Edwin Rubén Jiménez Ruiz.

ÍNDICE DE CONTENIDOS

PÁGINAS PRELIMINARES

Portada	i
Aprobación del tutor	ii
Autoría	iii
Aprobación de la comisión calificadora	v
Dedicatoria.....	vi
Agradecimiento.....	vii
Índice de contenidos	viii
Índice de figuras.....	xii
Índice de tablas	xiv
Resumen ejecutivo.....	xvi
Summary.....	xvii
Introducción	xviii

CAPÍTULO I EL PROBLEMA

1.1	Tema.....	1
1.2	Planteamiento del problema	1
1.3	Delimitación.....	2
1.4	Justificación.....	3
1.5	Objetivos	4
1.5.1	Objetivo General	4
1.5.2	Objetivos Específicos.....	4

CAPÍTULO II MARCO TEÓRICO

2.1	Antecedentes investigativos	5
2.2	Fundamentación teórica	6
2.2.1	Catálogo comercial.....	6
2.2.2	Catálogo en línea.....	6
2.2.3	Aplicación web.....	6

2.2.3.1	Servidor web	7
2.2.3.2	Base de datos.....	7
2.2.4	Arquitectura de software	8
2.2.4.1	MVC.....	8
2.2.5	Framework	8
2.2.6	Metodología de desarrollo de software	9
2.3	Propuesta de solución.....	9

CAPÍTULO III METODOLOGÍA

3.1	Modalidad de la investigación	10
3.2	Recolección de información.....	11
3.3	Procesamiento y análisis de datos	11
3.4	Desarrollo del proyecto	11
3.4.1	Metodología de desarrollo.....	12
3.4.2	Justificación de la metodología.....	14

CAPÍTULO IV DESARROLLO DE LA PROPUESTA

4.1	Concepción.....	18
4.1.1	Levantamiento de requerimientos	18
4.1.1.1	Realización de entrevistas	18
4.1.1.2	Observación de procesos.....	21
4.1.1.3	Análisis de procesos de marketing.....	22
4.1.2	Definición de ámbitos del sistema	24
4.1.2.1	Requerimientos funcionales	24
4.1.2.2	Requerimientos no funcionales	25
4.1.3	Establecer requerimientos de hardware y software.....	25
4.1.3.1	Lenguaje de programación	25
4.1.3.2	Base de datos.....	28
4.1.4	Análisis de la arquitectura MVC.....	29
4.1.4.1	Elementos	30
4.1.4.2	Funcionamiento.....	31
4.1.4.3	Ventajas y Desventajas	32
4.1.5	Definición de casos de uso	33

4.1.5.1	Diagrama caso de uso Público	34
4.1.5.2	Diagrama caso de uso Administrador	34
4.1.5.3	Diagrama caso de uso Cliente	40
4.2	Elaboración	46
4.2.1	Diagramas de secuencia	46
4.2.1.1	Diagramas de secuencia Público	46
4.2.1.2	Diagrama caso de uso Administrador	47
4.2.1.3	Diagrama caso de uso Cliente	52
4.2.2	Diseño de la base de datos	57
4.2.2.1	Diseño lógico	57
4.2.2.2	Diseño físico.....	58
4.2.3	Diccionario de datos.....	59
4.2.4	Diseño previo de la interfaz de usuario.....	63
4.2.4.1	Interfaz principal usuarios anónimos	63
4.2.4.2	Interfaz de inicio de sesión.....	64
4.2.4.3	Interfaz página principal	65
4.2.4.4	Interfaz de perfil de usuario	66
4.2.4.5	Interfaz ingreso de usuarios	67
4.2.4.6	Interfaz eliminar usuario	67
4.2.4.7	Interfaz productos.....	68
4.2.4.8	Interfaz información de producto.....	69
4.2.4.9	Interfaz carrito de compras.....	69
4.3	Construcción	70
4.3.1	Implementación del servidor web	70
4.3.2	Implementación de la base de datos.....	72
4.3.3	Implementación del código fuente de la aplicación.....	74
4.3.3.1	Configuración de entorno.....	74
4.3.3.2	Implementación de la interfaz gráfica.....	75
4.3.3.3	Implementación de la arquitectura MVC.....	76
4.4	Transición.....	85
4.4.1	Pruebas de caja blanca	85
4.4.2	Prueba de caja negra.....	88

4.5	Implantación.....	91
4.5.1	Instalación y configuración del servidor web	92
4.5.2	Configuración de PHP y MySQL	92
4.5.3	Migración y replicación de datos	93
4.5.4	Pruebas de funcionamiento	100
4.5.5	Capacitación al personal	100

CAPÍTULO V
CONCLUSIONES Y RECOMENDACIONES

5.1	Conclusiones	101
5.2	Recomendaciones.....	102
	BIBLIOGRAFÍA	103
	GLOSARIO DE TÉRMINOS Y ACRÓNIMOS.....	106
	ANEXOS Y APENDICES	108

ÍNDICE DE FIGURAS

Figura 2.1: Funcionamiento aplicaciones web	7
Figura 3.1: Ventajas y desventajas metodologías tradicionales	13
Figura 3.2: Ventajas y desventajas metodologías ágiles	14
Figura 3.3: Cuadro comparativo Metodologías	15
Figura 3.4: Cuadro comparativo Metodologías	15
Figura 4.1: Diagrama de proceso- precio del producto.....	22
Figura 4.2: Principal catálogo actual	23
Figura 4.3: Lista de turbos por marca.....	23
Figura 4.4: Turbo seleccionado	24
Figura 4.5: Funcionamiento de PHP	27
Figura 4.6: Arquitectura MVC	31
Figura 4.7: Casos de uso Público	34
Figura 4.8: Casos de uso Administrador	35
Figura 4.9: Casos de uso Cliente	41
Figura 4.10: Diagrama de secuencia login en el sistema	46
Figura 4.11: Diagrama de secuencia logout en el sistema	47
Figura 4.12: Diagrama de secuencia editar perfil propio.....	47
Figura 4.13: Diagrama de secuencia visualizar imagen de artículo	48
Figura 4.14: Diagrama de secuencia visualizar información de articulo	48
Figura 4.15: Diagrama de secuencia visualizar información de cliente	49
Figura 4.16: Diagrama de secuencia visualizar categoría de articulo	49
Figura 4.17: Diagrama de secuencia visualizar información de grupo	50
Figura 4.18: Diagrama de secuencia visualizar información de usuario	50
Figura 4.19: Diagrama de secuencia agregar usuario	51
Figura 4.20: Diagrama de secuencia activar/desactivar usuario	51
Figura 4.21: Diagrama de secuencia eliminar usuario	52
Figura 4.22: Diagrama de secuencia filtrar artículo por precio.....	52
Figura 4.23: Diagrama de secuencia filtrar artículo por categoría	53
Figura 4.24: Diagrama de secuencia añadir artículos al carrito de compras	53
Figura 4.25: Diagrama de secuencia visualizar contenido carrito de compras.....	54
Figura 4.26: Diagrama de secuencia editar contenido del carrito de compras	54
Figura 4.27: Diagrama de secuencia quitar artículos del carrito de compras	55
Figura 4.28: Diagrama de secuencia vaciar carrito de compras.....	55
Figura 4.29: Diagrama de secuencia realizar compra	56
Figura 4.30: Diseño lógico de base de datos	57
Figura 4.31: Diseño físico de base de datos	58
Figura 4.32: Interfaz principal usuarios anónimos	64
Figura 4.33: Interfaz de inicio de sesión.....	64
Figura 4.34: Interfaz página principal	65
Figura 4.35: Interfaz perfil de usuario	66
Figura 4.36: Interfaz ingreso de usuario	67
Figura 4.37: Interfaz eliminar usuario	67
Figura 4.38: Interfaz productos	68

Figura 4.39: Interfaz información del producto	69
Figura 4.40: Interfaz información del producto	70
Figura 4.41: Esquema del servidor WAMP	71
Figura 4.42: Generación de sql de base de datos	72
Figura 4.43: Implementación de nuevo esquema	73
Figura 4.44: Abrir script sql	73
Figura 4.45: Abrir script sql	73
Figura 4.46: Esquema de bootstrap	76
Figura 4.47: Grafo prueba del camino básico	87
Figura 4.48: Prueba de caja negra: interfaz entendible	89
Figura 4.49: Prueba de caja negra: campo requerido	89
Figura 4.50: Prueba de caja negra: tipo de datos incorrecto	90
Figura 4.51: Prueba de caja negra: inconsistencia de contraseñas	90
Figura 4.52: Prueba de caja negra: opción no seleccionada	91
Figura 4.53: Prueba de caja negra: registro existente	91
Figura 4.54: SGBD usado actualmente	93
Figura 4.55: Esquema de migración	94
Figura 4.56: Interfaz inicio de sesión programa migración de datos	95
Figura 4.57: Interfaz conexión a bases de datos	96
Figura 4.58: Interfaz migración	96
Figura 4.59: Tablas de catálogo de datos maestro	97
Figura 4.60: Tabla con la ruta del web service	97
Figura 4.61: Programa para migración de datos funcionando	98
Figura 4.62: Esquema de replicación	99
Figura 4.63: Triggers para la replicación de datos	99

ÍNDICE DE TABLAS

Tabla 3.1: Cuadro comparativo de metodologías de desarrollo	13
Tabla 4.1: Entrevista al administrador de sistemas.....	19
Tabla 4.2: Entrevista al ayudante de sistemas	20
Tabla 4.3: Guía de observación	21
Tabla 4.4: Tabla comparativa de lenguajes de programación web	26
Tabla 4.5: Tabla comparativa de base de datos	29
Tabla 4.6: Tabla ventajas y desventajas de MVC.....	33
Tabla 4.7: Especificación caso de uso: login en el sistema.....	34
Tabla 4.8: Especificación caso de uso: logout en el sistema.....	35
Tabla 4.9: Especificación caso de uso: Editar perfil.....	36
Tabla 4.10: Especificación caso de uso: Visualizar información de artículo	36
Tabla 4.11: Especificación caso de uso: Visualizar imagen de artículo.....	37
Tabla 4.12: Especificación caso de uso: Visualizar categoría de artículo.....	37
Tabla 4.13: Especificación caso de uso: Visualizar información de cliente	38
Tabla 4.14: Especificación caso de uso: Visualizar información de grupo.....	38
Tabla 4.15: Especificación caso de uso: Agregar información de usuario.....	39
Tabla 4.16: Especificación caso de uso: Visualizar información de usuario	39
Tabla 4.17: Especificación caso de uso: Eliminar información de usuario	40
Tabla 4.18: Especificación caso de uso: Editar información de usuario	40
Tabla 4.19: Especificación caso de uso: Filtrar artículo por precio	42
Tabla 4.20: Especificación caso de uso: Filtrar artículo por categoría.....	42
Tabla 4.21: Especificación caso de uso: Añadir artículo al carrito de compras	43
Tabla 4.22: Especificación caso de uso: Visualizar contenido carrito de compras	43
Tabla 4.23: Especificación caso de uso: Editar contenido de carrito de compras	44
Tabla 4.24: Especificación caso de uso: Quitar artículo del carrito de compras	44
Tabla 4.25: Especificación caso de uso: Vaciar carrito de compras	45
Tabla 4.26: Especificación caso de uso: Realizar compra	45
Tabla 4.27: Diccionario de datos tabla articulos.....	59
Tabla 4.28: Diccionario de datos tabla clasesarticulos	59
Tabla 4.29: Diccionario de datos tabla clientes	60
Tabla 4.30: Diccionario de datos tabla colorarticulos.....	60
Tabla 4.31: Diccionario de datos tabla fotoUsers	60
Tabla 4.32: Diccionario de datos tabla groups	61
Tabla 4.33: Diccionario de datos tabla imagenes	61
Tabla 4.34: Diccionario de datos tabla listaprecios	61
Tabla 4.35: Diccionario de datos tabla menus	62
Tabla 4.36: Diccionario de datos tabla menugrups.....	62
Tabla 4.37: Diccionario de datos tabla precios.....	62
Tabla 4.38: Diccionario de datos tabla subclasesarticulos.....	63
Tabla 4.39: Diccionario de datos tabla users	63
Tabla 4.40: Prueba de camino básico de función agregar usuario	86
Tabla 4.41: Caminos básicos.....	87
Tabla 4.42: Comprobación de camino básico.....	88

Tabla 4.43: Plan de Implantación	92
Tabla 4.44: Pruebas finales de funcionamiento	100
Tabla 4.45: Plan de capacitación	100

RESUMEN EJECUTIVO

Las empresas en la actualidad han notado que el uso de la tecnología les permite obtener ventajas competitivas en la promoción de sus productos y servicios; mejorando la relación con sus clientes y facilitando el proceso de venta de sus productos.

Turbomekanics S.A., es una empresa dedicada a la importación, distribución, mantenimiento y reparación de autopartes y turbocompresores con sus respectivos repuestos.

La presente investigación “Aplicación web de catálogos privados utilizando la arquitectura MVC para la empresa Turbomekanics S.A. de la ciudad de Ambato.” surge de la necesidad de la empresa de poseer un catálogo web actualizado que brinde la información necesaria que requiere conocer el cliente.

Es por ello que se propone el desarrollo de una aplicación web con el fin de modernizar y mejorar el proceso de administración de los catálogos que posee la empresa y que permita mantener la información del stock de productos de manera actualizada y ordenada.

Mediante esta web los clientes de la empresa poseerán de un medio de consulta de los productos que oferta la empresa con una interfaz de fácil uso.

El proyecto se desarrolla tomando como base los lineamientos de la arquitectura de programación MVC y utilizando nuevas tecnologías, métodos y herramientas dándole importancia al análisis de requerimientos para obtener un sistema eficiente y capaz de cumplir con lo que la empresa necesita.

SUMMARY

Companies today have noticed that the use of technology allows them to gain competitive advantages in the promotion of their products and services; Improving the relationship with its customers and facilitating the process of selling their products.

Turbomekanics S.A., is a company dedicated to the import, distribution, maintenance and repair of auto parts and turbochargers with their respective spare parts.

The present research “Aplicación web de catálogos privados utilizando la arquitectura MVC para la empresa Turbomekanics S.A. de la ciudad de Ambato.” It arises from the need of the company to have an updated web catalog that provides the necessary information that requires knowing the client.

It is for this reason that the development of a web application is proposed in order to modernize and improve the process of administration of the catalogs that the company owns and that allows to keep the stock information of fashion products updated and orderly.

By means of this web the clients of the company offer a medium of consultation of the products that offer the company with an interface of easy use.

The project is developed based on the guidelines of the MVC programming architecture and the use of new technologies, methods and tools give importance to the analysis of requirements to achieve an efficient system and able to meet what the company needs.

INTRODUCCIÓN

El presente trabajo de investigación con el tema: “APLICACIÓN WEB DE CATÁLOGOS PRIVADOS UTILIZANDO LA ARQUITECTURA MVC PARA LA EMPRESA TURBOMEKANICS S.A. DE LA CIUDAD DE AMBATO.” Para mejor comprensión se ha dividido en los capítulos que se detallan a continuación:

CAPÍTULO I “EL PROBLEMA”, se define el problema, la justificación y los objetivos por los cuales se realiza la investigación.

CAPÍTULO II “MARCO TEÓRICO”, comprende los antecedentes investigativos que permiten comprender de forma clara el problema y la fundamentación teórica que sirven como sustento para la investigación.

CAPÍTULO III “METODOLOGÍA”, se detalla la metodología y las modalidades de investigación utilizadas para la recolección y análisis de la información. Además se describe brevemente las actividades que se desarrollaran en base a los objetivos para conseguir el resultado propuesto.

CAPÍTULO IV “DESARROLLO DE LA PROPUESTA”, se detalla de manera clara todo el desarrollo de la propuesta de solución siguiendo el ciclo de vida de software y documentando este proceso por medio de diagramas, partiendo desde el levantamiento de requerimientos y finalizando en la implementación de la aplicación.

CAPÍTULO V “CONCLUSIONES Y RECOMENDACIONES”, contiene las conclusiones y recomendaciones a las que ha llegado el investigador después del desarrollo del proyecto.

Por último se incluyen la bibliografía consultada y los anexos con el glosario de términos, los instrumentos de recolección de información y los manuales técnico y de usuario.

CAPÍTULO I

EL PROBLEMA

1.1 Tema

Aplicación web de catálogos privados utilizando la arquitectura MVC para la empresa Turbomekanics S.A. de la ciudad de Ambato.

1.2 Planteamiento del problema

En la actualidad la tecnología ha permitido automatizar muchas de las actividades del ser humano incluidas las económicas y de negocios. Las empresas comerciales en busca de ser más competitivas y brindar mejores servicios implementan aplicaciones web en las que mantienen el catálogo de los productos que ofertan, lo que facilita publicitar la información de una manera adecuada a sus potenciales clientes.

En el Ecuador la industria automotriz tiene un papel muy importante en la economía, siendo reconocida por sus altos estándares de calidad [1]; este sector lo integran tanto firmas ensambladoras como proveedores de partes y piezas. Existen en el país contadas empresas de renombre que se dedican a la venta de turbocompresores para motor; muchas de ellas poseen catálogos como una herramienta de marketing directo que les permite vender de forma inmediata sus productos. Por ejemplo podemos citar a Garner Espinosa C.A, empresa quiteña de gran reconocimiento internacional que posee sucursales en las principales ciudades del país y con una larga trayectoria en la distribución de productos entre ellos turbos para motor; la cual cuenta con una página web en la que exponen sus productos en diferentes categorías. [2]. De igual manera la

empresa Turbo Master es un proveedor con más de 20 años de experiencia profesional en la venta y reparación de turbos posee un catálogo con los productos además de un catálogo de fallas con los posibles inconvenientes que podrían tener los productos. [3]

La ciudad de Ambato en los últimos años ha adquirido un alto crecimiento económico, lo que ha obligado a las empresas comercializadoras a buscar nuevos métodos para brindar sus servicios y exponer sus productos a la comunidad; es por esto que adoptan el uso de la tecnología para su uso cotidiano. En el ámbito de venta de turbocompresores para motores destaca en la ciudad la empresa TurboTech, la cual posee una web propia con información de sus productos. [4]

Turbomekanics S.A. es una empresa ambateña constituida el 21 de noviembre de 2012 y cuyas actividades dan inicio desde el 15 de octubre de 2013. Su principal actividad es la importación y distribución de autopartes, turbocompresores con sus piezas y cartuchos para compresores en gran variedad de marcas; también brindan servicio de mantenimiento, reparación, ensamblado y empacado de turbocompresores. La empresa posee una página web con un álbum fotográfico a manera de catálogo, sin embargo este se encuentra desactualizado y no muestra la información del producto que el cliente necesita conocer (precio, disponibilidad, etc.). Debido a esto existen problemas de logística de importaciones generando un exceso de inventario y poca rotación del mismo.

1.3 Delimitación

Área académica: Software.

Línea de investigación: Desarrollo de software.

Sublínea de Investigación: Aplicación Web.

Delimitación Espacial: La investigación se desarrollará en la empresa Turbomekanics ubicada en la avenida Real Audiencia entre La Rotonda y San Pedro Alejandrino, sector de la cárcel de la ciudad de Ambato.

Delimitación Temporal: La investigación tendrá una duración de 6 meses a partir de la aprobación del Honorable Consejo Directivo de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

1.4 Justificación

El comercio es una actividad ancestral que con el tiempo ha evolucionado para adaptarse a las necesidades de los clientes en un mundo cada vez más globalizado. En primera instancia la aparición de las ventas por catálogo trajo consigo el impulso a las ventas al por mayor; y en la actualidad, con la llegada del internet a los hogares, incluso las pequeñas empresas han modificado sus estrategias comerciales, beneficiándose de tener imagen corporativa en línea y la facilidad de venta inmediata de sus productos.

La empresa Turbomekanics es una empresa relativamente nueva que busca ser pionera en la venta de turbocompresores. Para brindar un servicio mucho más confiable a sus compradores requiere contar con catálogos electrónicos privados de acceso exclusivo mediante autenticación, requerimiento que no ha sido implementado en su actual página web.

Mediante la implementación de una aplicación web permitirá la administración adecuada de los productos en existencia para mostrarlos a los clientes de una manera rápida, cómoda y de acuerdo a sus propios requerimientos.

De igual manera proporcionará mayor agilidad para la realización de las compras al presentar a los usuarios cada uno de los productos y servicios de una forma dinámica y con un correcto diseño.

Como beneficiarios directos se encuentran los clientes, quienes por medio de la automatización obtendrán información precisa y confiable de los productos de la empresa. De igual manera, beneficiará al personal al poder contar con una herramienta que le permita gestionar de una mejor manera los productos que ofrece la empresa.

Todo esto aportará a mejorar el prestigio de la institución en el mercado automotriz, sector comercial cada vez más competitivo.

1.5 Objetivos

1.5.1 Objetivo General

Implementar una aplicación web para el manejo de catálogos privados de productos en la empresa Turbomekanics.

1.5.2 Objetivos Específicos

- Realizar un análisis de los procesos de marketing y catálogos que utilizan actualmente en la empresa.
- Determinar las herramientas adecuadas para el desarrollo del proyecto de acuerdo con los recursos y requerimientos de la empresa
- Realizar el diseño de la base de datos e interfaz de la aplicación web de catálogos privados.
- Desarrollar la aplicación para el manejo adecuado de catálogos privados dentro de la empresa Turbomekanics.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes investigativos

Se ha determinado que existen trabajos con variables afines a este estudio y se los establece como antecedentes investigativos que a continuación se describen:

En la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, existe una tesis denominada “Diseño e implantación de una sitio web para el control de pedidos de la empresa Comercial Yolanda Salazar Cía. Ltda.” Realizada por Nancy Alicia Moyolema Morales. Su principal conclusión es la existencia de un sitio web permite mejorar la calidad de atención a los clientes que adquieren los productos y que se encuentran en otras ciudades del país. El catálogo de productos está a disposición de todas las personas que ingresen a nuestro sitio web, describiendo en forma detallada cada uno de ellos. [5]

Por otra parte Ivónn Vanessa Villacís Palacios en su proyecto “Diseño e implementación de una aplicación Web, para realizar pedidos on - line en la empresa Disama Cía. Ltda.” concluye que la implementación del sistema ha permitido mejorar la calidad de servicio al cliente, brindándole un espacio en donde enviar sus sugerencias. [6]

En lo que respecta a fuentes bibliográficas de Internet se ha encontrado los siguientes trabajos de tesis:

En el repositorio de la Escuela Politécnica Nacional, Facultad de Ingeniería en Sistemas, se encuentra la tesis denominada “Desarrollo de un sistema web ventas por catálogo para Agroimzoo Cía. Ltda.” realizada por Paul Alejandro Aguayo Eldredge y David Patricio Salcedo Mena. Quienes concluyen que la arquitectura Modelo-Vista-Controlador (MVC) aplicada en el desarrollo del sistema web, permitió escribir código mucho más ordenado de tal manera que las clases y sus componentes tengan una alta cohesión y un bajo acoplamiento. Al utilizar un framework de desarrollo orientado a una arquitectura como MVC, el equipo de desarrollo puede enfocarse en las tareas propias de la lógica del negocio más no en tareas triviales tales como conexión a la base de datos. [7]

2.2 Fundamentación teórica

2.2.1 Catálogo comercial

“Un catálogo comercial es, básicamente, una herramienta para mostrar en detalle la oferta de productos y servicios de una empresa.” [8]. Por lo general contiene:

- Fotografías del producto con sus accesorios.
- Planos, diagramas, esquemas de estructura, funcionamiento o montaje.
- Explicación técnica de sus características.
- Información comercial: precio, forma de pago, stock.

2.2.2 Catálogo en línea

“Presentación electrónica (en internet) de productos que tradicionalmente se mostraban mediante catálogos impresos” [9]. Su ventaja es que estos se pueden adaptar a las condiciones cambiantes de mercado con respecto a precios, productos y promociones cada día o incluso cada hora.

2.2.3 Aplicación web

“Las aplicaciones web son aplicaciones a las que se accede mediante un navegador y están alojadas en servidores dentro de una Intranet o en Internet.” [10].

Es decir es una aplicación codificada en un lenguaje soportado por el navegador el cual sirve para ejecutarlo.

Este tipo de aplicaciones tienen ventajas sobre las de escritorios debido a que se pueden ejecutar en cualquier equipo informático con cualquier sistema operativo que tenga conexión a internet o intranet, no necesita instalación únicamente usa el navegador, no se consume espacio en el computador (todo se encuentra almacenado en el servidor), la información se encuentra accesible desde cualquier lugar del mundo y a cualquier hora.

Las aplicaciones web funcionan de la siguiente manera:



Figura 2.1: Funcionamiento aplicaciones web
Fuente: [11]

2.2.3.1 Servidor web

“Los servidores web son los encargados de recibir las peticiones referidas a páginas o elementos de la web a través del protocolo http o https y de devolver el resultado de la petición, que suele ser un recurso alojado en el servidor.” [12]

Es un programa que responde adecuadamente a las solicitudes de los usuarios mediante el protocolo HTTP para traducirlos en forma legible en el navegador.

2.2.3.2 Base de datos

“Una base de datos (BD) es un conjunto de datos estructurados apropiadamente y relacionados entre sí (como, por ejemplo, nuestra lista de discos).” [13]

Las bases de datos contienen información que representan a las entidades con sus propias interrelaciones y proporcionan la infraestructura para la toma de decisiones y pueden ser usados por distintos usuarios de forma compartida.

2.2.4 Arquitectura de software

“Entendemos por Arquitectura Software la representación de alto nivel de la estructura de un sistema o aplicación, que describe las partes que la integran, las interacciones entre ellas, los patrones que supervisan su composición, y las restricciones a la hora de aplicar patrones” [14]

En otras palabras la arquitectura define la organización y estructura fundamental que tiene aplicación de software para que cumpla con criterios de calidad establecidos en los requerimientos como facilidad de uso, desempeño, seguridad, robustez, entre otros.

La arquitectura tiene entre sus objetivos permitir:

- Planificar la evolución de la aplicación para establecer posibles cambios.
- Reutilizar el código para problemas similares.
- Manejar y comprender como está estructurada una aplicación.

2.2.4.1 MVC

“El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.” [15]

2.2.5 Framework

“El framework es una interfaz entre el sistema operativo y la aplicación, que hace funcionar a las aplicaciones en un entorno seguro y fiable, independientemente del sistema operativo y del procesador” [16]

Un framework web constituye todas las herramientas y prácticas que permiten la creación de sitios web dinámicos, aplicaciones web y web services (servicios web) enfocado a una arquitectura específica; aliviando la tarea mediante la reutilización de código y conectividad de componentes. Generalmente constan de lenguajes de programación y bibliotecas de programación.

2.2.6 Metodología de desarrollo de software

“Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo.” [17]

La metodología representa el proceso sistemáticamente completo para gestionar, administrar e implementar un proyecto de software definiendo actividades, prácticas y técnicas recomendadas desde el surgimiento de la idea hasta cumplir el objetivo para el cual ha sido creado.

2.3 Propuesta de solución

La presente investigación propone la implementación de una aplicación web de catálogos privados para la empresa Turbomekanics S.A. que permita a los usuarios interesados en adquirir un producto, el acceso a un catálogo de los productos comercializados por la empresa, de acuerdo a sus requerimientos previa autenticación en la aplicación. Además, la aplicación permitirá al personal la administración de la información de los productos almacenados en la base de datos.

CAPÍTULO III

METODOLOGÍA

3.1 Modalidad de la investigación

El presente proyecto es de tipo investigación y desarrollo (I+D) ya que se buscará la implementación de una aplicación web de catálogos privados utilizando la arquitectura MVC en la empresa Turbomekanics S.A. poniendo en práctica los conocimientos adquiridos en la investigación aplicada.

Investigación de campo

Se acudirá a la empresa y se tendrá contacto con los involucrados (personal y clientes de la empresa) para obtener información por medio de técnicas e instrumentos de recolección de datos a fin de analizar las respectivas causas y efectos del problema.

Investigación bibliográfica – documental

Se realizará una investigación bibliográfica – documental con el propósito de obtener información detallada con respecto al problema proveniente de libros, artículos del internet, libros virtuales, proyectos de tesis de grado, informes, así como documentación propia de la empresa; información que servirá como sustento teórico para la investigación.

3.2 Recolección de información

Se recolectará información teniendo contacto directo con el personal de la empresa Turbomekanics S.A., usando la observación documental y de procesos para posteriormente analizar estos datos y establecer las respectivas conclusiones.

La información secundaria se la obtendrá de las fuentes bibliográficas consultadas.

3.3 Procesamiento y análisis de datos

Para la recolección, procesamiento y análisis de la información se aplicará los siguientes procedimientos:

- Elaboración del instrumento de recolección de datos
- Aplicación de instrumento de recolección de información.
- Análisis cualitativo de la información.
- Organización de la información obtenida
- Interpretación de los resultados y establecimiento de las respectivas conclusiones.

3.4 Desarrollo del proyecto

Actividades a realizar para cumplir cada uno de los objetivos específicos:

- Realizar un análisis de los procesos de marketing y catálogos que utilizan actualmente en la empresa.
 - Estudio de los catálogos e información actual que posee la empresa.
- Determinar las herramientas adecuadas para el desarrollo del proyecto de acuerdo con los recursos y requerimientos de la empresa
 - Análisis comparativo y elección de las herramientas para la realización del sistema.
- Analizar la arquitectura MVC a fin de aplicarla en el proyecto propuesto.

- Análisis de la arquitectura MVC, haciendo énfasis en sus ventajas y desventajas.
- Realizar el diseño de la base de datos e interfaz de la aplicación web de catálogos privados.
 - Diseño de la base de datos con su respectivo diccionario de datos.
 - Diseño de las interfaces de entrada y salida de datos.
- Desarrollar la aplicación para el manejo adecuado de catálogos privados dentro de la empresa Turbomekanics.
 - Creación de la base de datos.
 - Codificación de la aplicación.
- Implementar la aplicación, integrándola al sistema de inventarios de la empresa Turbomekanics.
 - Implementación de la aplicación.
 - Realización de pruebas de caja blanca y de caja negra para comprobar el correcto funcionamiento.
 - Realización de la documentación (manual de usuario y manual técnico)

3.4.1 Metodología de desarrollo

“Una metodología de desarrollo es una recopilación de técnicas y procedimientos estructurados en fases para la producción de productos de software de manera eficaz y englobando todo el ciclo de vida del mismo” [18]

Para el desarrollo de un sistema web es importante escoger una metodología de desarrollo adecuada para que el producto sea estable, seguro y que cumpla con las necesidades requeridas.

Existen gran cantidad de propuestas metodológicas para el desarrollo de un software, estas tienen un impacto importante en el éxito del producto final ya que guían las actividades que llevarán a las metas trazadas y el cumplimiento de las necesidades definidas al inicio del proyecto.

Escoger una metodología adecuada ya sea tradicional o ágil, aumenta la calidad del proyecto y maximiza el potencial en el uso de recursos y tiempo. A continuación se

muestra un cuadro comparativo entre metodologías tradicionales versus metodologías ágiles:

Metodologías tradicionales	Metodologías ágiles
Resistencia a cambios.	Preparada para cambios.
Arquitectura se define desde el inicio del proyecto	Mejora arquitectura a los largo del proyecto
Procesos controlados con políticas y normas	Menos control en los procesos
El cliente interactúa con el desarrollador por medio de reuniones	El cliente es parte del equipo de desarrollo
Cualquier tipo de proyectos sobre todo grandes y grupos de trabajo grandes	Orientado a proyectos pequeños y de corta duración

Tabla 3.1: Cuadro comparativo de metodologías de desarrollo
Elaborado por: Rubén Jiménez – Investigador

Metodologías tradicionales

Las propuestas tradicionales establecen rigurosamente las actividades en el proceso de desarrollo y se centran en la realización exhaustiva de la documentación de todo el ciclo de vida del proyecto. Las metodologías tradicionales planean procesos que no conlleven cambios a lo largo del proceso.

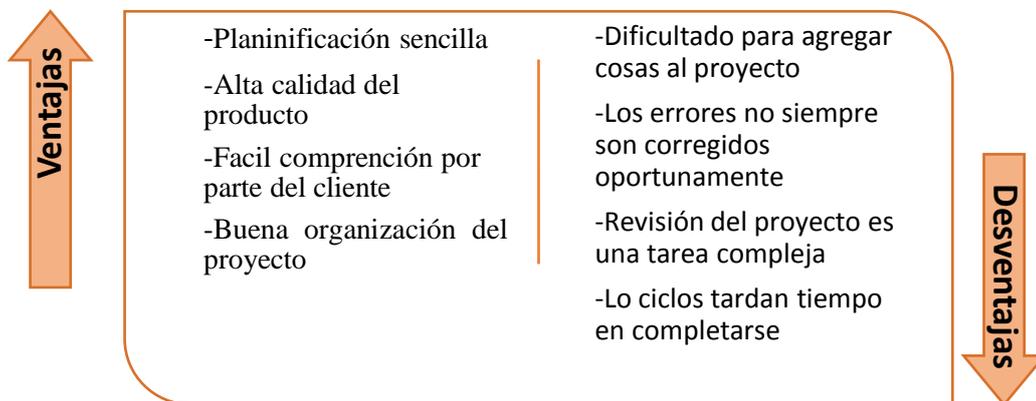


Figura 3.1: Ventajas y desventajas metodologías tradicionales
Elaborado por: Rubén Jiménez – Investigador

Metodologías ágiles

Las propuestas ágiles se centran en la colaboración del cliente por medio de iteraciones cortas haciendo uso del desarrollo incremental. Este tipo de metodologías resaltan la

importancia en la relación con el cliente y la respuesta oportuna a los cambios que este requiera.

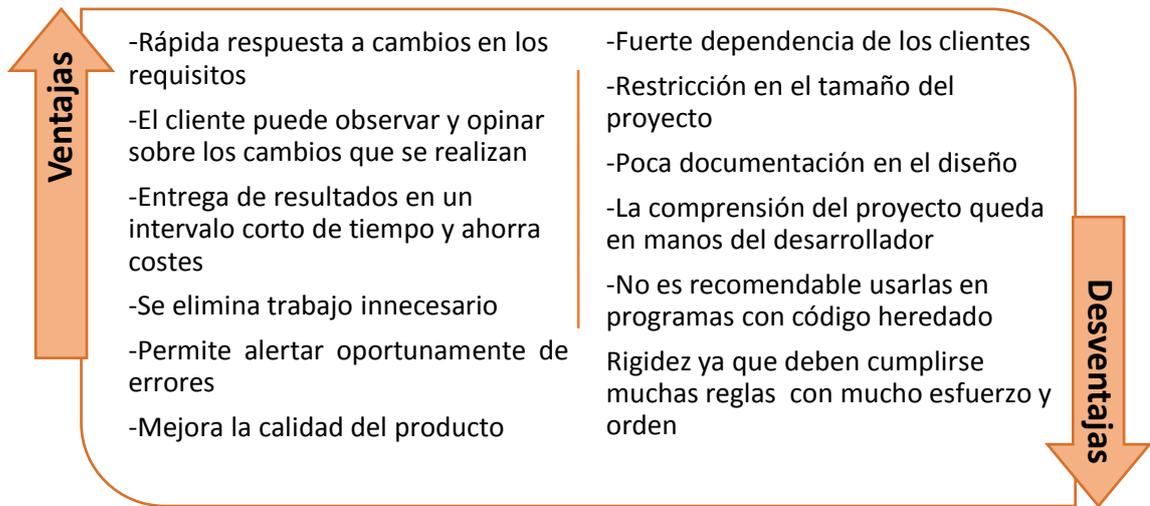


Figura 3.2: Ventajas y desventajas metodologías ágiles
Elaborado por: Rubén Jiménez – Investigador

Las metodologías ágiles presentan innumerables ventajas al disminuir la cantidad de documentación y estar orientado al código fuente para generar un producto de calidad con el tiempo y costo estimados, sin embargo no es recomendable usarlas cuando se requieren código heredado y con documentación

3.4.2 Justificación de la metodología

Para la elección de la metodología a utilizarse se restringe el estudio comparativo a las más utilizadas en el desarrollo de software, calificando de manera cualitativa los siguientes ítems:

- Descripción breve de la metodología.
- Relación del cliente con el desarrollador.
- Tipo de software que se realiza con el uso de la metodología.
- Etapas que cumplen el ciclo de vida.
- Características del modelo.

Item	Modelo Lineal	M. Cascada	M. Prototipo	RUP	RAD	Android
Descripción Breve	Ciclo de vida tradicional	Es una extensión del modelo de capas, pero con un reconocimiento de ciclos de retroalimentación entre etapas	Representa aquellos aspectos del software que serán visibles para el cliente, el cual lo evalúa, para así refinar los requisitos del software que se desarrollará.	Proporciona un enfoque disciplinado para la asignación de tareas y responsabilidades dentro de una organización de desarrollo.	Es un MLS pero que enfatiza en un ciclo extremadamente corto el desarrollo de software, convirtiéndose en una versión suya de "alta velocidad".	Android es un sistema operativo basado en Linux diseñado principalmente para dispositivos móviles con pantalla táctil
Relación Cliente-Desarrollador	Estrecha relación con el cliente	Estrecha relación con el cliente	Posee relación con el cliente en su fase de diseño	Estrecha relación con el cliente	no posee gran relación con el cliente	Estrecha relación con el cliente
Tipo de Software (ERP Empresarial, App, Web)	Principalmente Empotrado y Web	Principalmente Empotrado y Web	Principalmente web y móviles	Principalmente web y móviles.	Se recomienda para empresas grandes	-Aplicaciones móviles
Etapas	-Plan Operativo -Especificación de Requerimientos -Especificación Funcional -Diseño -Implementación -Integración -Validación -Mantenimiento	- Análisis de Requerimiento -Diseño -Implementación - Pruebas -Mantenimiento	-Investigación Preliminar -Diseño y Construcción -Evaluación -Modificación -Diseño técnico -Programación y prueba -Operación y Mantenimiento	-Inicio -Elaboración -Construcción - Transición	-Modelado de Gestión -Modelado de datos -Modelado de Procesos - Generación de Aplicaciones - Pruebas y Entrega.	- Aplicaciones - Marco de trabajo de aplicaciones - Bibliotecas - Run time de Android - Núcleo Linux
Característica propia del modelo	Sus requerimientos son específicos esto Permite reducir la detección de errores	Lo mismo del modelo Anterior, con la gran diferencia que puede volver una etapa atrás.	Crea una maqueta, la cual muestra la interfaz de la aplicación (interfaz estática, No procesa Datos)	RUP tiene características esenciales: está dirigido por los Casos de Uso y es iterativo e incremental.	Una de sus características es: -Visualización temprana	Fácil de aprender Gran impacto Implementación de modelos de negocio

Figura 3.3: Cuadro comparativo Metodologías

Fuente: [19]

Item	iOS	XP	Móvil D	Espiral	Evolutivo	Incremental
Descripción Breve	Es un sistema operativo móvil La interfaz de usuario de iOS está basada en el concepto de manipulación directa, usando gestos multitáctiles.	Se basa en el trabajo orientado directamente al objetivo, basándose para esto en las en la velocidad de reacción para la implementación	Modelo de rápida realización. Ideal para proyectos cortos y de alta complejidad.	El modelo incremental es una evolución del modelo de cascada; viene a suplir el problema de no poder retroceder en las fases de desarrollo.	Modelo que busca reemplazar el viejo sistema con uno nuevo que tendrá la propiedad de satisfacer los nuevos requerimientos lo más rápido posible.	El modelo incremental es una unión de las mejores funcionalidades del modelo de cascada y del modelo de prototipos.
Relación Cliente-Desarrollador	Trabaja con los clientes	Trabaja estrechamente con los clientes	Trabaja estrechamente con los clientes	Trabaja con el cliente	Trabaja con el usuario a medida que avanza en el proyecto	Trabaja con el usuario a medida que avanza en el proyecto
Tipo de Software (ERP Empresarial, App, Web)	Dispositivos móviles	Utilizado para el desarrollo de aplicaciones móviles.	Utilizado para el desarrollo de aplicaciones móviles.	Web y app	-Empresarial	-Empresarial
Etapas	La capa del núcleo del sistema operativo La capa de Servicios Principales La capa de Medios -la capa de Cocoa Touch.	-Pruebas -Planificación -Diseño -Codificación	-Exploración -Iniciación -Productización -Estabilización -Prueba y reparación	- Planificación - Análisis de Riesgos - Ingeniería (Construcción del prototipo) - Evaluación por el cliente	-Especificación inicial -Desarrollo del producto -Implementación, uso y evaluación -Versiones del Software -Re-especificaciones	-Análisis -Diseño -Programación -Pruebas
Característica propia del modelo	Los elementos de control consisten de deslizadores y botones. La respuesta a las órdenes del usuario es inmediata y provee de una interfaz fluida.	Se Basa en las UseStories que definen los detalles técnicos detalles de implementación.	Requerimientos son claros ya que el cliente va inmediatamente proporcionando información de lo que quiere.	Básicamente consiste en una serie de ciclos que se repiten en forma de espiral, comenzando desde el centro.	En el modelo evolutivo se asume que los requisitos pueden cambiar en cualquier momento del ciclo de vida y no solo en la etapa de análisis.	Es una aproximación muy parecida a la evolutiva. Se desarrolla el sistema para satisfacer un subconjunto de los requisitos especificados

Figura 3.4: Cuadro comparativo Metodologías

Fuente: [19]

La aplicación que se va a realizar mantiene objetivos claros en los requerimientos del cliente y se enfoca en la utilización de la programación orientada a objetos.

La metodología a ocupar es Proceso Unificado Racional (RUP), metodología estándar muy utilizada en el desarrollo web ya que es adaptable a las necesidades del proyecto.

Igualmente se han tomado en cuenta otras razones para escoger dicha metodología, las cuales se nombran a continuación:

- Es aplicable a todo tipo de proyectos ya sean grandes o pequeños.
- Posee un ciclo de vida incremental lo que reduce riesgos en el desarrollo al obtener versiones del software en cada iteración.
- Existe gran cantidad de documentación acerca de la metodología en tanto en libros como en la web.
- Utiliza el lenguaje de modelado visual UML que es muy utilizado y conocido en la actualidad y esto facilita las tareas de diseño y programación.
- Es una metodología organizada ya que posee un enfoque estructurado de las tareas y responsabilidades en el desarrollo.
- Acceso fácil a herramientas y plantillas para actividades críticas en el desarrollo.
- Permite mejoras en el diseño y reutilización del código.

Fases de la metodología

1. Concepción

1.1. Levantamiento de requerimientos

1.1.1. Realización de entrevistas

1.1.2. Observación de procesos

1.1.3. Análisis de procesos de marketing

1.2. Definición de ámbitos del sistema

1.3. Establecer requerimientos de software

1.4. Análisis de la arquitectura MVC

1.5. Definición de casos de uso

2. Elaboración

2.1. Elaboración de diagramas de secuencia

- 2.2. Diseño del modelo de base de datos
- 2.3. Diseño del diccionario de datos
- 2.4. Diseño previo de las interfaces
- 3. Construcción
 - 3.1. Implementación del servidor web
 - 3.2. Implementación de la base de datos
 - 3.3. Implementación del código fuente de la aplicación
- 4. Transición
 - 4.1. Pruebas de caja blanca
 - 4.2. Pruebas de caja negra
 - 4.3. Implantación

CAPÍTULO IV

DESARROLLO DE LA PROPUESTA

4.1 Concepción

Presenta una visión general del proyecto para entender los requerimientos y servir como base de las fases posteriores.

4.1.1 Levantamiento de requerimientos

Se utilizan como técnicas de levantamiento de requerimientos la entrevista y observación de campo con los cuales ayudarán a determinar qué es lo que la empresa necesita para el mejoramiento de los catálogos existentes.

4.1.1.1 Realización de entrevistas

Se contó con colaboración del personal de la empresa a los que se les realizó una entrevista estructurada con el fin de recolectar información necesaria para la realización de la aplicación (Anexo 1).

Entrevista al administrador de sistemas

Preguntas	Respuestas
<p>1. ¿Cómo se registra actualmente la información de los productos que oferta la empresa?</p>	<p>El inventario general y los clientes se obtienen del sistema Contable Microplus instalado en la empresa, mientras que la información detallada de cada producto se la obtiene de un sistema externo OM. La información se encuentra registrada en la base de datos Sybase Adaptive Server Anywhere versión 9.</p>
<p>2. ¿A qué clientes van dirigidos los productos que ofrece la empresa?</p>	<p>La empresa Turbomekanics S.A posee clientes tanto nacionales como extranjeros, es por ello que es necesario que los clientes puedan acceder a la información de acuerdo al tipo y a las necesidades que poseen.</p>
<p>3. ¿Considera usted que facilitaría el manejo de la información la creación de una aplicación web?</p>	<p>En la actualidad las empresas en todos los sectores de mercado poseen un alto número de competidores, es por ello que es importante en una compañía aplicar estrategias comerciales y herramientas de ventas adecuadas que se adapten al mundo de hoy. Al implementar un catálogo web efectivamente permitirá que se registre de mejor manera los productos y que estos puedan ser consultados de forma oportuna por los clientes de acuerdo a sus requerimientos.</p>
<p>4. ¿Qué documentos de salida ser requiere generar en la aplicación web?</p>	<p>No se requiere documentos de salida generados en la aplicación.</p>

Tabla 4.1: Entrevista al administrador de sistemas
Elaborado por: Rubén Jiménez – Investigador

Entrevista al ayudante de sistemas

Preguntas	Respuestas
1. ¿Considera usted que facilitaría el manejo de la información la creación de una aplicación web?	Hoy en día estamos en la era digital y las empresas para posicionarse en un segmento alto del mercado deben adaptar sus catálogos de productos para ser usados en medios electrónicos. Un catálogo web además de permitir una mejor administración de la información, permite la presentación de productos de una manera intuitiva y explicativa con incorporación de recursos como videos, imágenes incluso animaciones.
2. ¿Qué contenidos deben figurar en la aplicación web?	Información de productos con el precio fijado de acuerdo al cliente que ha ingresado a la aplicación.
3. ¿Considera que un manual de usuario le ayudaría a mejorar el servicio?	Un manual de usuario ayudaría no solo a los clientes sino también al personal de la empresa a entender el uso de la aplicación.

Tabla 4.2: Entrevista al ayudante de sistemas
Elaborado por: Rubén Jiménez – Investigador

Análisis de los resultados de las entrevistas

Una vez realizada las entrevistas se pudo conocer las necesidades de los usuarios frente al manejo y consulta de los catálogos de la empresa. Se puede concluir que actualmente se maneja de manera ineficiente el registro de los productos ya que posee dos sistemas:

Microplus que es un sistema contable en el que se registran el inventario general de los productos y los clientes; y un sistema externo llamado OM que de igual manera registra productos pero de manera más detallada. Ninguno de estos dos sistemas se encuentran enlazados y además almacenan los datos por separado siendo manejados por diferentes sistemas de gestión de base de datos.

Además los gestores de base de datos hacen ineficiente el almacenamiento de la información: Microsoft Access es un sistema de gestión de base de datos básico que posee limitaciones en la capacidad de almacenamiento, en el procesamiento de datos y

en la estabilidad; por otro lado Sybase Adaptive Server Anywhere es un software de pago por lo que la empresa debe pagar cada cierto tiempo la renovación de la licencia de uso, además que la versión utilizada se encuentra obsoleta ya que su liberación original fue en el año 2002.

La empresa posee una página web con la que los clientes pueden acceder a los productos sin embargo requiere que estos puedan acceder a la información de manera exclusiva y de acuerdo a sus requerimientos, ya que Turbomekanics S.A no solo posee clientes nacionales sino también internacionales. De igual manera la página actual es estática ya que solo presenta los productos sin la posibilidad de que el cliente pueda realizar la compra directamente desde la página web.

4.1.1.2 Observación de procesos

Mediante el uso de una guía de observación se puede identificar la manera en la que la empresa administra sus catálogos con los respectivos usuarios que intervienen en el proceso. A continuación se resume la información obtenida mediante la observación:

GUIA DE OBSERVACIÓN
<p>Objetivo: Recolectar información sobre el manejo actual de los catálogos de la empresa para la realización de una aplicación web.</p>
<p>Medio de recolección de la información</p> <ul style="list-style-type: none"> - Microplus: sistema contable que proporciona la información de los productos (inventario general) y clientes. - OM: sistema externo integrado que proporciona información más detallada del inventario ingresado en el sistema microplus.
<p>Medio de almacenamiento</p> <ul style="list-style-type: none"> - Base de datos Sybase Adaptive Server Anywhere versión 9 conectado a Microplus. - Microsoft Access conectado a OM.
<p>Documentos generados</p> <ul style="list-style-type: none"> - No se generan documentos.
<p>Disponibilidad de la información</p> <ul style="list-style-type: none"> - Información de clientes y precios sin enlazar a un sistema informático. - Clientes no pueden realizar la compra online del producto.

Tabla 4.3: Guía de observación

A continuación se detalla en un diagrama de proceso en que el administrador de sistemas establece los precios de los productos del catálogo de la empresa en los sistemas existentes:

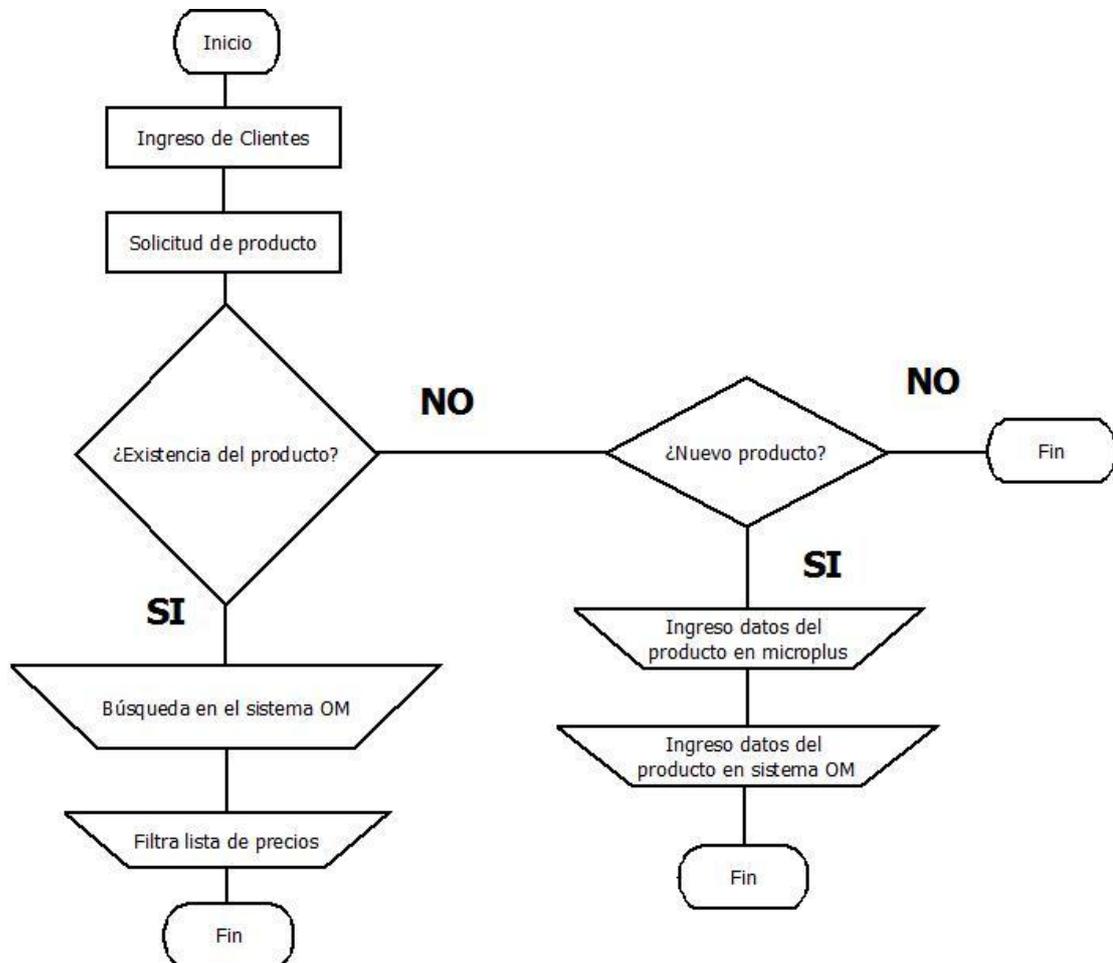


Figura 4.1: Diagrama de proceso- precio del producto
Elaborado por: Rubén Jiménez – Investigador

4.1.1.3 Análisis de procesos de marketing

Objetivos

La estrategia de marketing actual de la empresa busca cumplir con los objetivos:

- Tener un sitio web acorde a las necesidades de la empresa.
- Brindar facilidades para compra de los productos.
- Dar a conocer a los clientes las características de los productos.

- Fidelizar clientes.

Herramientas actuales

Al ingresar al catálogo web se puede constatar que se encuentra dividido por marcas.



Figura 4.2: Principal catálogo actual
Elaborado por: Rubén Jiménez – Investigador

Al seleccionar una marca nos presenta los turbos disponibles para esa marca:



Figura 4.3: Lista de turbos por marca
Elaborado por: Rubén Jiménez – Investigador

Al seleccionar los turbos únicamente se muestra imágenes de los productos.



Figura 4.4: Turbo seleccionado
Elaborado por: Rubén Jiménez – Investigador

Sin embargo, la empresa requiere que se muestre toda la información completa de los productos con los precios filtrados de acuerdo al cliente que ha iniciado sesión y que este pueda realizar la compra online.

Producto y precio

Los atributos tangibles e intangibles que se requiere incluir son precio, fabricante, color. El precio debe ser filtrado de acuerdo a una lista de clientes.

4.1.2 Definición de ámbitos del sistema

4.1.2.1 Requerimientos funcionales

Público

El cliente anónimo podrá visualizar la página principal de la aplicación con la opción de iniciar sesión en caso de poseer una cuenta de usuario.

Administrador

El usuario administrador podrá ingresar, eliminar y actualizar usuarios en la aplicación. Además podrá visualizar la información de grupos (los cuales son fijos en la aplicación), clientes, artículos y sus categorías.

Cliente

El usuario cliente podría visualizar la información de los artículos filtrados por su precio y por su categoría. Asimismo podrá gestionar las compras realizadas mediante un carro de compras virtual y recibir notificaciones al correo electrónico.

4.1.2.2 Requerimientos no funcionales

Se requiere que la aplicación sea capaz de realizar las siguientes funciones:

- La aplicación debe tener un diseño que sea de fácil uso para los usuarios.
- Deber ser eficiente para procesar gran cantidad de consultas y transacciones.
- Deber tener un diseño responsivo para poder ser visualizado adecuadamente ya sea en un computador, tablet o teléfono inteligente.
- La información debe presentarse a los usuarios de acuerdo a los privilegios con los que hayan accedido.
- Facilitará la búsqueda de información al usuario.
- Los datos de los clientes deben ser mantenidos de manera confiable por la aplicación.
- Cualquier intercambio de datos vía internet que realice el software se realizará por medio del protocolo encriptado https.

4.1.3 Establecer requerimientos de hardware y software

Se conoce que la empresa requiere reducir su gasto en licenciamiento de software razón por la cual el investigador propone trabajar con herramientas open source para el desarrollo.

Para determinar el software adecuado se ha realizado un estudio comparativo de las herramientas a utilizar (lenguaje de programación y sistema de gestión de base de datos) de aspectos generales como la experiencia a de uso, nivel de conocimiento, actualizaciones y el manual actualizado.

4.1.3.1 Lenguaje de programación

Se han comparado tres de los lenguajes de programación web con mayor demanda en la actualidad:

	PHP	PYTHON	RUBY ON RAILS
Última actualización	10 de noviembre de 2016 (versión 5.6.28)	26 de junio de 2016 (versión 3.5.2)	11 de agosto de 2016 (versión 5.0.0.1)
Comunidad activa	https://secure.php.net/	https://www.python.org/	www.rubyonrails.org
Manual actualizado	Manual actualizado hasta la versión 4.x	Manual actualizado hasta la versión 3.x	Manual actualizado hasta la versión 4.2
Nivel de conocimiento y experiencia de uso	Alto	Medio	Medio
Servicios web	SOAP	SOAP	REST
Tipo de aplicaciones	Proyectos web	Proyectos web	Aplicaciones de escritorio
Curva de aprendizaje	Baja	Baja	Alta
Demanda de trabajo [20]	8,238	2,937	1,587
Bases de datos soportadas	CUBRID, DB++, DBase, filePro, Firebird/Interbase/ FrontBase, IBM DB2, Informix, Ingres, MaxDB, MongoDB, mSQL, Microsoft SQL Server, MySQL, Oracle OCI8, Paradox, PostgreSQL, SQLite, Sybase [21]	IBM DB2, Firebird/ Interbase, Informix, Ingres, MySQL, Oracle, PostgreSQL, SAP DB (MaxDB)/ Microsoft SQL Server, Microsoft Access, SQLite, Sybase [22]	MySQL, SQLite, PostgreSQL, SQLite3, IBM Data Servers.

Tabla 4.4: Tabla comparativa de lenguajes de programación web

Análisis

Tomando en cuenta los aspectos analizados anteriormente se escogió el uso de PHP por las siguientes razones:

- Posee soporte para la base de datos usada actualmente en la empresa (Sybase).
- Su curva de aprendizaje es relativamente baja con una extensa comunidad y documentación en su web oficial.
- Existe una gran demanda de trabajo para los desarrolladores PHP.
- Se encuentra más actualizado
- Se tiene más conocimiento y experiencia de uso.

“PHP es un lenguaje de script que se ejecuta en el lado del servidor, cuyo código se incluye en una página HTML clásica.” [23]

El código de PHP se ejecuta en el servidor y el resultado se integra al código HTML para ser enviada al explorador.



Figura 4.5: Funcionamiento de PHP

Fuente: [24]

Las ventajas y características principales que destacan en PHP son las siguientes:

- Lenguaje multiplataforma, funciona en diversos sistemas operativos como Linux, Unix o Windows.
- Orientado a objeto, soporta herencia y creación de clases con sus constructores.
- Amplia biblioteca de funciones.
- Expandible gracias a su gran cantidad de módulos (extensiones).
- Manejo de excepciones.

4.1.3.2 Base de datos

La empresa posee un contrato de hosting web con compatibilidad con motores de base de datos MySQL, PostgreSQL y SQL Server. Como se propone la utilización de un gestor de base de datos open source se realiza un análisis entre MySQL y PostgreSQL:

	MySQL	PostgreSQL
Velocidad de respuesta a búsqueda e inserciones [25]	Alta	Media
Consumo de recursos	Pocos	Bastantes
Lenguajes de programación soportados	C, C#, Java, Haskell, Perl, PHP, Python, Ruby	Php, Java, Python, Perl, .Net, C, C++
Escalabilidad	No soporta	Si
Script del lado de servidor	Si	Sólo funciones definidas por usuario
Seguridad	Media	Alta
Triggers	Si	Si
Soporte para transacciones	No soporta	Si
Migración de datos desde Sybase active server anywhere	Si	Si

version 9		
------------------	--	--

Tabla 4.5: Tabla comparativa de base de datos
Elaborado por: Rubén Jiménez – Investigador

Análisis

Dado a que la empresa posee información almacenada en Sybase active server anywhere versión 9, es necesario la que dichos datos sean migrados para ser usados en la aplicación y para ello es necesario un mayor consumo de recursos con una mayor rapidez. Por consiguiente, una mejor opción sería optar por MySQL en lugar de PostgreSQL ya que el primero trabaja de manera más optimizada con PHP en la web. MySQL junto con PHP forman un buen equipo para la creación de aplicaciones dinámicas que requieran velocidad y que la seguridad no sea un elemento esencial ya que en este punto PostgreSQL destaca como mejor opción.

Se opta por la utilización de la base de datos MySQL para el desarrollo debido que entre otras ventajas posee un amplio uso entre los desarrolladores por su facilidad de uso e instalación, rapidez y eficiencia.

“MySQL es un sistema de administración de base de datos relacionales (SGBDR) rápido, robusto y fácil de usar. Se adapta bien a la administración de datos en un entorno de red, especialmente en arquitecturas cliente/servidor. Se proporciona con muchas herramientas y es compatible con muchos lenguajes de programación. Es el más célebre SGBDR del mundo Open Source, en particular gracias a su compatibilidad con el servidor de páginas Web Apache y el lenguaje de páginas Web dinámicas PHP” [26]

4.1.4 Análisis de la arquitectura MVC

IEEE Std 1471-2000 define a Arquitectura de Software como “La organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución” [27]

La arquitectura representa la estructura de la aplicación con las partes que la componen y sus respectivas interacciones y restricciones.

Debido al gran auge del desarrollo web, se ha tomado en cuenta una arquitectura que sea extensible y permita adaptaciones a cambios futuros; debido a esto se plantea el uso de uno de los más utilizados en la actualidad como es el patrón MVC.

“El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocios de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones” [15]

4.1.4.1 Elementos

Esta arquitectura separa tres componentes que corresponden a los datos de la aplicación, la interfaz gráfica de usuario y la lógica de negocio.

Modelo:

Representa la información almacenada en la base de datos con la cual opera la aplicación. Gestiona las validaciones y las operaciones básicas sobre los datos (inserción, consulta, actualización y eliminación). Las funciones del modelo son:

- Acceder a la capa de almacenamiento de datos.
- Definir la funcionalidad de la aplicación (reglas de negocio).
- Notificación de cambios en los datos a las vistas.

Vista:

Representa la información del modelo enviado a través del controlador para ser visualizado por el usuario en un formato legible. En otras palabras, constituye la interfaz para que el usuario pueda interactuar con la aplicación. Generalmente es una página web aunque pueden también ser un fragmento de página como un encabezado o un pie de página. Las funciones de las vistas son:

- Recibir datos desde el controlador o el modelo.
- Mostrar información al usuario.

Controlador:

Representa un intermediario entre el modelo y la vista. Recibe y responde los eventos realizados desde la vista, y a su vez invoca a métodos correspondientes en el modelo para generar una respuesta necesaria con los datos adecuados. Las funciones de los controladores son:

- Recibir peticiones de entrada.
- Gestionar eventos que pueden ser peticiones a las vistas o al modelo.

4.1.4.2 Funcionamiento

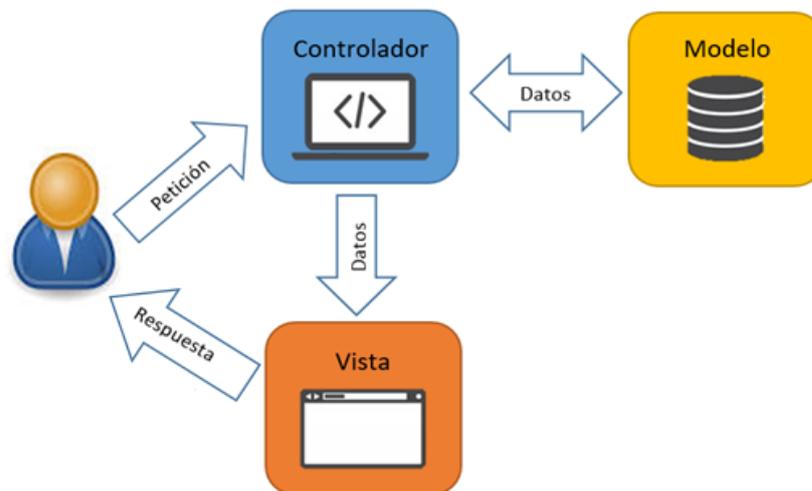


Figura 4.6: Arquitectura MVC
Elaborado por: Rubén Jiménez – Investigador

El principio de funcionamiento de la arquitectura Modelo – Vista – Controlador es el siguiente:

- El cliente introduce una petición HTTP (una acción o evento) mediante la interacción con la interfaz de usuario.
- El controlador recibe el evento, lo traduce y gestiona generalmente a través de un gestor de eventos (handler) o callback.

- El controlador envía la petición accediendo al modelo o a la vista directamente.
- El controlador envía los datos del modelo que ejecuta los métodos para su manipulación (ingreso, visualización, actualización, eliminación) de acuerdo a la acción solicitada por el usuario.
- El controlador recibe la información desde el modelo y los trasmite a la vista.

La vista presenta una respuesta al usuario en un formato legible presentando los cambios en el modelo.

4.1.4.3 Ventajas y Desventajas

Ventajas	Desventajas
<ul style="list-style-type: none"> •Separa los componentes (interfaz de usuario - lógica de negocio - acceso a datos) y permite su implementación por separado, es por ello que si un componente falla puede modificarse sin que el resto del programa se vea afectado •La conexión entre el modelo y sus vistas se produce en tiempo de ejecución, no de compilación. •Se puede construir varias vistas para un solo modelo sin modificarlo. •Posee mecanismos de configuración de componentes complejos de manera más sencilla que en programación basada en eventos, permite su reutilización y mantenimiento de manera sencilla. •Facilidad de realizar pruebas unitarias. •Las vistas se actualizan automáticamente. •Las aplicaciones realizadas siguiendo el patrón MVC presentan gran mantenibilidad, modificabilidad y extensibilidad. •Muchos frameworks incluyen librerías Javascript facilitando validar formularios. •Controla de mejor manera los recursos del servidor, evita errores que repercuten en el rendimiento. 	<ul style="list-style-type: none"> •Es necesario el desarrollo de un mayor número de clases que usando otro tipo de arquitectura. •Requiere de una arquitectura inicial que incluya mecanismos de eventos. •Su implementación es costosa y difícil en lenguajes que no sigan el paradigma orientado a objetos. •La separación en capas hace que la aplicación sea más compleja. •Gran cantidad de archivos a mantener y desarrollar. •Curva de aprendizaje alta.

Tabla 4.6: Tabla ventajas y desventajas de MVC
Elaborado por: Rubén Jiménez – Investigador

4.1.5 Definición de casos de uso

“Un caso de uso es una manera de utilizar el sistema o de interactuar con él. Los casos de uso proporcionan una definición de las necesidades a cubrir por un proyecto desde el punto de vista del usuario” [28]

Los casos de uso representan la usabilidad y comportamiento que tendrán los usuarios en el sistema. Además, permiten identificar que se conozcan todos los requisitos del cliente. Se han determinado los siguientes usuarios en la aplicación:

- Público, Administrador y Cliente.

4.1.5.1 Diagrama caso de uso Público

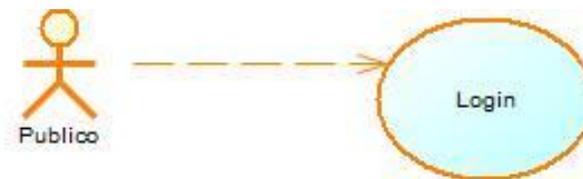


Figura 4.7: Casos de uso Público
Elaborado por: Rubén Jiménez – Investigador

Especificación del caso de uso

Caso de uso: Login en el sistema	
Actor:	Público
Resumen:	Permite el acceso del usuario a la aplicación
Precondición:	Usuario registrado en el sistema.
Descripción:	Trayectoria Básica: <ul style="list-style-type: none"> • Ingresar usuario y contraseña válidos. • Pulsar botón “Ingresar” • Redireccionamiento a la pantalla principal Trayectoria Alterna: <ul style="list-style-type: none"> • Mensajes de error
Postcondición:	Mostrar la página principal con opciones de menú acorde al rol.

Tabla 4.7: Especificación caso de uso: login en el sistema
Elaborado por: Rubén Jiménez – Investigador

4.1.5.2 Diagrama caso de uso Administrador

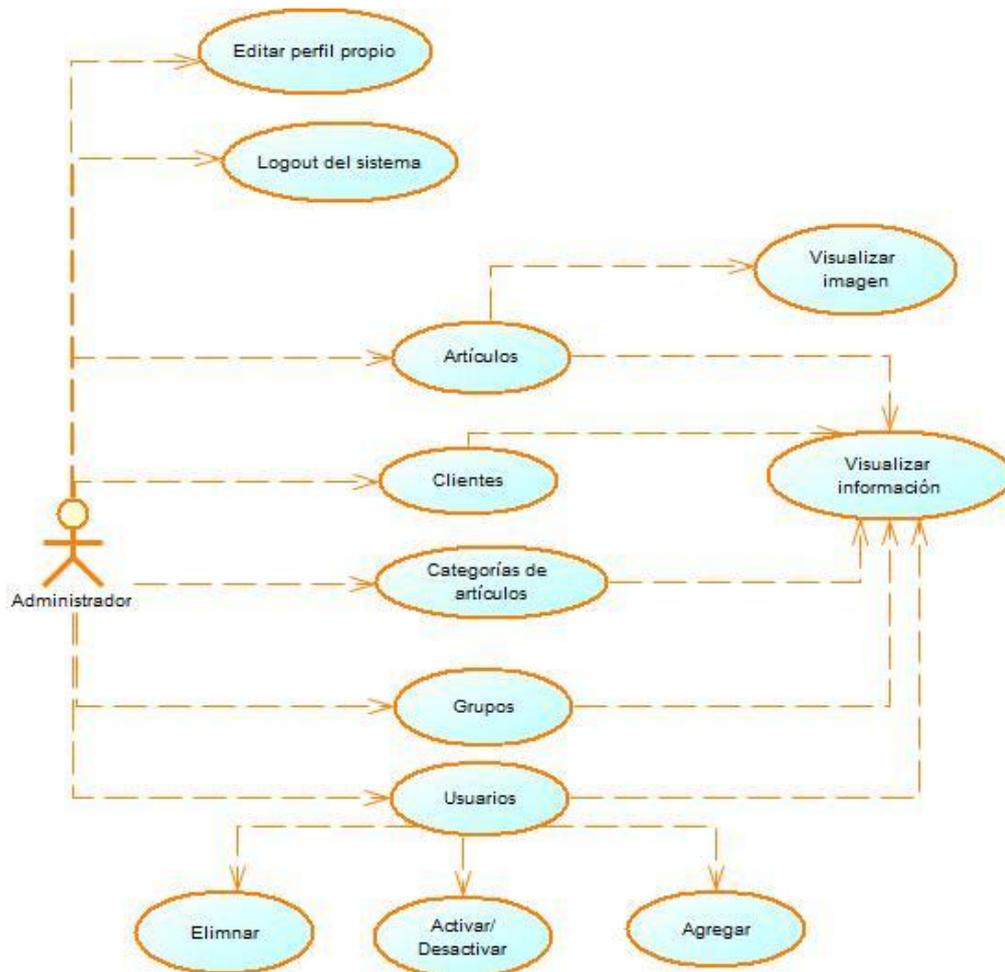


Figura 4.8: Casos de uso Administrador
Elaborado por: Rubén Jiménez – Investigador

Especificaciones de caso de uso

Caso de uso: Logout en el sistema	
Actor:	Administrador, Cliente
Resumen:	Cierra la sesión del usuario.
Precondición:	Usuario debe haber iniciado sesión.
Descripción:	Trayectoria Básica: <ul style="list-style-type: none"> • Pulsar botón “Cerrar sesión” • Redireccionamiento a la pantalla principal
Postcondición:	Ninguna.

Tabla 4.8: Especificación caso de uso: logout en el sistema
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Editar perfil propio

Actor:	Administrador, Cliente
Resumen:	Permite al usuario cambiar su información de cuenta en el sistema.
Precondición:	Haber iniciado sesión en la aplicación.
Descripción:	<p>Trayectoria Básica:</p> <ul style="list-style-type: none"> • Pulsar botón Editar perfil. • Mostrar modal de edición. • Modificar la información deseada. • Pulsar botón “Guardar” • Verificar información ingresada. • Guardar y cerrar ventana modal de edición. <p>Trayectoria Alterna:</p> <ul style="list-style-type: none"> • Pulsar botón “Cancelar”. • Cerrar ventana modal de edición.
Postcondición:	Ninguna.

Tabla 4.9: Especificación caso de uso: Editar perfil
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Visualizar información de artículo	
Actor:	Administrador, Cliente
Resumen:	Permite ver la información completa de un artículo de la aplicación.
Precondición:	Artículo ingresado en la base de datos.
Descripción:	<p>Trayectoria Básica:</p> <ul style="list-style-type: none"> • Seleccionar categoría de artículo • Seleccionar artículo • Mostrar información del artículo seleccionado
Postcondición:	Ninguna.

Tabla 4.10: Especificación caso de uso: Visualizar información de artículo
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Visualizar imagen de artículo

Actor:	Administrador, Cliente
Resumen:	Permite visualizar la imagen de un artículo de la aplicación.
Precondición:	Artículo ingresado en la base de datos.
Descripción:	Trayectoria Básica: <ul style="list-style-type: none"> • Seleccionar imagen de artículo. • Mostrar modal con imagen del artículo seleccionado
Postcondición:	Ninguna.

Tabla 4.11: Especificación caso de uso: Visualizar imagen de artículo
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Visualizar información de categoría de artículo	
Actor:	Administrador
Resumen:	Permite ver la información de las categorías de artículos de la aplicación.
Precondición:	Ingresar como rol de administrador Categorías de artículos ingresados en la base de datos.
Descripción:	Trayectoria Básica: <ul style="list-style-type: none"> • Seleccionar categoría de artículo. • Mostrar información de la categoría seleccionada.
Postcondición:	Ninguna.

Tabla 4.12: Especificación caso de uso: Visualizar categoría de artículo
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Visualizar información de cliente

Actor:	Administrador
Resumen:	Permite ver la información completa de un cliente de la aplicación.
Precondición:	Ingresar como rol de administrador Cliente ingresado en la base de datos.
Descripción:	Trayectoria Básica: <ul style="list-style-type: none"> • Seleccionar cliente. • Mostrar información del cliente seleccionado
Postcondición:	Ninguna.

Tabla 4.13: Especificación caso de uso: Visualizar información de cliente
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Visualizar información de grupos	
Actor:	Administrador
Resumen:	Permite ver la información de los grupos de la aplicación.
Precondición:	Ingresar como rol de administrador Grupos ingresados en la base de datos.
Descripción:	Trayectoria Básica: <ul style="list-style-type: none"> • Seleccionar grupo. • Mostrar información del grupo seleccionada.
Postcondición:	Ninguna.

Tabla 4.14: Especificación caso de uso: Visualizar información de grupo
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Agregar información de usuarios
--

Actor:	Administrador
Resumen:	Permite agregar un nuevo usuario a la aplicación.
Precondición:	Ingresar como rol de administrador
Descripción:	<p>Trayectoria Básica:</p> <ul style="list-style-type: none"> • Seleccionar usuario. • Pulsar el botón “Agregar” • Mostrar modal de ingreso de nuevo usuario. • Ingresar la información pedida en el formulario. • Pulsar botón “Guardar” • Validar la información ingresada. • Confirmar, guardar la información y cerrar ventana modal. <p>Trayectoria Alterna:</p> <ul style="list-style-type: none"> • Pulsar botón “Cancelar”. • Cerrar ventana modal.
Postcondición:	Ninguna.

Tabla 4.15: Especificación caso de uso: Agregar información de usuario
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Visualizar información de usuarios	
Actor:	Administrador
Resumen:	Permite ver la información de los usuarios de la aplicación.
Precondición:	Ingresar como rol de administrador Usuarios ingresados en la base de datos.
Descripción:	<p>Trayectoria Básica:</p> <ul style="list-style-type: none"> • Seleccionar usuario. • Mostrar información del usuario seleccionada.
Postcondición:	Ninguna.

Tabla 4.16: Especificación caso de uso: Visualizar información de usuario
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Eliminar información de usuarios
--

Actor:	Administrador
Resumen:	Permite eliminar un usuario de la aplicación.
Precondición:	Ingresar como rol de administrador Usuario ingresado en la base de datos.
Descripción:	<p>Trayectoria Básica:</p> <ul style="list-style-type: none"> • Seleccionar usuario. • Pulsar el botón “Eliminar” • Mostrar modal de confirmación. • Eliminar el registro. <p>Trayectoria Alterna:</p> <ul style="list-style-type: none"> • Pulsar botón “Cancelar”. • Cerrar ventana modal.

Tabla 4.17: Especificación caso de uso: Eliminar información de usuario
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Editar información de usuarios	
Actor:	Administrador
Resumen:	Permite editar un usuario de la aplicación.
Precondición:	Ingresar como rol de administrador Usuario ingresado en la base de datos.
Descripción:	<p>Trayectoria Básica:</p> <ul style="list-style-type: none"> • Seleccionar usuario. • Pulsar el botón “Activar o desactivar”
Postcondición:	Si el usuario se encontrará inactivo, no podrá usar la aplicación.

Tabla 4.18: Especificación caso de uso: Editar información de usuario
Elaborado por: Rubén Jiménez – Investigador

4.1.5.3 Diagrama caso de uso Cliente

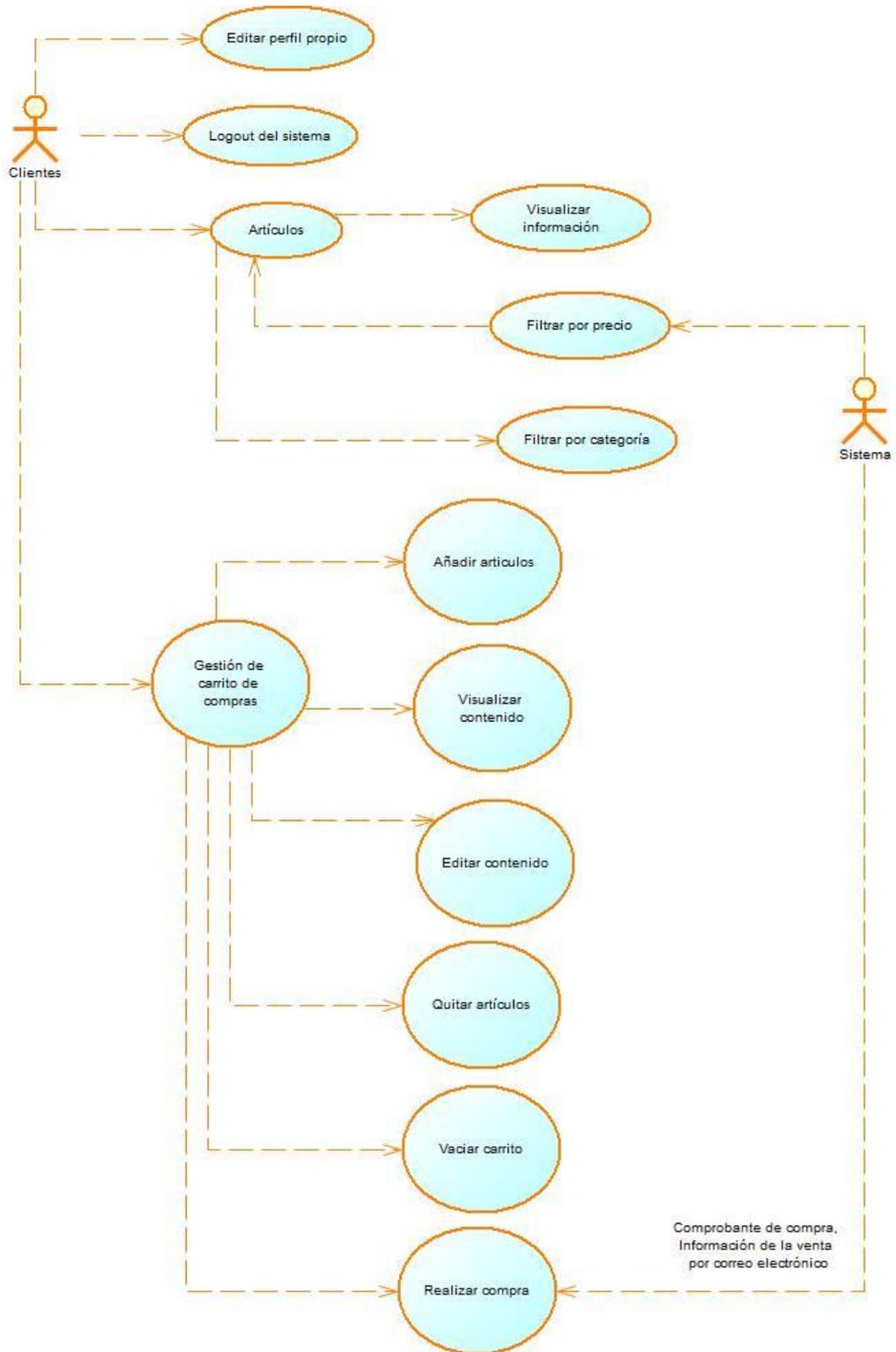


Figura 4.9: Casos de uso Cliente
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Filtrar artículo por precio	
Actor:	Sistema

Resumen:	El sistema filtra los artículos por su precio para que sean consultados por el cliente.
Precondición:	Ninguna.
Descripción:	Trayectoria Básica: <ul style="list-style-type: none"> • Ingreso del usuario cliente. • Asocia al cliente a una lista de precios. • Filtrar los artículos de acuerdo al cliente que ha ingresado.
Postcondición:	Datos filtrados por precio.

Tabla 4.19: Especificación caso de uso: Filtrar artículo por precio
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Filtrar artículo por categoría	
Actor:	Cliente
Resumen:	Permite al usuario filtrar los artículos por su categoría.
Precondición:	Ingresar como rol de cliente.
Descripción:	Trayectoria Básica: <ul style="list-style-type: none"> • Página principal de artículos. • Seleccionar categoría en opción de filtrado. • Ingresar parámetro de búsqueda. • Presentar datos filtrados por categoría ingresada
Postcondición:	Datos filtrados por categoría ingresada.

Tabla 4.20: Especificación caso de uso: Filtrar artículo por categoría
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Añadir artículo carrito de compras
--

Actor:	Cliente
Resumen:	Permite al usuario añadir un artículo seleccionado al carrito de compras.
Precondición:	Ingresar como rol de cliente.
Descripción:	<p>Trayectoria Básica:</p> <ul style="list-style-type: none"> • Página principal de artículos. • Seleccionar opción añadir al carrito de compras. • El artículo es añadido al carrito de compras.
Postcondición:	El sistema realiza un decremento en el stock del artículo seleccionado.

Tabla 4.21: Especificación caso de uso: Añadir artículo al carrito de compras
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Visualizar contenido de carrito de compras	
Actor:	Cliente
Resumen:	Permite al usuario visualizar todo el contenido de su carrito de compras.
Precondición:	Ingresar como rol de cliente.
Descripción:	<p>Trayectoria Básica:</p> <ul style="list-style-type: none"> • Seleccionar carrito de compras. • Presentar la lista de artículos almacenados en el carrito de compras. <p>Trayectoria Alterna:</p> <ul style="list-style-type: none"> • Si no existen artículos se notificará que el carrito de compras se encuentra vacío.
Postcondición:	Filtrados de datos ingresados al carrito de compras.

Tabla 4.22: Especificación caso de uso: Visualizar contenido carrito de compras
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Editar contenido de carrito de compras	
Actor:	Cliente
Resumen:	Permite al usuario editar el carrito de compras.
Precondición:	Ingresar como rol de cliente. Deben existir artículos ingresados en el carrito de compras.
Descripción:	Trayectoria Básica: <ul style="list-style-type: none"> • Seleccionar carrito de compras. • Presentar la lista de artículos almacenados en el carrito de compras. • Aumentar o reducir la cantidad de artículos.
Postcondición:	El sistema realiza un aumento o decremento en el stock del artículo según sea el caso.

Tabla 4.23: Especificación caso de uso: Editar contenido de carrito de compras
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Quitar artículo del carrito de compras	
Actor:	Cliente
Resumen:	Permite al usuario quitar un artículo del carrito de compras.
Precondición:	Ingresar como rol de cliente. Deben existir artículos ingresados en el carrito de compras.
Descripción:	Trayectoria Básica: <ul style="list-style-type: none"> • Seleccionar carrito de compras. • Seleccionar eliminar en un artículo.
Postcondición:	El sistema eliminará el o los productos seleccionados.

Tabla 4.24: Especificación caso de uso: Quitar artículo del carrito de compras
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Vaciar carrito de compras	
Actor:	Cliente
Resumen:	Permite al usuario vaciar el carrito de compras.
Precondición:	Ingresar como rol de cliente. Deben existir artículos ingresados en el carrito de compras.
Descripción:	Trayectoria Básica: <ul style="list-style-type: none"> • Seleccionar vaciar el carrito de compras.
Postcondición:	El sistema eliminará todos los productos del carrito de compras.

Tabla 4.25: Especificación caso de uso: Vaciar carrito de compras
Elaborado por: Rubén Jiménez – Investigador

Caso de uso: Realizar compra	
Actor:	Cliente
Resumen:	Permite al usuario aceptar los ítems del carrito, el precio y confirmar la compra.
Precondición:	Ingresar como rol de cliente. Deben existir artículos ingresados en el carrito de compras.
Descripción:	Trayectoria Básica: <ul style="list-style-type: none"> • El usuario solicita realizar la compra de los artículos seleccionados. • El sistema muestra el pedido actual con los datos del usuario. • El usuario acepta el pedido. • El sistema genera un comprobante de compra y lo envía al correo electrónico del cliente. Trayectoria Alterna: <ul style="list-style-type: none"> • Cancelación: el usuario puede cancelar la compra en cualquier momento. • El usuario puede modificar el pedido.
Postcondición:	Se genera comprobante.

Tabla 4.26: Especificación caso de uso: Realizar compra
Elaborado por: Rubén Jiménez – Investigador

4.2 Elaboración

Se diseña una solución preliminar.

4.2.1 Diagramas de secuencia

“Los diagramas de secuencia muestran el flujo de mensajes (es decir la forma en que se invocan) entre objetos para un determinado caso de uso. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos.” [29]

4.2.1.1 Diagramas de secuencia Público

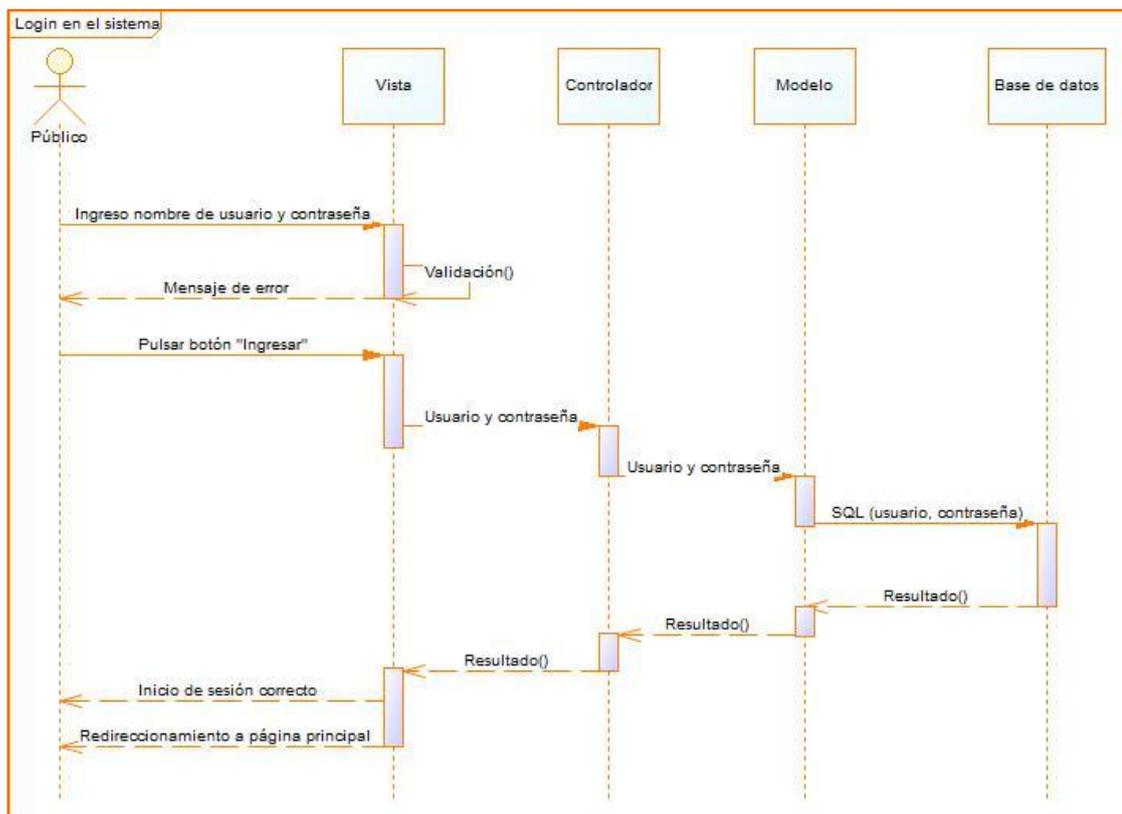


Figura 4.10: Diagrama de secuencia login en el sistema
Elaborado por: Rubén Jiménez – Investigador

4.2.1.2 Diagrama caso de uso Administrador

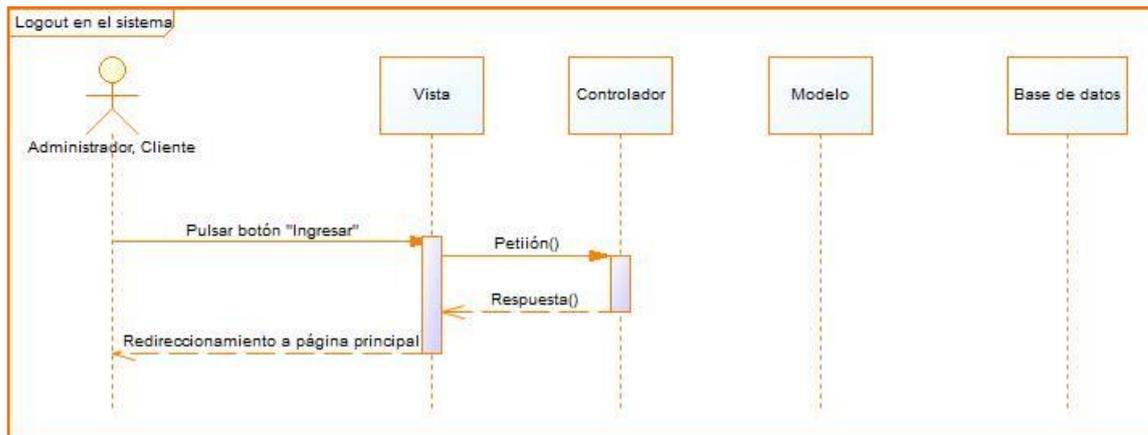


Figura 4.11: Diagrama de secuencia logout en el sistema
Elaborado por: Rubén Jiménez – Investigador

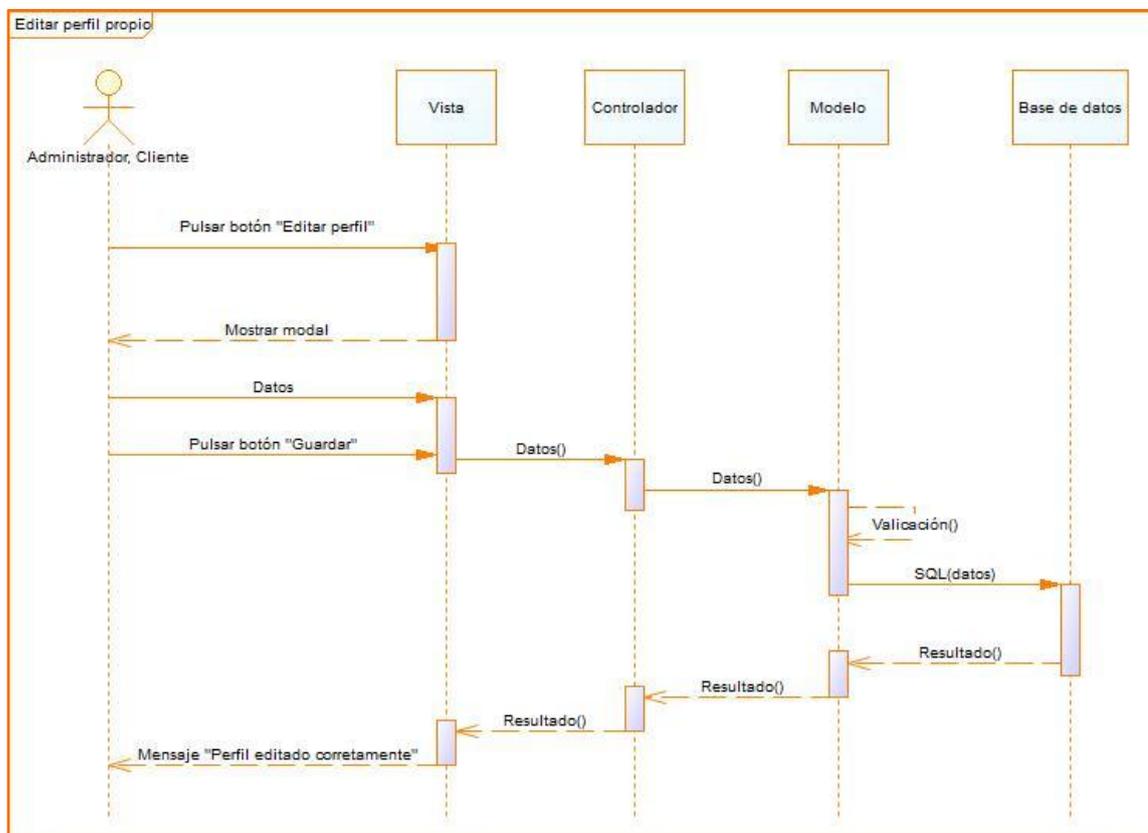


Figura 4.12: Diagrama de secuencia editar perfil propio
Elaborado por: Rubén Jiménez – Investigador

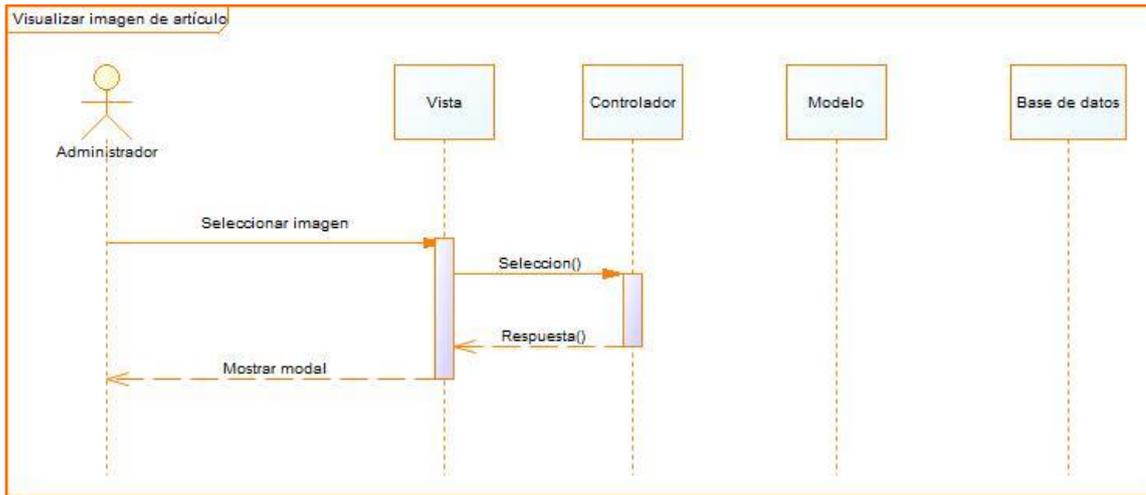


Figura 4.13: Diagrama de secuencia visualizar imagen de artículo
Elaborado por: Rubén Jiménez – Investigador

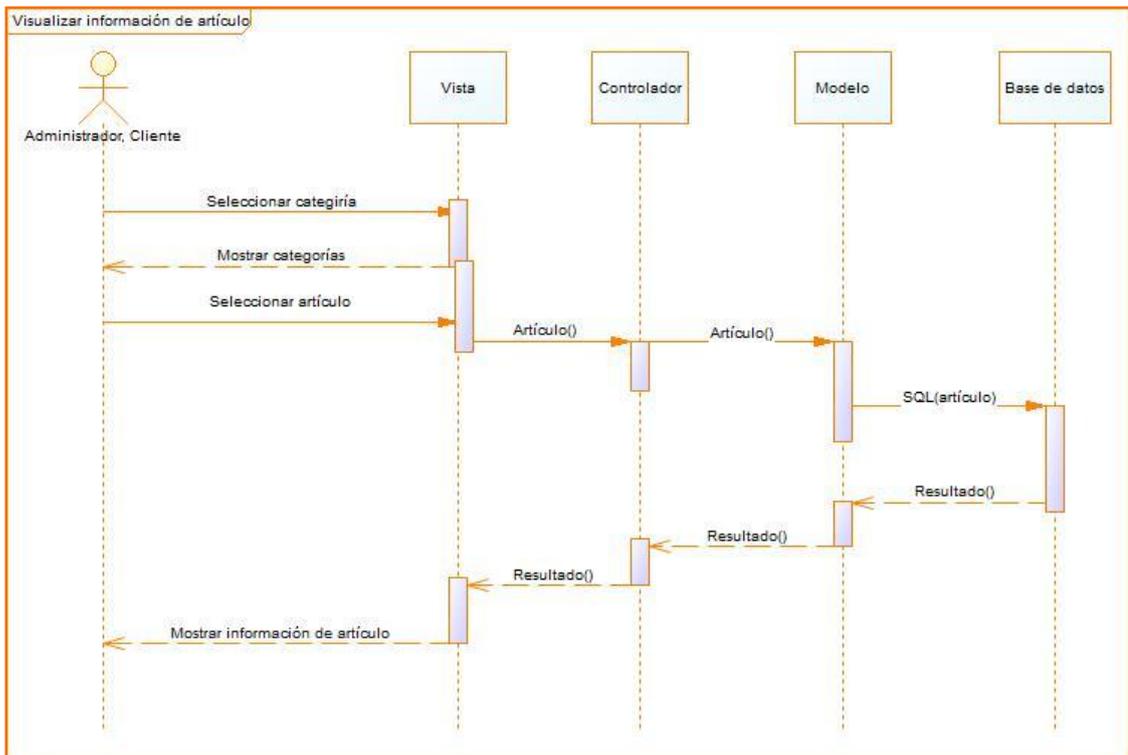


Figura 4.14: Diagrama de secuencia visualizar información de artículo
Elaborado por: Rubén Jiménez – Investigador

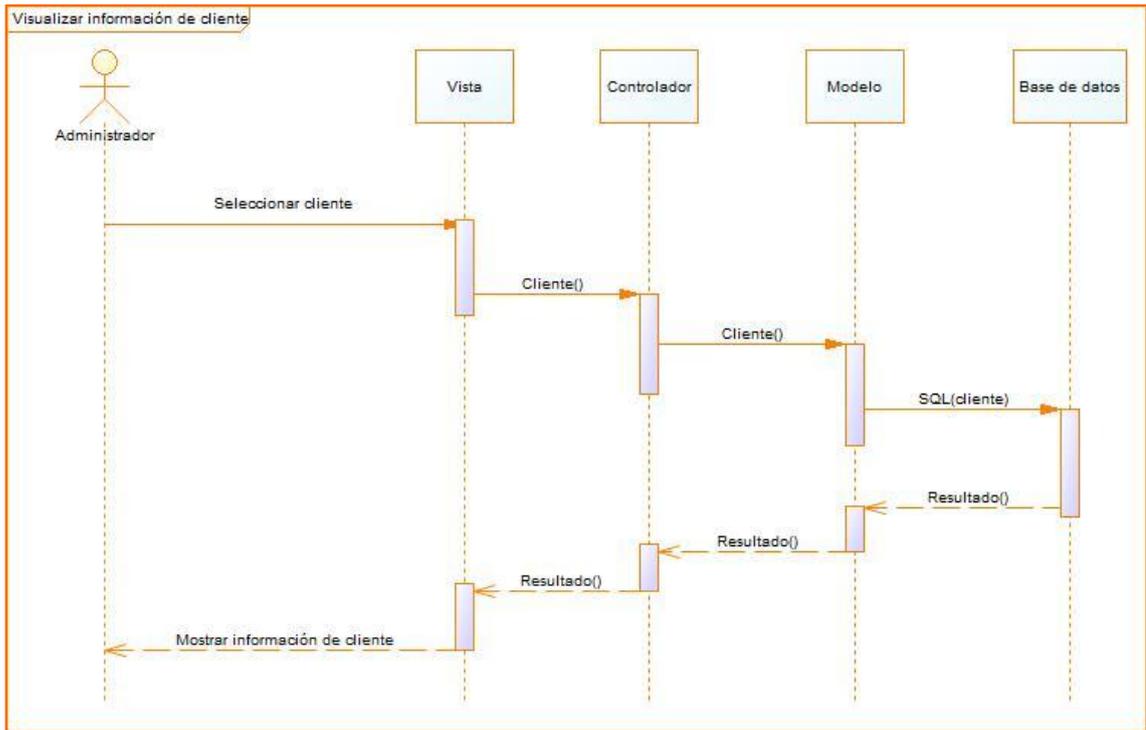


Figura 4.15: Diagrama de secuencia visualizar información de cliente
Elaborado por: Rubén Jiménez – Investigador

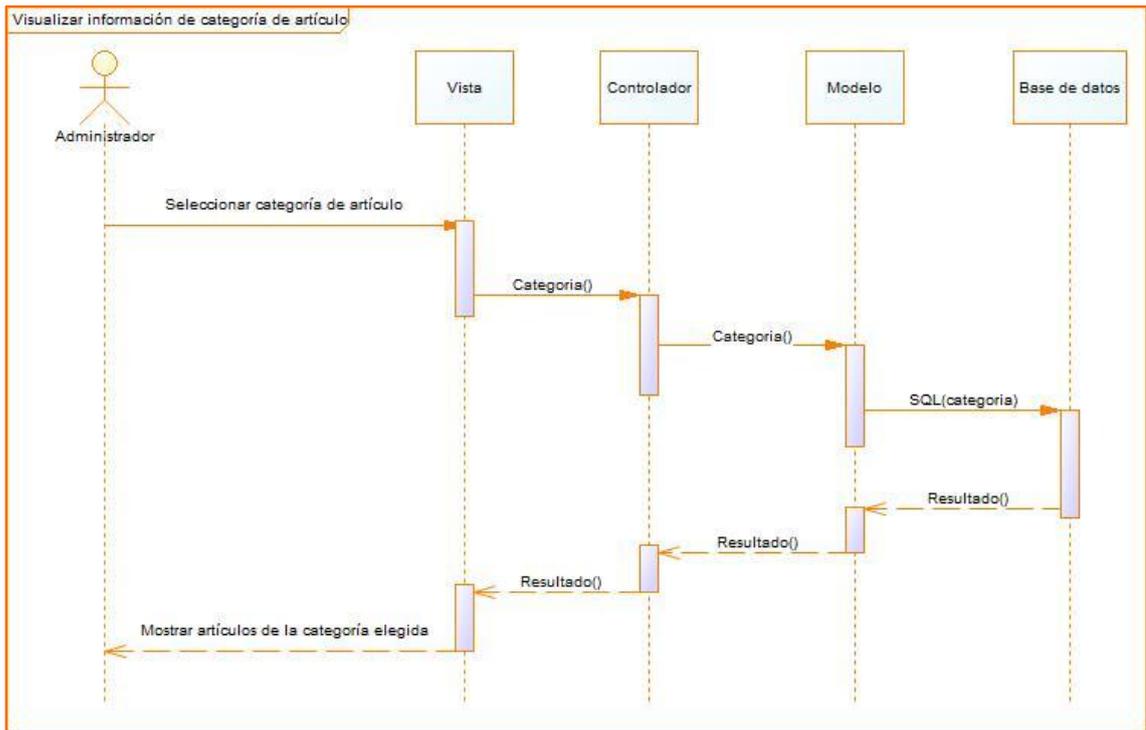


Figura 4.16: Diagrama de secuencia visualizar categoría de artículo
Elaborado por: Rubén Jiménez – Investigador

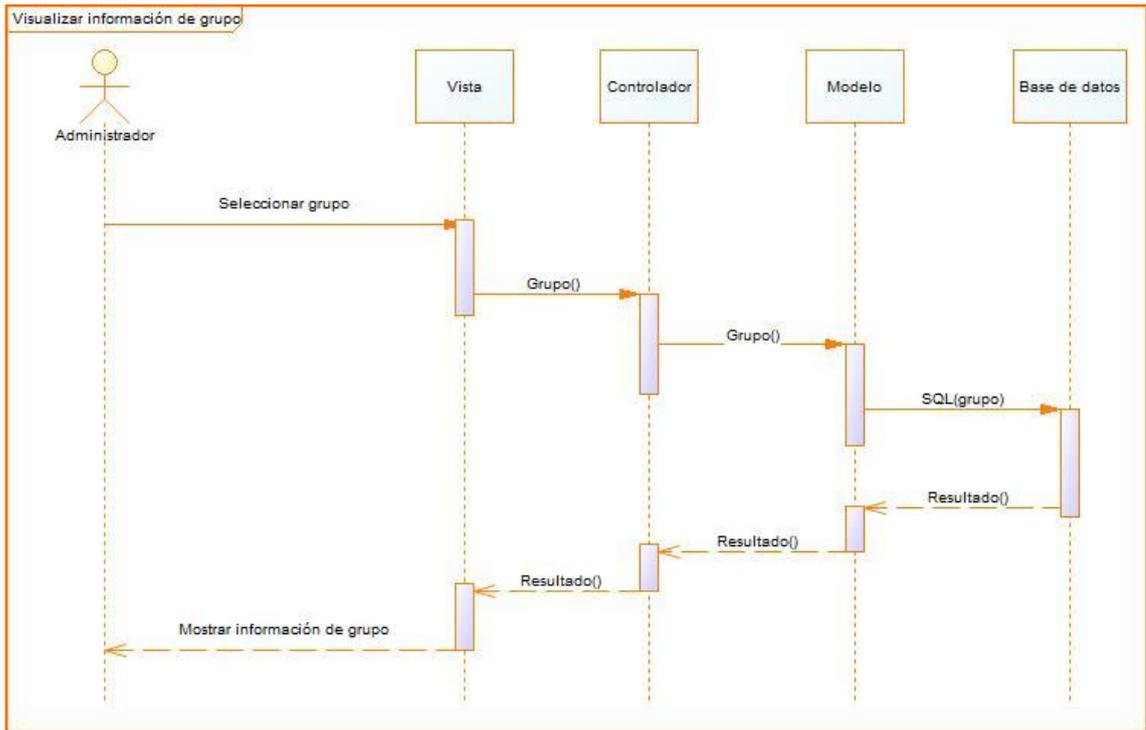


Figura 4.17: Diagrama de secuencia visualizar información de grupo
Elaborado por: Rubén Jiménez – Investigador

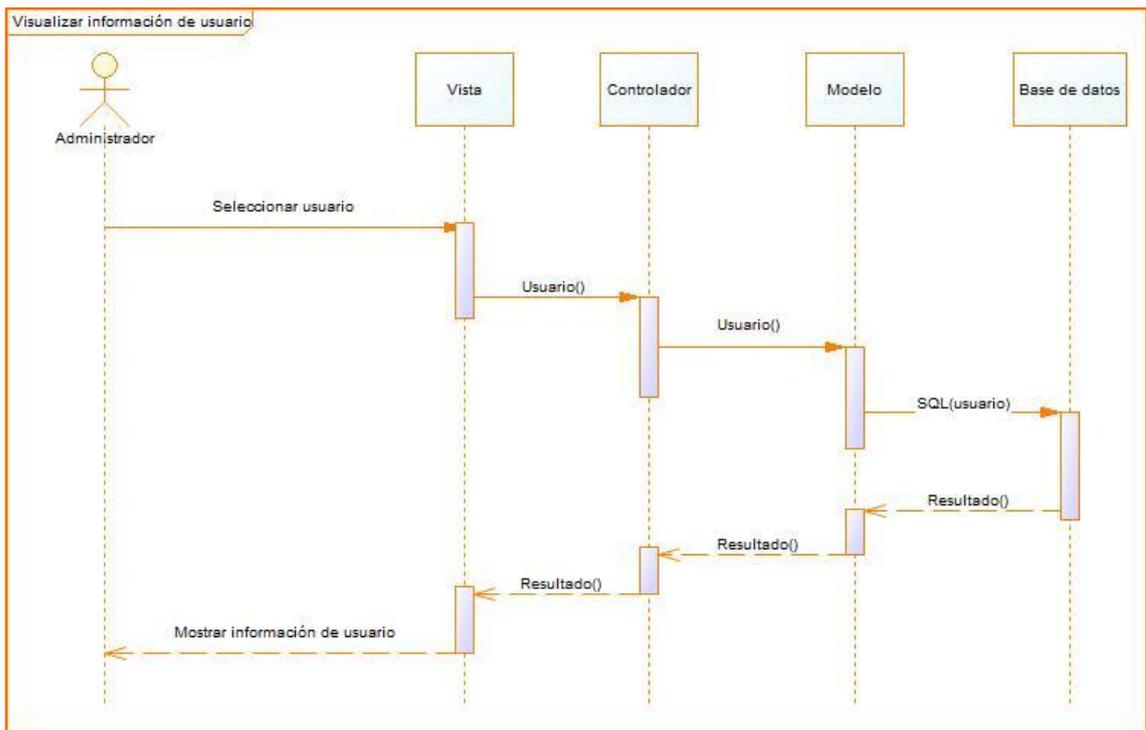


Figura 4.18: Diagrama de secuencia visualizar información de usuario
Elaborado por: Rubén Jiménez – Investigador

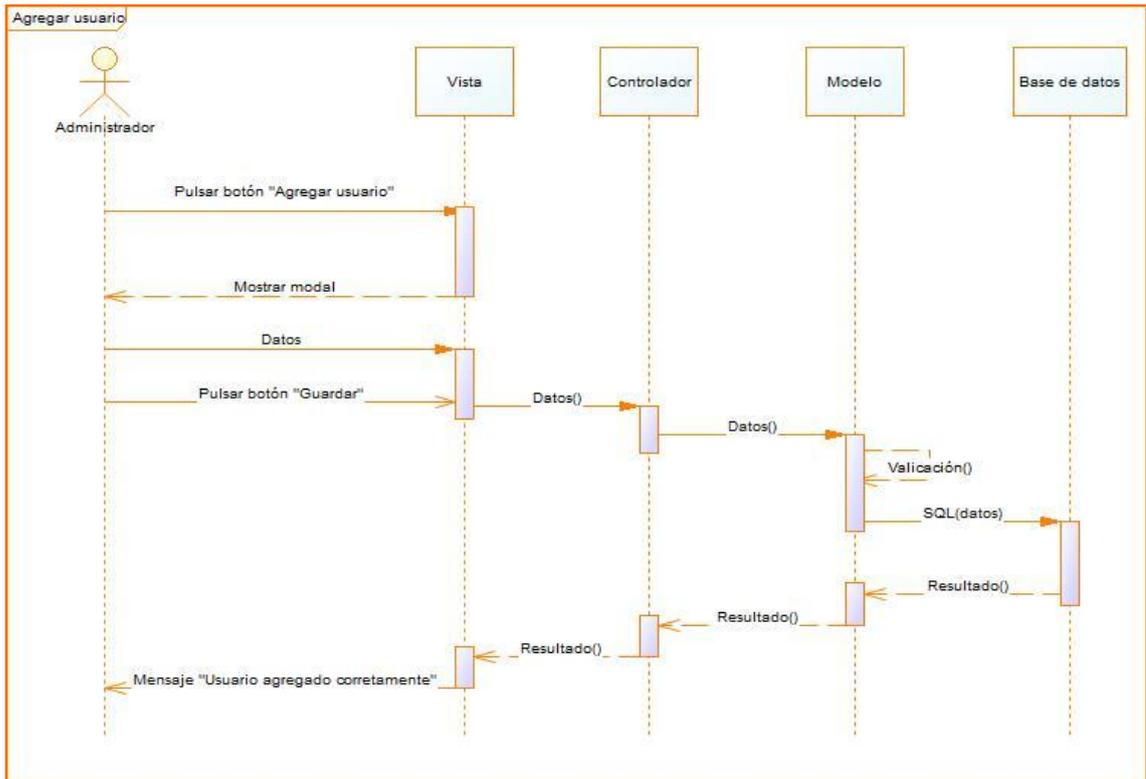


Figura 4.19: Diagrama de secuencia agregar usuario
Elaborado por: Rubén Jiménez – Investigador

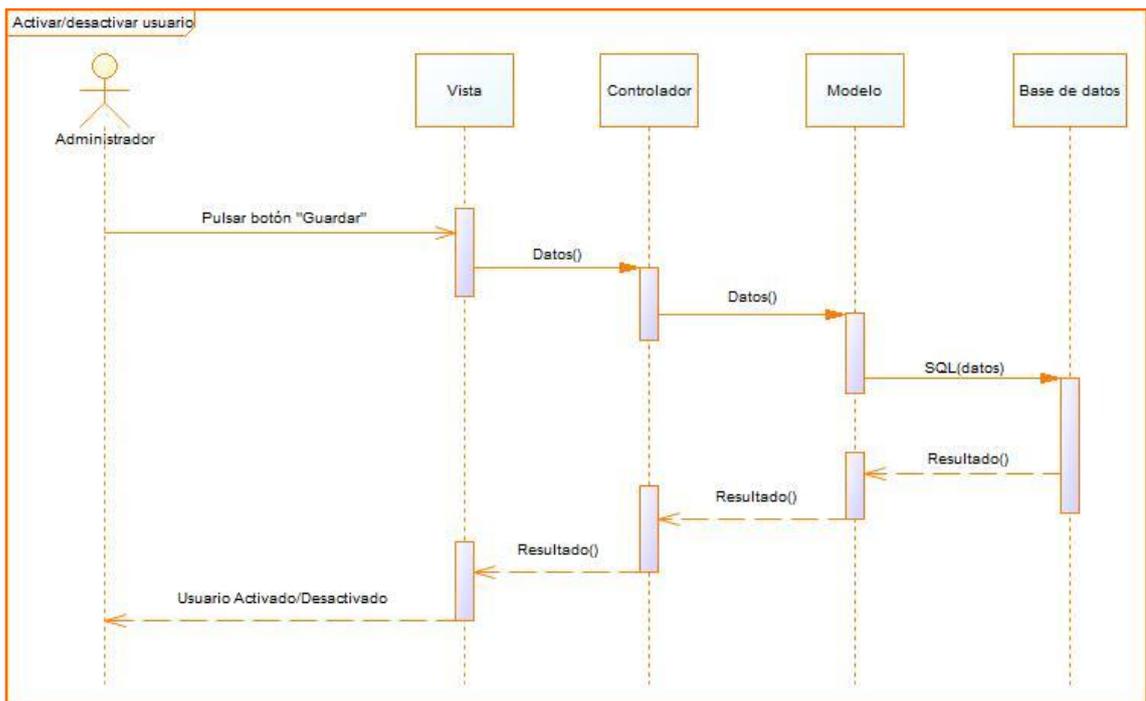


Figura 4.20: Diagrama de secuencia activar/desactivar usuario
Elaborado por: Rubén Jiménez – Investigador

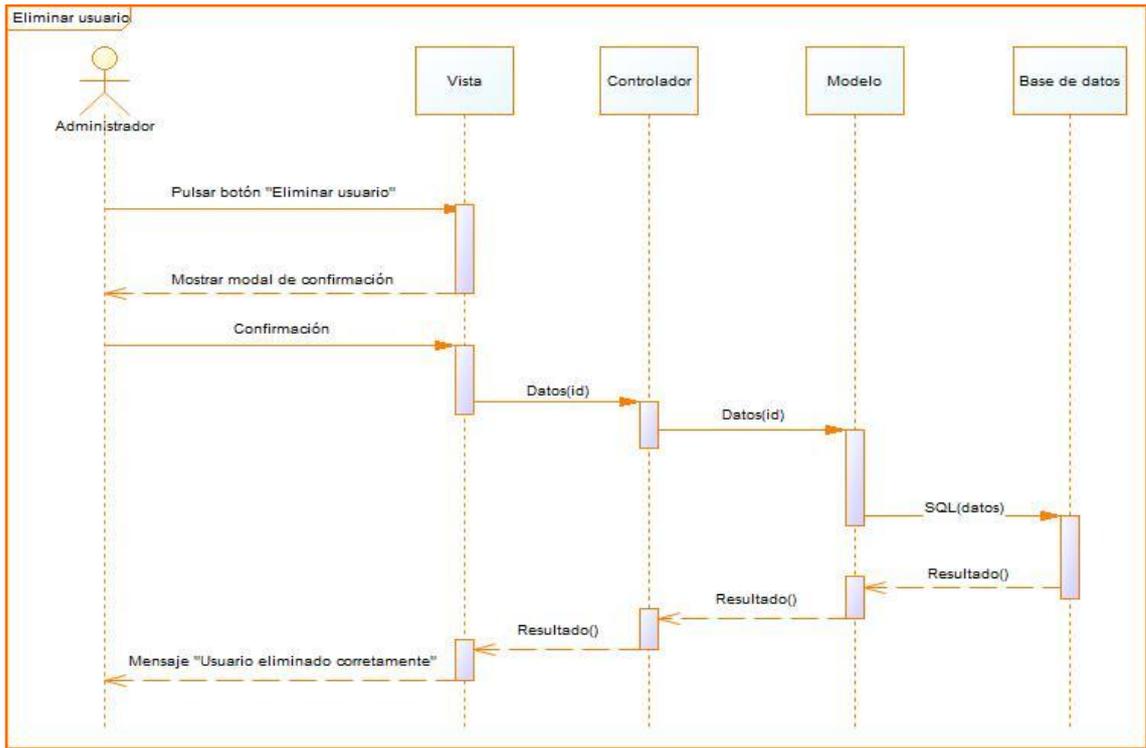


Figura 4.21: Diagrama de secuencia eliminar usuario
Elaborado por: Rubén Jiménez – Investigador

4.2.1.3 Diagrama caso de uso Cliente

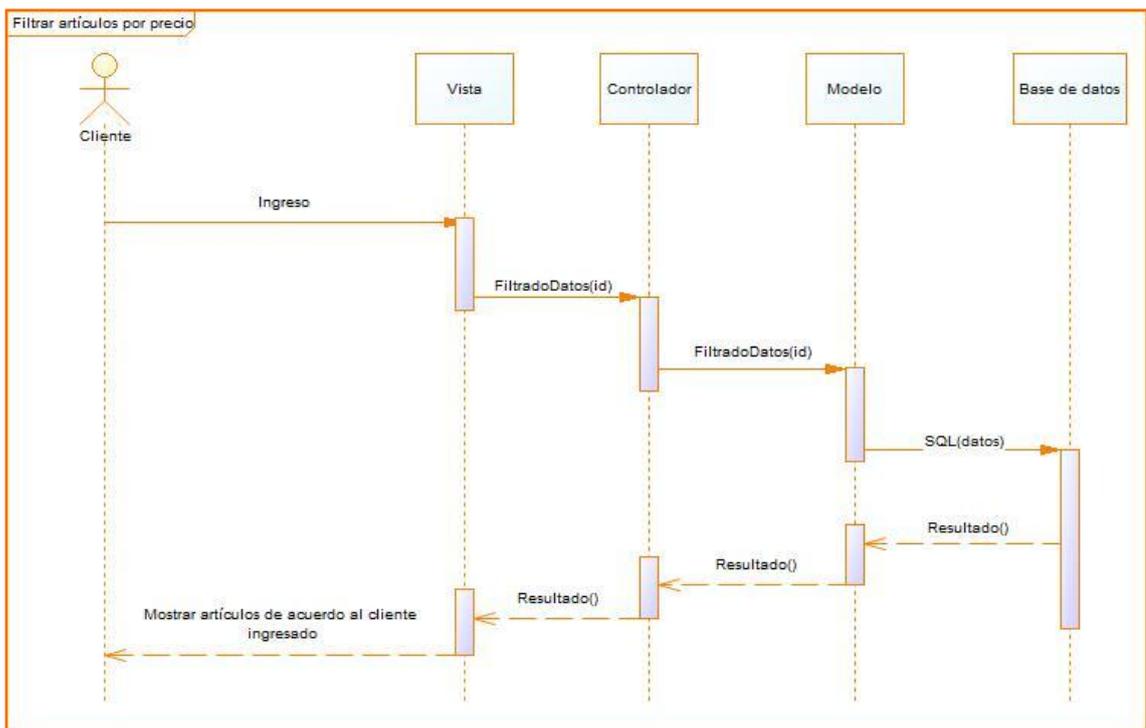


Figura 4.22: Diagrama de secuencia filtrar artículo por precio
Elaborado por: Rubén Jiménez – Investigador

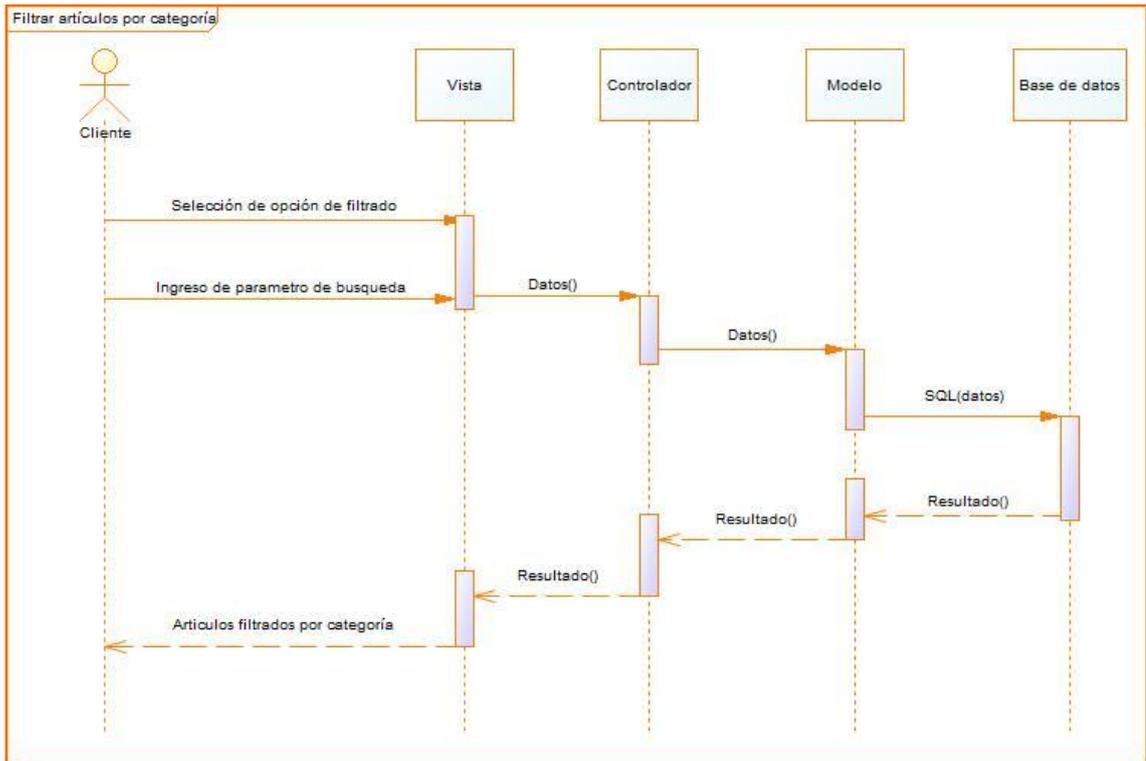


Figura 4.23: Diagrama de secuencia filtrar artículo por categoría
Elaborado por: Rubén Jiménez – Investigador

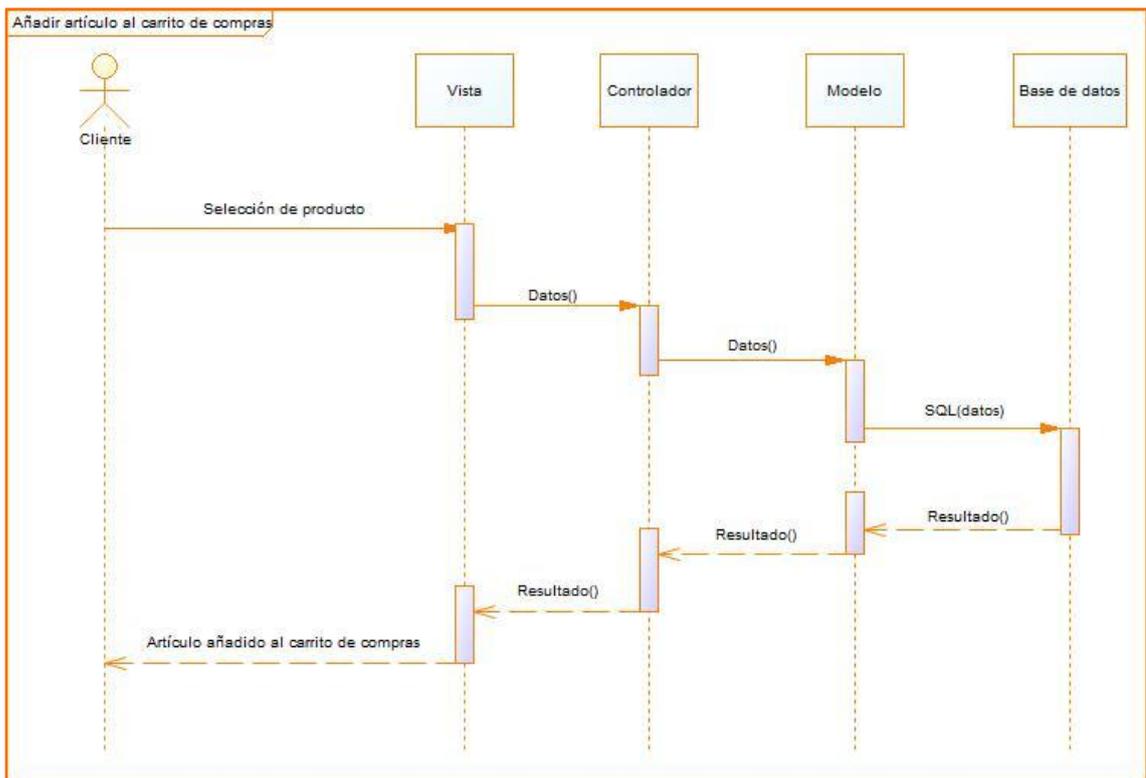


Figura 4.24: Diagrama de secuencia añadir artículos al carrito de compras
Elaborado por: Rubén Jiménez – Investigador

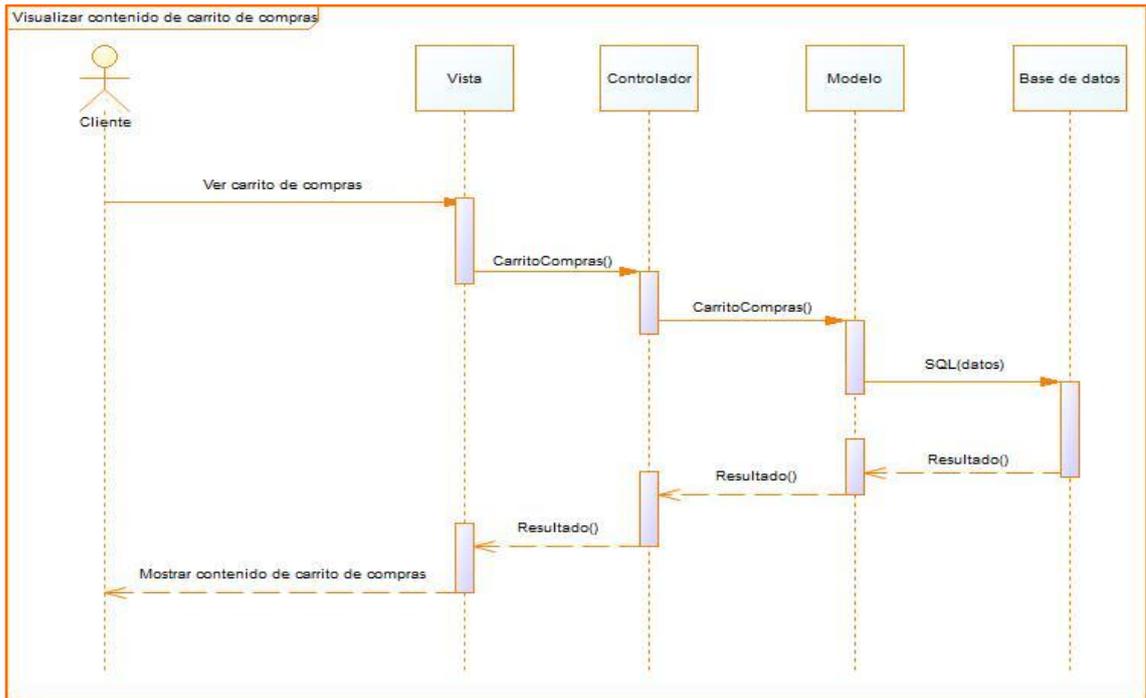


Figura 4.25: Diagrama de secuencia visualizar contenido carrito de compras
Elaborado por: Rubén Jiménez – Investigador

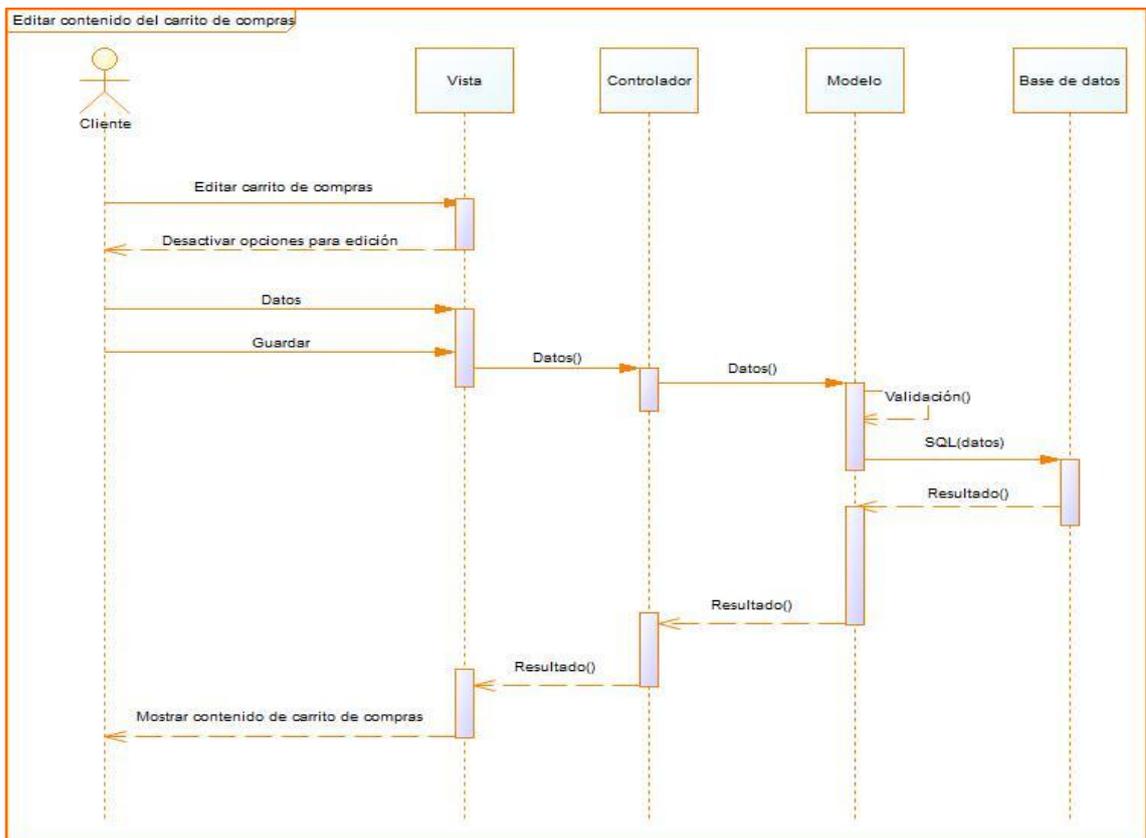


Figura 4.26: Diagrama de secuencia editar contenido del carrito de compras
Elaborado por: Rubén Jiménez – Investigador

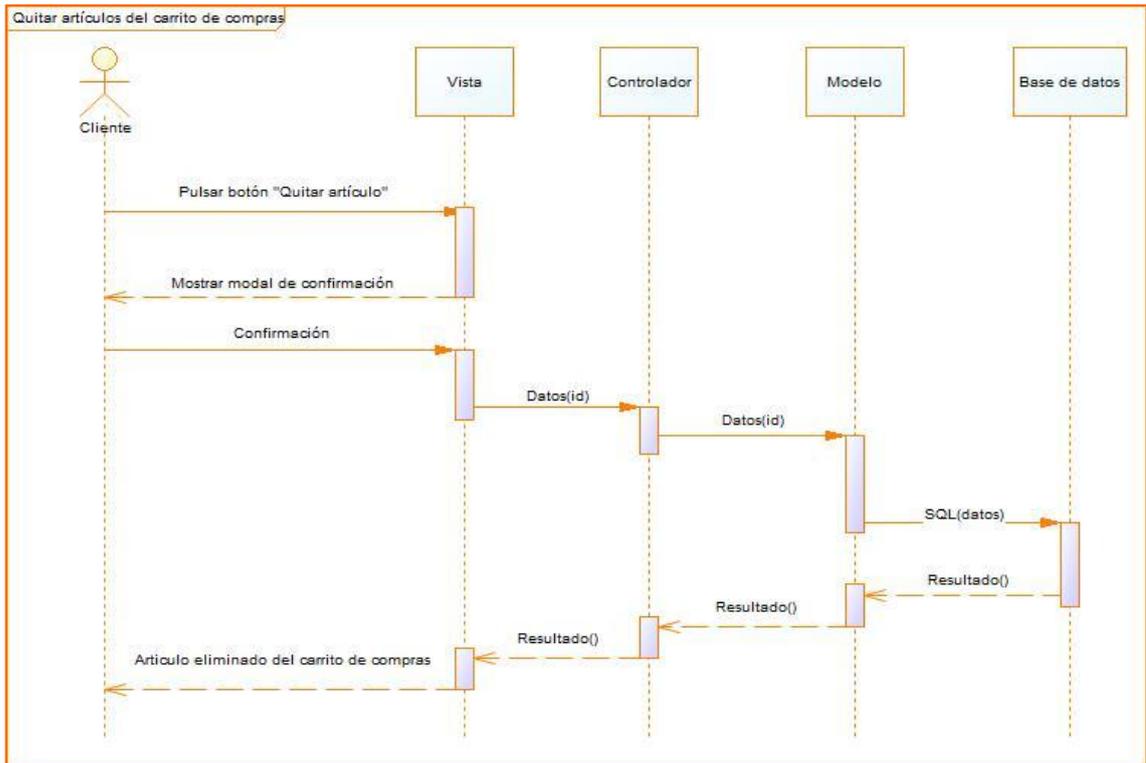


Figura 4.27: Diagrama de secuencia quitar artículos del carrito de compras
Elaborado por: Rubén Jiménez – Investigador

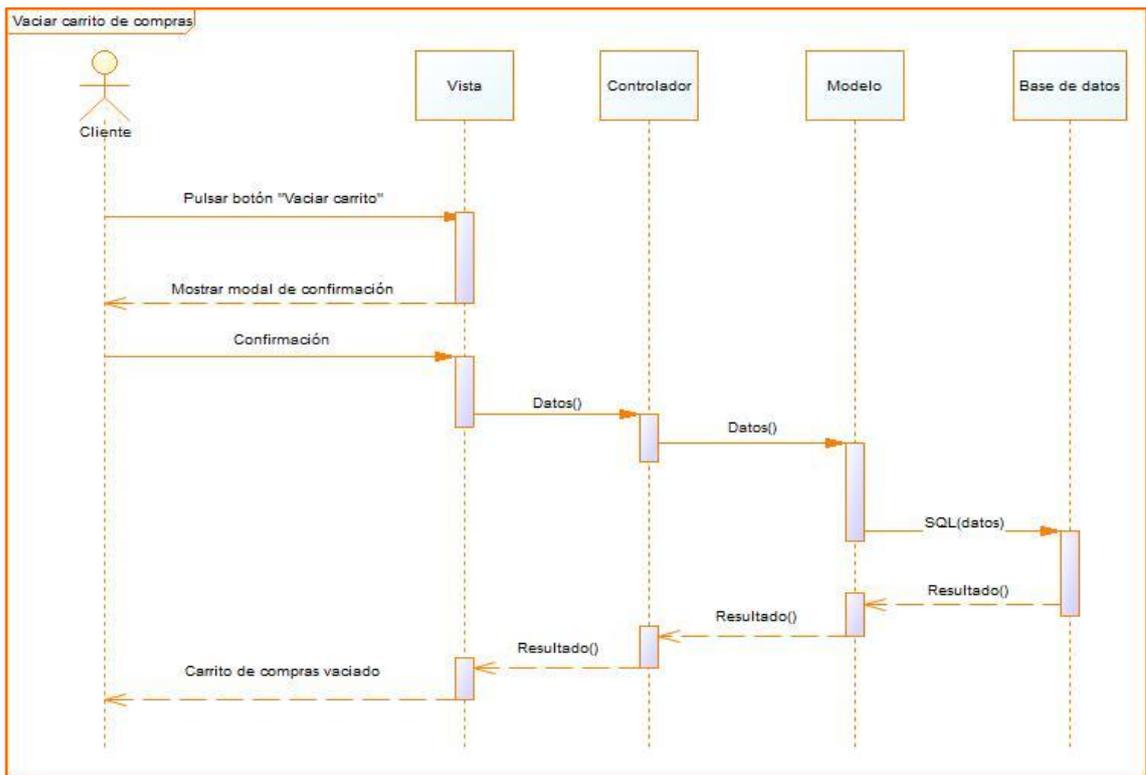


Figura 4.28: Diagrama de secuencia vaciar carrito de compras
Elaborado por: Rubén Jiménez – Investigador

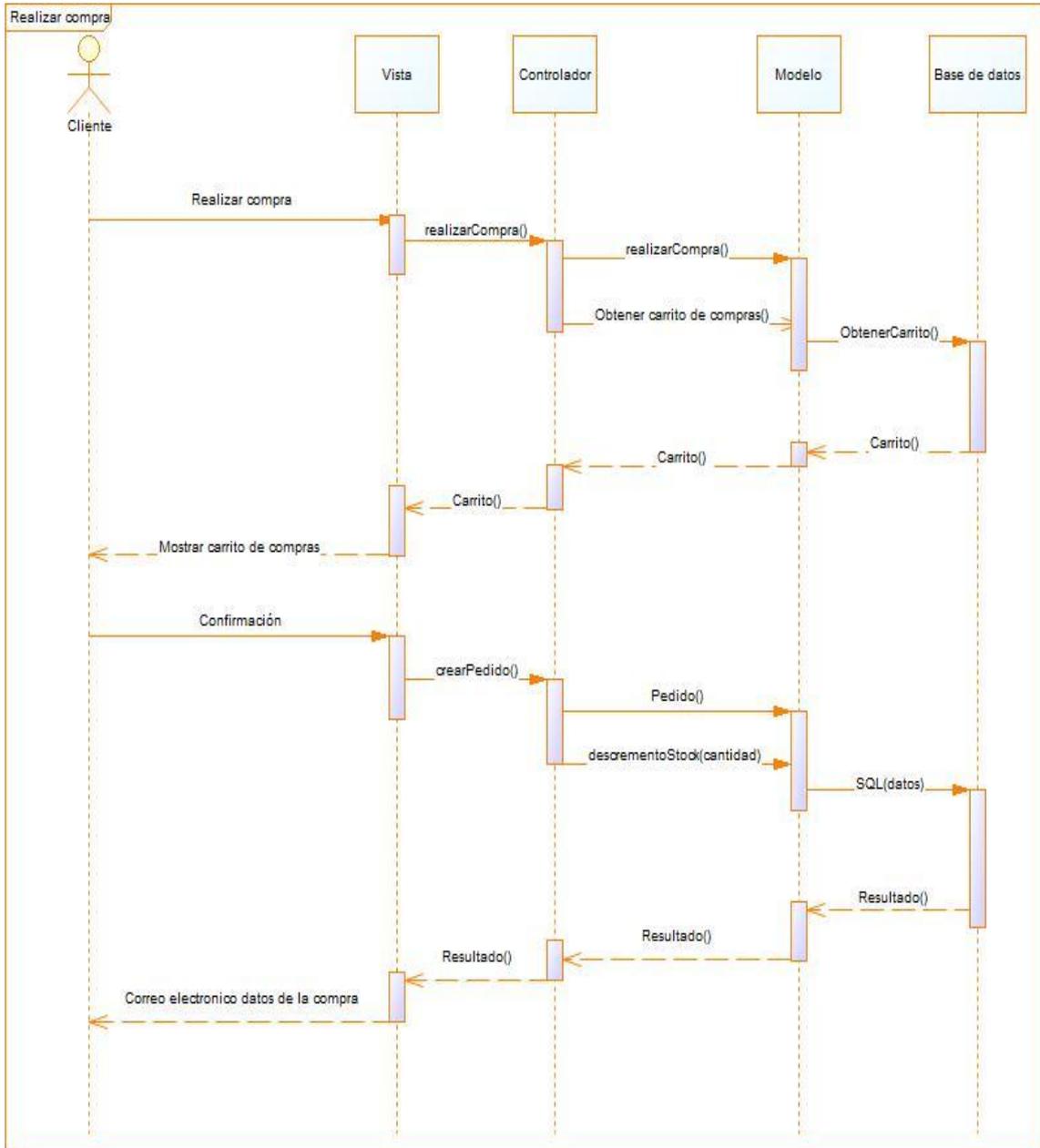


Figura 4.29: Diagrama de secuencia realizar compra
Elaborado por: Rubén Jiménez – Investigador

4.2.2 Diseño de la base de datos

4.2.2.1 Diseño lógico

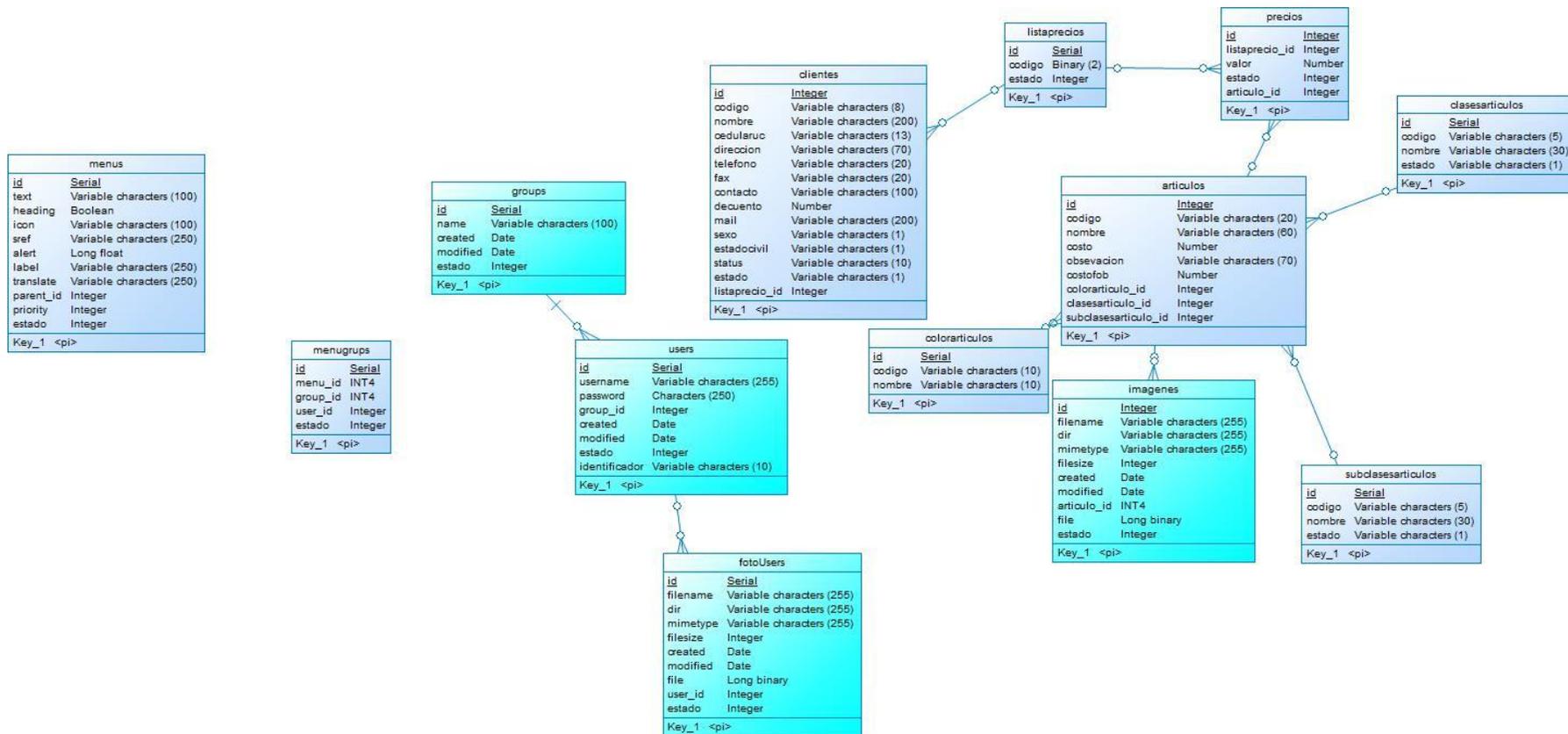


Figura 4.30: Diseño lógico de base de datos
Elaborado por: Rubén Jiménez – Investigador

4.2.2.2 Diseño físico

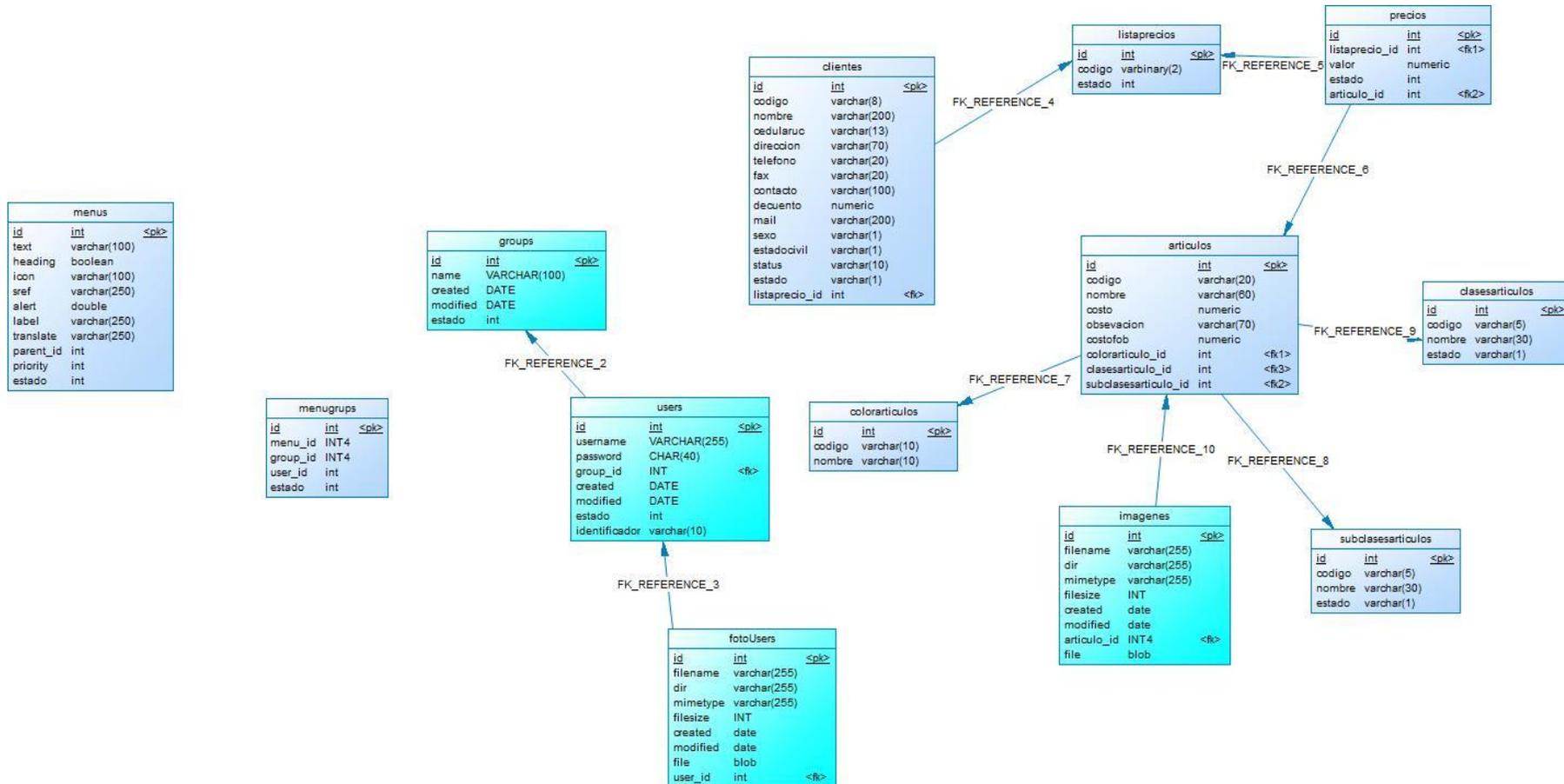


Figura 4.31: Diseño físico de base de datos
Elaborado por: Rubén Jiménez – Investigador

4.2.3 Diccionario de datos

Detalla las tablas almacenadas en la base de datos con sus respectivos campos. A continuación se ordenan de forma alfabética cada una de las tablas de la aplicación:

Tabla articulos

Campo	Tipo de dato	Tamaño	Restricción	Tabla relacionada
id	int		primary key	
codigo	varchar	20		
nombre	varchar	60		
costo	numeric			
observacion	varchar	70		
costofob	numeric			
colorarticulo_id	int		foreign key	coloresarticulos
clasearticulo_id	int		foreign key	clasesarticulos
subclasearticulo_id	int		foreign key	subclasesarticulos

Tabla 4.27: Diccionario de datos tabla articulos
Elaborado por: Rubén Jiménez – Investigador

Tabla clasesarticulos

Campo	Tipo de dato	Tamaño	Restricción	Tabla relacionada
Id	int		primary key	
Código	varchar	5		
nombre	varchar	30		
Estado	varchar	1		

Tabla 4.28: Diccionario de datos tabla clasesarticulos
Elaborado por: Rubén Jiménez – Investigador

Tabla clientes

Campo	Tipo de dato	Tamaño	Restricción	Tabla relacionada
id	int		primary key	
codigo	varchar	8		
nombre	varchar	200		
cedularuc	varchar	13		

direccion	varchar	70		
telefono	varchar	20		
fax	varchar	20		
contacto	varchar	100		
descuento	numeric			
mail	varchar	200		
sexo	varchar	1		
estadocivil	varchar	1		
status	varchar	10		
estado	varchar	10		
listaprecio_id	int		foreign key	listaprecios

Tabla 4.29: Diccionario de datos tabla clientes
Elaborado por: Rubén Jiménez – Investigador

Tabla colorarticulos

Campo	Tipo de dato	Tamaño	Restricción	Tabla relacionada
id	int		primary key	
codigo	varchar	10		
nombre	varchar	10		

Tabla 4.30: Diccionario de datos tabla colorarticulos
Elaborado por: Rubén Jiménez – Investigador

Tabla fotoUsers

Campo	Tipo de dato	Tamaño	Restricción	Tabla relacionada
id	int		primary key	
filename	varchar	255		
dir	varchar	255		
mimetype	varchar	255		
filesize	int			
created	date			
modified	date			
file	blob			
user_id	int		foreign key	users

Tabla 4.31: Diccionario de datos tabla fotoUsers
Elaborado por: Rubén Jiménez – Investigador

Tabla groups

Campo	Tipo de dato	Tamaño	Restricción	Tabla relacionada
id	int		primary key	
name	varchar	100		
created	date			
modified	date			
estado	int			

Tabla 4.32: Diccionario de datos tabla groups
Elaborado por: Rubén Jiménez – Investigador

Tabla imagenes

Campo	Tipo de dato	Tamaño	Restricción	Tabla relacionada
id	int		primary key	
filename	varchar	255		
dir	varchar	255		
mimetype	varchar	255		
filesize	int			
created	date			
modified	date			
file	blob			
articulo_id	int		foreign key	articulos

Tabla 4.33: Diccionario de datos tabla imagenes
Elaborado por: Rubén Jiménez – Investigador

Tabla listaprecios

Campo	Tipo de dato	Tamaño	Restricción	Tabla relacionada
id	int		primary key	
codigo	varbinary	2		
estado	int			

Tabla 4.34: Diccionario de datos tabla listaprecios
Elaborado por: Rubén Jiménez – Investigador

Tabla menus

Campo	Tipo de dato	Tamaño	Restricción	Tabla relacionada
id	int		primary key	
text	varchar	100		
heading	boolean			
icon	varchar	100		
sref	varchar	250		
alert	double			
label	varchar	250		
translate	varchar	250		
parent_id	Int			
priority	int			
estado	int			

Tabla 4.35: Diccionario de datos tabla menus
Elaborado por: Rubén Jiménez – Investigador

Tabla menugrups

Campo	Tipo de dato	Tamaño	Restricción	Tabla relacionada
id	int		primary key	
menu_id	int4		foreign key	menus
group_id	int4		foreign key	groups
user_id	int		foreign key	users
estado	int			

Tabla 4.36: Diccionario de datos tabla menugrups
Elaborado por: Rubén Jiménez – Investigador

Tabla precios

Campo	Tipo de dato	Tamaño	Restricción	Tabla relacionada
id	int		primary key	
listaprecio_id	int		foreign key	listaprecios
valor	numeric			
estado	int			
articulo_id	int		foreign key	artículos

Tabla 4.37: Diccionario de datos tabla precios
Elaborado por: Rubén Jiménez – Investigador

Tabla subclassesarticulos

Campo	Tipo de dato	Tamaño	Restricción	Tabla relacionada
id	int		primary key	
codigo	varchar	5		
nombre	varchar	30		
estado	varchar	1		

Tabla 4.38: Diccionario de datos tabla subclassesarticulos
Elaborado por: Rubén Jiménez – Investigador

Tabla users

Campo	Tipo de dato	Tamaño	Restricción	Tabla relacionada
id	int		primary key	
username	varchar	255		
password	varchar	40		
group_id	int		foreign key	groups
created	date			
Modified	date			
Estado	int			
identificador	varchar	10		

Tabla 4.39: Diccionario de datos tabla users
Elaborado por: Rubén Jiménez – Investigador

4.2.4 Diseño previo de la interfaz de usuario

El diseño de la interfaz de usuario define la ubicación de los controles y proporciona una idea previa del entorno visual en que se presentará la información en la aplicación antes de que sea implementada mediante una herramienta de desarrollo.

4.2.4.1 Interfaz principal usuarios anónimos

Página de inicio presentada a todos los usuarios que no han iniciado sesión en la aplicación.

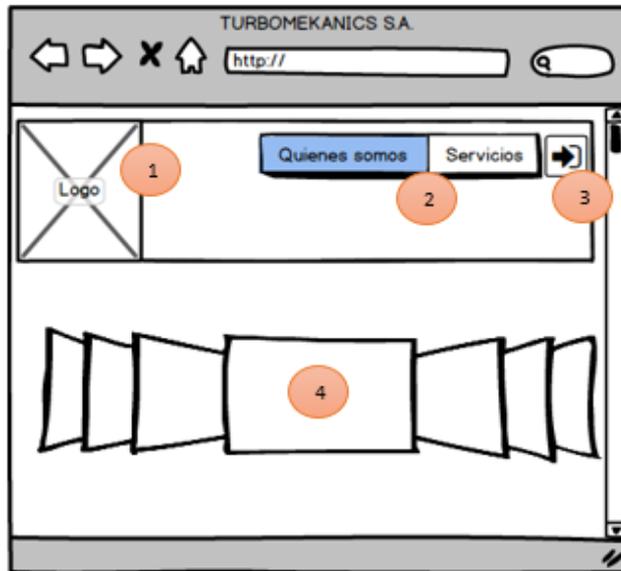


Figura 4.32: Interfaz principal usuarios anónimos
Elaborado por: Rubén Jiménez – Investigador

- 1) **Logo:** imagen representativa de la empresa.
- 2) **Información:** Información de la empresa y los servicios que brinda a los clientes.
- 3) **Botón de inicio de sesión:** botón que direcciona a la página de login.
- 4) **Slide de imágenes:** imágenes de los productos que oferta la empresa.

4.2.4.2 Interfaz de inicio de sesión

Permite al usuario ingresar al escribir su nombre de usuario y contraseña.

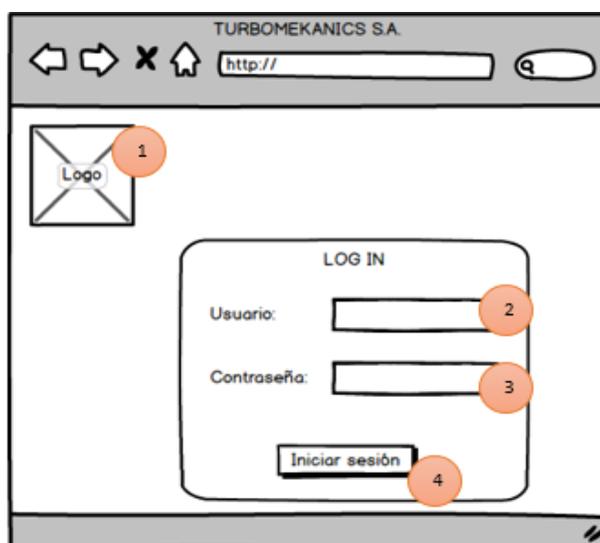


Figura 4.33: Interfaz de inicio de sesión
Elaborado por: Rubén Jiménez – Investigador

- 1) **Logo:** imagen representativa de la empresa.
- 2) **Usuario:** cuadro de texto en el que el usuario debe escribir su nombre de usuario.
- 3) **Contraseña:** cuadro de texto en el que el usuario debe escribir su contraseña.
- 4) **Botón de inicio de sesión:** botón que verifica los datos y permite el acceso al sistema.

4.2.4.3 Interfaz página principal

Pantalla mostrada a los usuarios una vez iniciada sesión.

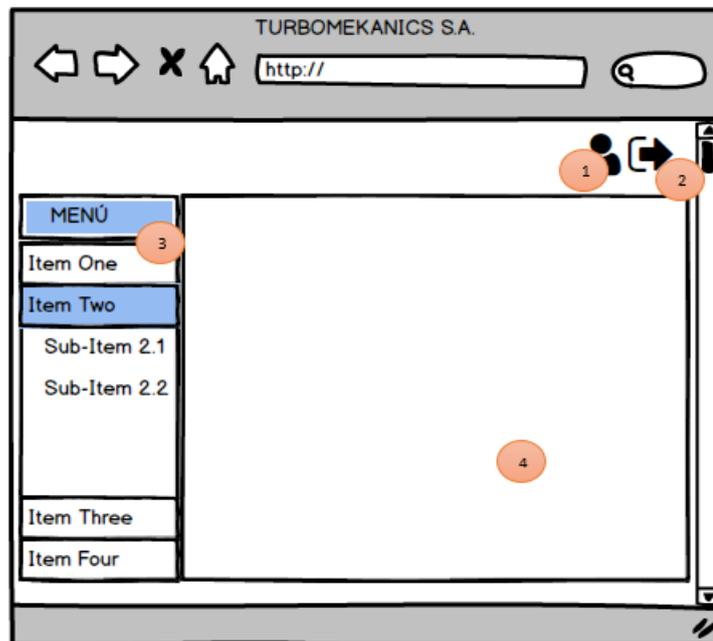


Figura 4.34: Interfaz página principal
Elaborado por: Rubén Jiménez – Investigador

- 1) **Usuario:** imagen del usuario que ha iniciado sesión.
- 2) **Botón de cerrar sesión:** permite al usuario salir del sistema.
- 3) **Menú:** menú con opciones e información de acuerdo al usuario que ha iniciado sesión.
- 4) **Área de trabajo:** espacio que permite la visualización y/o manipulación de información correspondiente a la opción elegida en el menú.

4.2.4.4 Interfaz de perfil de usuario

Ventana para la consulta y administración de la información de usuario.

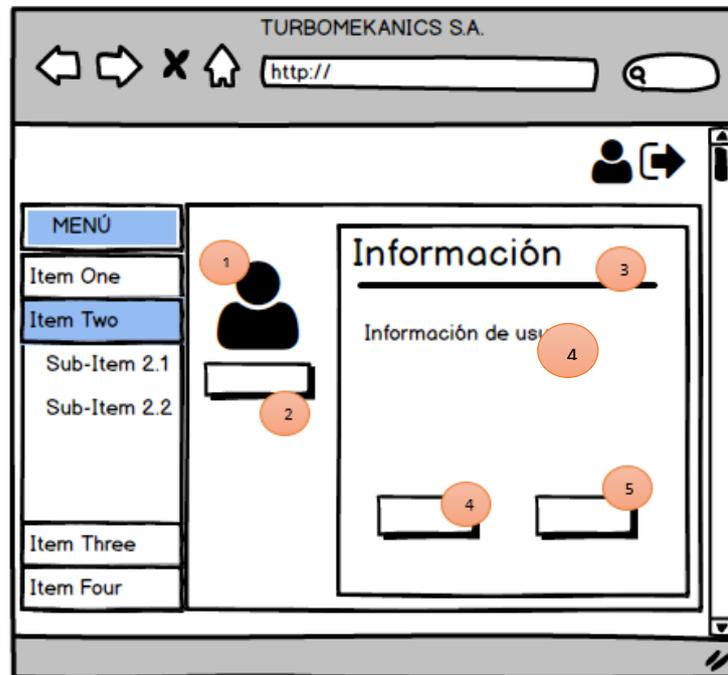


Figura 4.35: Interfaz perfil de usuario
Elaborado por: Rubén Jiménez – Investigador

- 1) **Usuario:** imagen del usuario.
- 2) **Botón de editar imagen:** despliega una ventana modal para cargar imágenes desde el ordenador.
- 3) **Información:** información modificable del usuario.
- 4) **Botón guardar:** valida la información y guarda los datos en la base de datos.
- 5) **Botón cancelar:** permite salir a la pantalla principal sin guardar los cambios.

4.2.4.5 Interfaz ingreso de usuarios

Ventana modal para el ingreso de usuarios.

El diagrama muestra una ventana modal con el título "TÍTULO" en la parte superior. Debajo del título hay tres campos de texto etiquetados como "Texto 1:", "Texto 2:" y "Texto 3:". En la parte inferior de la ventana hay dos botones: "Guardar" a la izquierda y "Cancelar" a la derecha. Seis círculos numerados (1 a 6) indican puntos de interés: 1) el título, 2) el campo de texto "Texto 2", 3) el botón "Guardar", 4) el botón "Cancelar", 5) el borde derecho de la ventana, y 6) el campo de texto "Texto 2" (repetido).

Figura 4.36: Interfaz ingreso de usuario
Elaborado por: Rubén Jiménez – Investigador

- 1) **Título:** título de la ventana.
- 2) **Área de datos:** controles para ingreso y edición de datos del usuario.
- 3) **Botón guardar:** valida la información y guarda los datos en la base de datos.
- 4) **Botón cancelar:** permite salir a la pantalla principal sin guardar los cambios.

4.2.4.6 Interfaz eliminar usuario

Ventana modal que confirma la eliminación de la información en el proceso seleccionado

El diagrama muestra una ventana modal con el título "Eliminar" en la parte superior. Debajo del título hay una pregunta: "Desea eliminar usuario?". En la parte inferior de la ventana hay dos botones: "No" a la izquierda y "Si" a la derecha.

Figura 4.37: Interfaz eliminar usuario

- 1) **Texto:** información sobre el registro que se va a eliminar.
- 2) **No:** cancela la acción.
- 3) **Si:** confirma la acción.

4.2.4.7 Interfaz productos

Interfaz que muestra al usuario los productos existentes en el sistema.

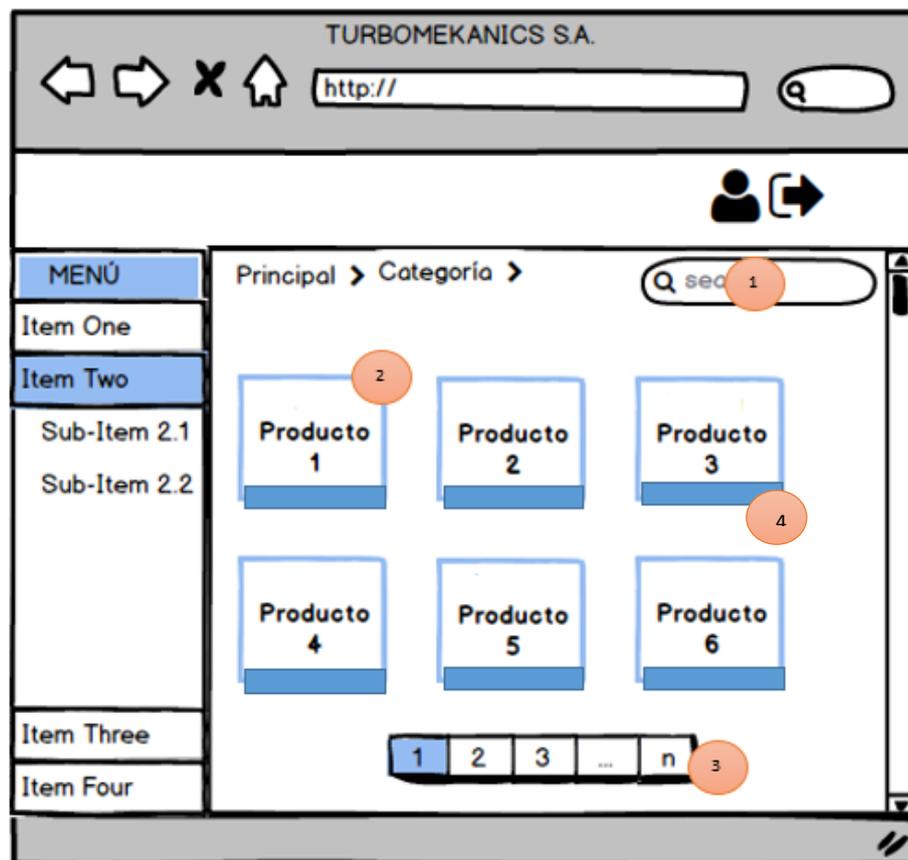


Figura 4.38: Interfaz productos
Elaborado por: Rubén Jiménez – Investigador

- 1) **Barra de búsqueda:** Filtra los datos.
- 2) **Imagen del producto.**
- 3) **Botón ver información:** despliega una ventana modal con la información del producto.
- 4) **Paginador:** permite navegar en más productos ingresados.

4.2.4.8 Interfaz información de producto

Información detallada del producto con el precio filtrado de acuerdo al cliente que inició sesión.

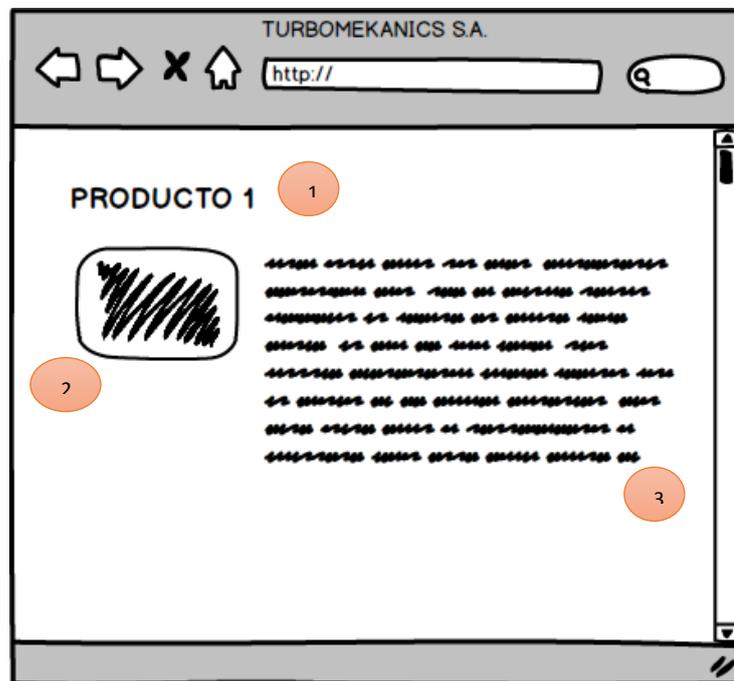


Figura 4.39: Interfaz información del producto
Elaborado por: Rubén Jiménez – Investigador

- 1) **Producto:** nombre del producto.
- 2) **Imagen:** foto activa del producto.
- 3) **Información:** datos del producto filtrados de acuerdo al cliente que ha iniciado sesión.

4.2.4.9 Interfaz carrito de compras

Información de los productos seleccionados para la compra con la posibilidad de editar la misma.

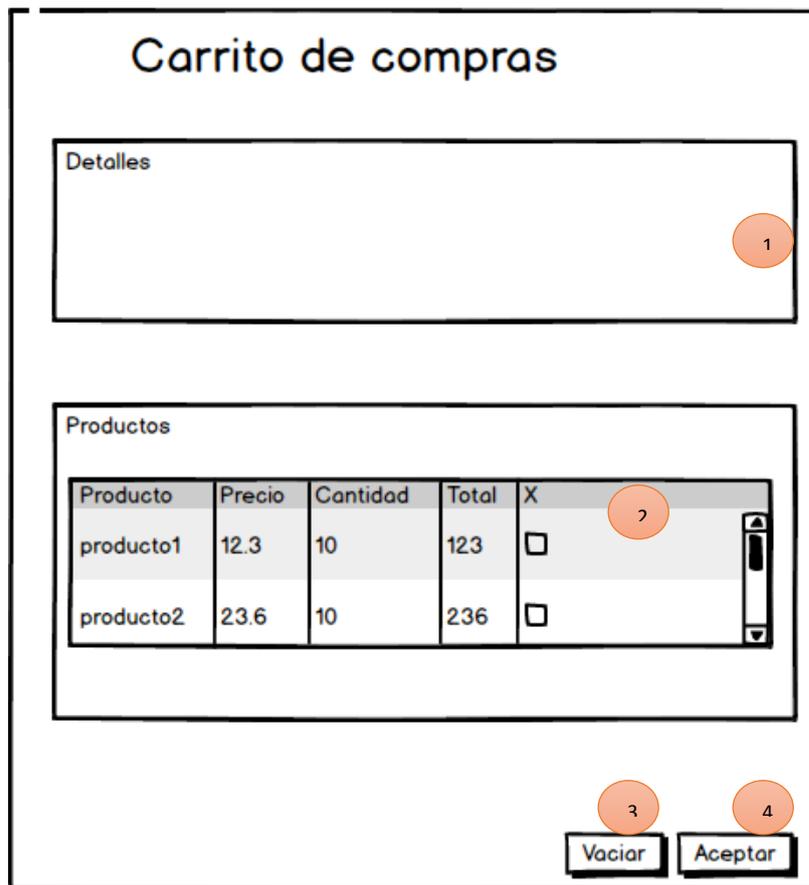


Figura 4.40: Interfaz información del producto
Elaborado por: Rubén Jiménez – Investigador

- 1) **Detalles:** información completa de la compra.
- 2) **Productos:** información de productos con opciones como eliminar producto.
- 3) **Botón vaciar:** elimina todos los productos del carro de compras.
- 4) **Botón aceptar:** envía un correo electrónico al cliente para informarle de la compra realizada.

4.3 Construcción

Se completa la funcionalidad del sistema siguiendo la arquitectura con código listo para la producción.

4.3.1 Implementación del servidor web

Se definió utilizar Wampserver en este proyecto para servidor de la aplicación. La instalación del Wampserver se detalla en el Anexo 2.

Wampserver es un entorno de desarrollo web gratuito que permite la creación de aplicaciones usando las siguientes herramientas:

- Windows: sistema operativo.
- Apache: servidor web.
- MySQL: base de datos.
- PHP: lenguaje de programación.

WAMP permite desarrollar y gestionar páginas HTML de forma local.

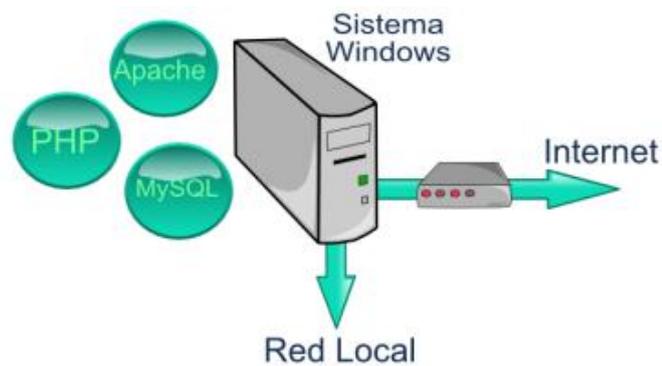


Figura 4.41: Esquema del servidor WAMP

Fuente: [30]

Dentro de Wampserver se encuentra el servidor web apache.

“El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual.” [31]

Apache presenta varias ventajas entre las que destacan:

- Usado por una gran comunidad de usuarios por lo que es posible encontrar soporte fácilmente.
- Software de código abierto por lo que se actualiza, mejora y extiende su funcionalidad constantemente.
- Es totalmente gratuito.
- Portable y multiplataforma, puede ser instalado en todos los sistemas operativos.
- Modular, que aportan diversas funcionalidades.

4.3.2 Implementación de la base de datos

Para la implementación de la base de datos se realizó el diagrama utilizando la herramienta de diseño Power Designer 16.5.

Para generar el script .sql se siguen los siguientes pasos:

1. Barra de herramientas opción Database.
2. Generate Database

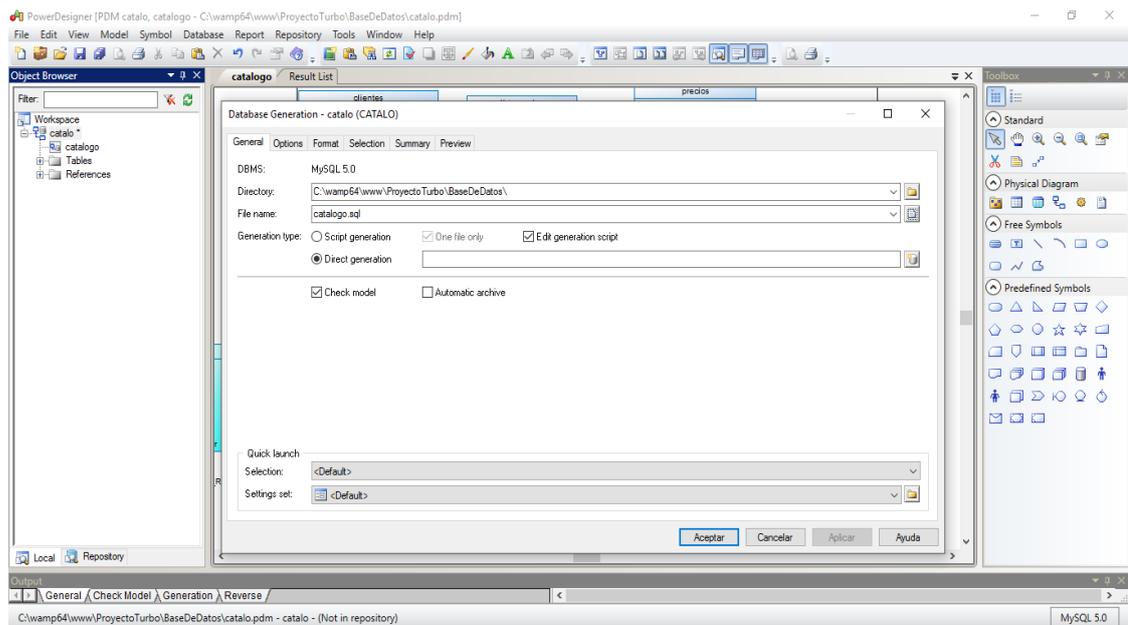


Figura 4.42: Generación de sql de base de datos
Elaborado por: Rubén Jiménez – Investigador

El código SQL generado se visualiza en el Anexo 3.

Una vez generado el script sql se utiliza la herramienta visual para administración y mantenimiento de base de datos MySQL Workbench 6.1. cuya instalación se encuentra detallado en el Anexo 4.

“MySQL Workbench es una herramienta visual unificada para arquitectos de base de datos, desarrolladores y administradores de bases de datos. MySQL Workbench ofrece modelado de datos, desarrollo de SQL y herramientas completas de administración para la configuración de servidores, administración de usuarios, backup y mucho más. MySQL Workbench está disponible en Windows, Linux y Mac OS X.” [32]

Se crea un nuevo esquema:

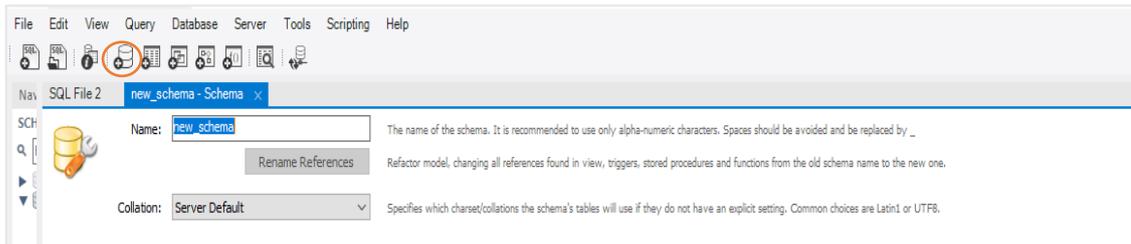


Figura 4.43: Implementación de nuevo esquema
Elaborado por: Rubén Jiménez – Investigador

Abrir el archivo SQL:

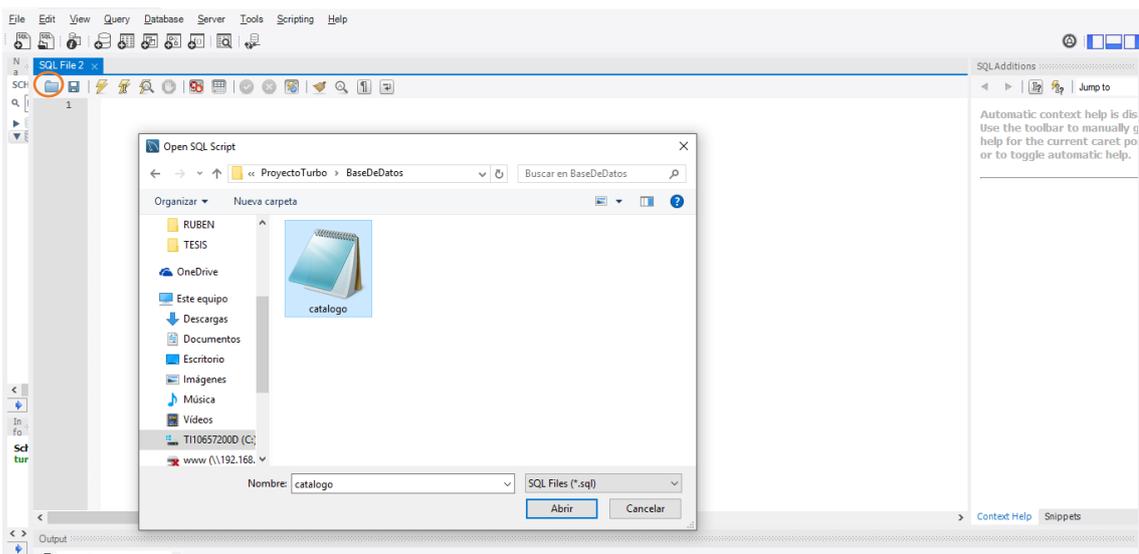


Figura 4.44: Abrir script sql
Elaborado por: Rubén Jiménez – Investigador

Y ejecutar

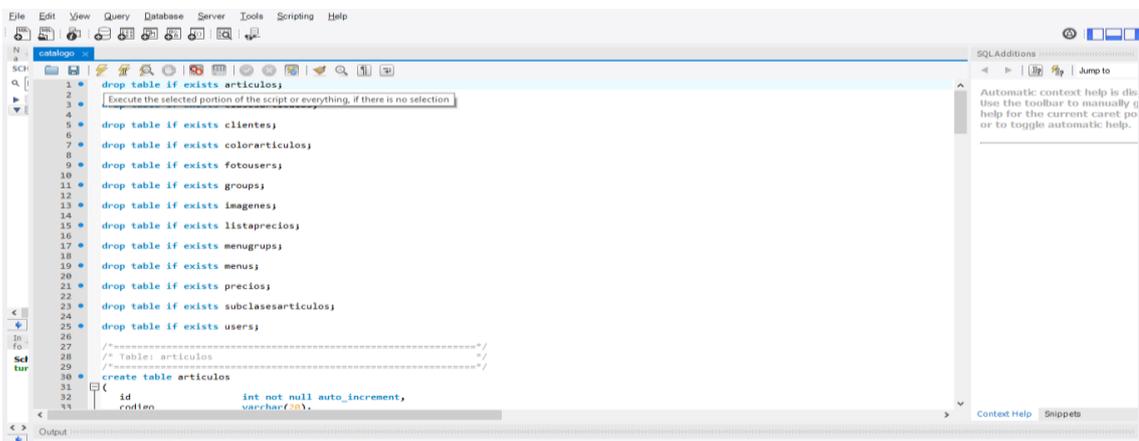


Figura 4.45: Abrir script sql
Elaborado por: Rubén Jiménez – Investigador

4.3.3 Implementación del código fuente de la aplicación

4.3.3.1 Configuración de entorno

En el desarrollo web se distinguen dos tipos de tecnologías, las que trabajan en el lado del servidor (back-end) que constituye el código que se encarga de interactuar con el servidor para manejar los datos y las que trabajan en el lado del cliente (front-end) que constituye el código que presenta la información definiendo el cómodo uso para el usuario.

Para optimizar y agilizar el trabajo en el proyecto, se seleccionaron frameworks (marcos de desarrollo) que utilizan la arquitectura de programación Modelo Vista Controlador tanto para el back-end como el front-end.

Back-end

El framework de desarrollo escogido para el back-end es Cakephp, que permite trabajar en el lado del servidor.

CakePHP es un framework de código libre (open source) con licencia Licencia MIT que constituye una estructura base flexible para la creación de aplicaciones web. Tiene como objetivo permitir a los programadores centrarse en la lógica de la aplicación mientras trabaja de una manera estructurada.

Las principales características de este framework son las siguientes:

- CRUD (creación, lectura, actualización, eliminación) integrado para la interacción con la base de datos.
- Amplia comunidad de ayuda.
- Generación de código.
- Compatibilidad con PHP4 y PHP5.
- Poca configuración para su uso.
- Incluye clases y objetos php para controladores y modelos, y plantillas html para las vistas que proporcionan reusabilidad y extensibilidad de código.

- Validación integrada.
- Facilidad para la utilización de ayudantes (helpers) para AJAX, Javascript, formularios HTML, entre otros mediante plantillas rápidas y flexibles.
- Despachador de peticiones con rutas limpias personalizadas.

Front-end

De igual manera el front-end será realizado haciendo uso del framework javascript AngularJS que amplía el HTML definiendo las animaciones, efectos y permitiendo la sincronización de la vista con el modelo mejorando el rendimiento.

“AngularJS es un framework Javascript open-source desarrollado por Google.” [33]. Permite construir aplicaciones web interactivas de manera rápida.

Las principales características de este framework son las siguientes:

- Utiliza el patrón MVC.
- Extiende HTML con directivas propias.
- Compatibilidad con todos los navegadores de última generación.
- Inyección de dependencias.

4.3.3.2 Implementación de la interfaz gráfica

Se muestra a continuación algunas de las interfaces de la aplicación que han sido implementadas en base a lo planteado en la etapa de diseño haciendo uso del framework Bootstrap.

“Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice.” [34]

Este framework simplifica la creación de aplicaciones web combinando Javascript y CSS para obtener un diseño responsivo; es decir, que la interfaz se podrá adaptar a los navegadores en donde se la visualice de manera automática.

El principio de Bootstrap ubica cada uno de los componentes de la siguiente manera:

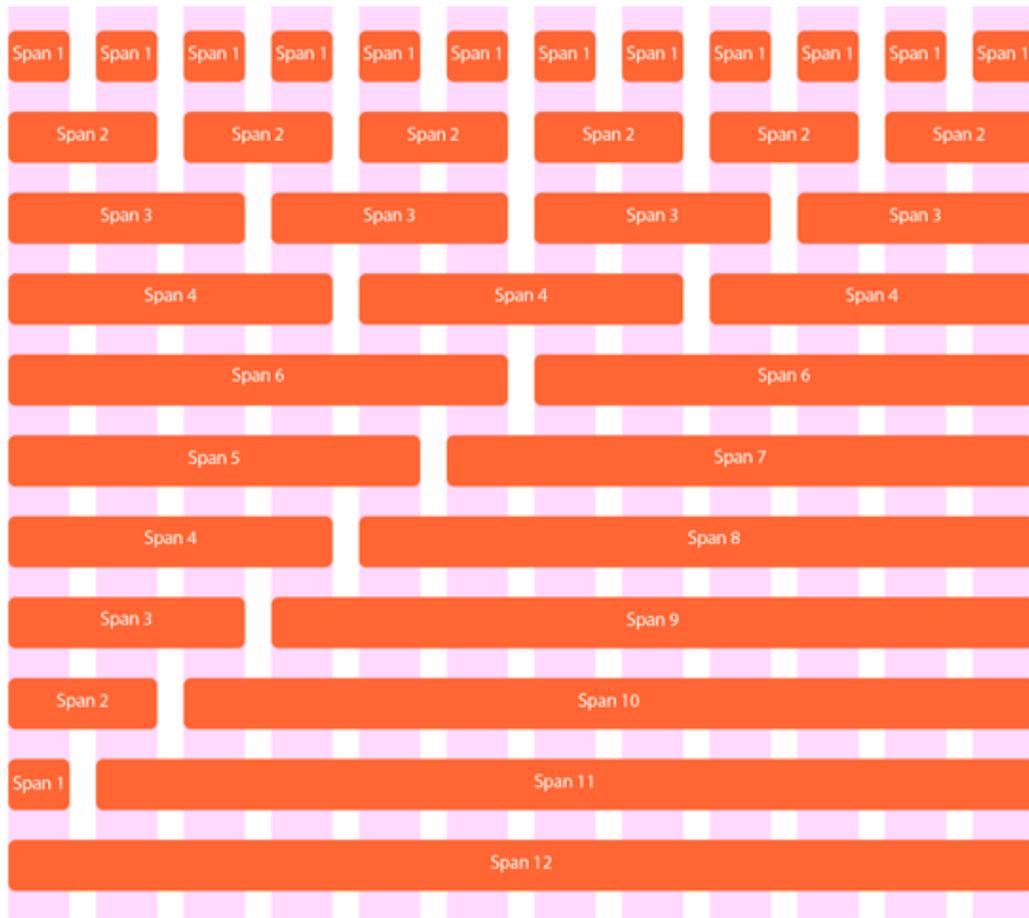


Figura 4.46: Esquema de bootstrap
Fuente: [35]

El esquema de trabajo de Bootstrap se basa en un potente sistema de columnas (grids) que divide la pantalla en doce partes con cuatro rangos de resoluciones, de esta manera un elemento se ajusta a la pantalla.

4.3.3.3 Implementación de la arquitectura MVC

Modelo

La convención de nombres de las clases de los modelos son en plural en formato CamelCase, y terminan en Table.php, ejemplo: UsersTable.php.

El siguiente fragmento de código corresponde al modelo UsersTable.php y consiste en una función que permite la conexión con la tabla que almacena la información de usuarios y la envía hacia el controlador. Además se encuentran en el modelo validaciones necesarias para que los datos sean filtrados correctamente:

```

class UsersTable extends Table
{
    /**
     * Initialize method
     *
     * @param array $config The configuration for the Table.
     * @return void
     */
    public function initialize(array $config)
    {
        parent::initialize($config);

        $this->table('users');
        $this->displayField('id');
        $this->primaryKey('id');

        $this->addBehavior('Timestamp');

        $this->belongsTo('Groups', [
            'foreignKey' => 'group_id',
            'joinType' => 'INNER'
        ]);
        $this->hasMany('Fotousers', [
            'foreignKey' => 'user_id'
        ]);

        $this->addBehavior('Acl.Acl', ['type' => 'requester']);
    }

    /**
     * Default validation rules.
     *
     * @param \Cake\Validation\Validator $validator Validator instance.
     * @return \Cake\Validation\Validator
     */
    public function validationDefault(Validator $validator)
    {
        $validator
            ->integer('id')
            ->allowEmpty('id', 'create');

        $validator
            ->requirePresence('username', 'create')
            ->notEmpty('username');

        $validator
            ->requirePresence('password', 'create')
            ->notEmpty('password');

        $validator
            ->integer('estado')
            ->allowEmpty('estado');

        $validator
            ->allowEmpty('identificador');
    }
}

```

```

        return $validator;
    }

    /**
     * Returns a rules checker object that will be used for validating
     * application integrity.
     *
     * @param \Cake\ORM\RulesChecker $rules The rules object to be modified.
     * @return \Cake\ORM\RulesChecker
     */
    public function buildRules(RulesChecker $rules)
    {
        $rules->add($rules->isUnique(['username']));
        $rules->add($rules->existsIn(['group_id'], 'Groups'));

        return $rules;
    }

    public function beforeSave(\Cake\Event\Event $event, \Cake\ORM\Entity
    $entity,
    \ArrayObject $options)
    {
        $hasher = new DefaultPasswordHasher;
        $entity->password = $hasher->hash($entity->password);
        return true;
    }
}

```

Vista

El siguiente fragmento de código corresponde a la vista index.ctp de usuarios y representa la interfaz de usuarios con sus campos y operaciones que se pueden realizar sobre la tabla:

```

<nav class="large-3 medium-4 columns" id="actions-sidebar">
    <ul class="side-nav">
        <li class="heading"><?=__('Actions') ?></li>
        <li><?=$this->Html->link(__('New User'), ['action' => 'add'])
        ?></li>
        <li><?=$this->Html->link(__('List Groups'), ['controller' =>
        'Groups', 'action' => 'index']) ?></li>
        <li><?=$this->Html->link(__('New Group'), ['controller' => 'Groups',
        'action' => 'add']) ?></li>
        <li><?=$this->Html->link(__('List Fotousers'), ['controller' =>
        'Fotousers', 'action' => 'index']) ?></li>
        <li><?=$this->Html->link(__('New Fotouser'), ['controller' =>
        'Fotousers', 'action' => 'add']) ?></li>
    </ul>
</nav>
<div class="users index large-9 medium-8 columns content">
    <h3><?=__('Users') ?></h3>

```

```

<table cellpadding="0" cellspacing="0">
  <thead>
    <tr>
      <th scope="col"><?=$this->Paginator->sort('id') ?></th>
      <th scope="col"><?=$this->Paginator->sort('username')
?></th>
      <th scope="col"><?=$this->Paginator->sort('password')
?></th>
      <th scope="col"><?=$this->Paginator->sort('group_id')
?></th>
      <th scope="col"><?=$this->Paginator->sort('created') ?></th>
      <th scope="col"><?=$this->Paginator->sort('modified')
?></th>
      <th scope="col"><?=$this->Paginator->sort('estado') ?></th>
      <th scope="col"><?=$this->Paginator->sort('identificador')
?></th>
      <th scope="col" class="actions"><?=__('Actions') ?></th>
    </tr>
  </thead>
  <tbody>
    <?php foreach ($users as $user): ?>
      <tr>
        <td><?=$this->Number->format($user->id) ?></td>
        <td><?=$this->Html->link($user->group-
>name, ['controller' => 'Groups', 'action' => 'view', $user->group->id]) : ''
?></td>
        <td><?=$this->Html->link($user->group-
>name, ['controller' => 'Groups', 'action' => 'view', $user->group->id]) : ''
?></td>
        <td><?=$this->Html->link($user->group-
>name, ['controller' => 'Groups', 'action' => 'view', $user->group->id]) : ''
?></td>
        <td><?=$this->Html->link($user->group-
>name, ['controller' => 'Groups', 'action' => 'view', $user->group->id]) : ''
?></td>
        <td><?=$this->Html->link($user->group-
>name, ['controller' => 'Groups', 'action' => 'view', $user->group->id]) : ''
?></td>
        <td><?=$this->Html->link($user->group-
>name, ['controller' => 'Groups', 'action' => 'view', $user->group->id]) : ''
?></td>
        <td class="actions">
          <?=$this->Html->link(__('View'), ['action' => 'view',
$user->id]) ?>
          <?=$this->Html->link(__('Edit'), ['action' => 'edit',
$user->id]) ?>
          <?=$this->Form->postLink(__('Delete'), ['action' =>
'delete', $user->id], ['confirm' => __('Are you sure you want to delete #
{0}?', $user->id)]) ?>
        </td>
      </tr>
    <?php endforeach; ?>
  </tbody>
</table>
<div class="paginator">
  <ul class="pagination">
    <?=$this->Paginator->prev('< ' . __('previous')) ?>
    <?=$this->Paginator->numbers() ?>
    <?=$this->Paginator->next(__('next') . ' >') ?>
  </ul>
  <p><?=$this->Paginator->counter() ?></p>
</div>
</div>

```

Controlador

La convención de nombres de las clases de los controladores son en plural en formato CamelCase, y terminan en Controller.php, ejemplo: UsersController.php.

El siguiente fragmento de código corresponde al modelo UsersController.php y consiste en funciones para realizar las principales operaciones sobre los datos.

```
<?php
namespace App\Controller;

use App\Controller\AppController;

/**
 * Users Controller
 *
 * @property \App\Model\Table\UsersTable $Users
 */
class UsersController extends AppController
{
    /**
     * Index method
     *
     * @return \Cake\Network\Response|null
     */
    public function index()
    {
        $this->paginate = [
            'contain' => ['Groups']
        ];
        $users = $this->paginate($this->Users);

        $this->set(compact('users'));
        $this->set('_serialize', ['users']);
    }

    public function indexJson()
    {
        //$this->autoRender = false;
        $users=$this->Users->find('all',[
            'contain'=>['Fotousers'=> function ($q) {
                return $q
                    ->select(['id','filename','dir','user_id','estado'])
                    ->where(['Fotousers.estado'=>1]);
            }
        ]]);

        $this->set('users', $users);
        $this->set('_serialize', ['users']);
    }
}
```

```

        //echo json_encode($users);
    }

    /**
     * View method
     *
     * @param string|null $id User id.
     * @return \Cake\Network\Response|null
     * @throws \Cake\Datasource\Exception\RecordNotFoundException When record
not found.
     */
    public function view($id = null)
    {
        $user = $this->Users->get($id, [
            'contain' => ['Groups', 'Fotousers']
        ]);

        $this->set('user', $user);
        $this->set('_serialize', ['user']);
    }

    public function viewJson($id=null)
    {
        //$this->autoRender = false;
        $users=$this->Users->get($id,[
            'contain'=>['Fotousers'=> function ($q) {
                return $q
                    ->select(['id','filename','dir','user_id','estado'])
                    ->where(['Fotousers.estado'=>1]);
            }
        ]);

        $this->set('users', $users);
        $this->set('_serialize', ['users']);
        //echo json_encode($users);
    }

    /**
     * Add method
     *
     * @return \Cake\Network\Response|void Redirects on successful add,
renders view otherwise.
     */
    public function add()
    {
        $user = $this->Users->newEntity();
        if ($this->request->is('post')) {
            $user = $this->Users->patchEntity($user, $this->request->data);
            if ($this->Users->save($user)) {
                $this->Flash->success(__('The user has been saved.'));

                return $this->redirect(['action' => 'index']);
            } else {

```

```

        $this->Flash->error(__('The user could not be saved. Please,
try again.'));
    }
}
$groups = $this->Users->Groups->find('list', ['limit' => 200]);
$this->set(compact('user', 'groups'));
$this->set('_serialize', ['user']);
}

public function addJson()
{
    $result=null;
    $users=$this->Users->newEntity();
    if ($this->request->is('PUT')) {
        $this->autoRender = FALSE;
        $users=$this->Users->patchEntity($users, $this->request->data);
        if (!$users->errors()) {
            if ($this->Users->save($users)) {
                $result=$this->Noti->success('Usuario
'.'.$users['username'].' , guardado correctamente',true,$users);
            } else {
                $result=$this->Noti->error('No se pudo guardar el
Usuario '.'.$users['username'].'',$users);
            }
        }else {
            $errores = $users->errors();
            $result=$this->Noti->warning($errores,$users);
        }
        echo json_encode($result);
    }
}

/**
 * Edit method
 *
 * @param string|null $id User id.
 * @return \Cake\Network\Response|void Redirects on successful edit,
renders view otherwise.
 * @throws \Cake\Network\Exception\NotFoundException When record not
found.
 */
public function edit($id = null)
{
    $user = $this->Users->get($id, [
        'contain' => []
    ]);
    if ($this->request->is(['patch', 'post', 'put'])) {
        $user = $this->Users->patchEntity($user, $this->request->data);
        if ($this->Users->save($user)) {
            $this->Flash->success(__('The user has been saved.'));

            return $this->redirect(['action' => 'index']);
        } else {
            $this->Flash->error(__('The user could not be saved. Please,
try again.'));
        }
    }
}

```

```

    }
    $groups = $this->Users->Groups->find('list', ['limit' => 200]);
    $this->set(compact('user', 'groups'));
    $this->set('_serialize', ['user']);
}

public function editJson($id=null)
{
    $result=null;
    $this->autoRender = FALSE;
    $users=$this->Users->get($id, [
        'contain' => []
    ]);
    if ($this->request->is('patch')) {
        $new=[
            'username'=>$this->request->data['username'],
            'password'=>$this->request->data['password'],
            'group_id'=>$this->request->data['group_id'],
            'estado'=>$this->request->data['estado'],
            'identificador'=>$this->request->data['identificador']
        ];
        $users=$this->Users->patchEntity($users, $new);
        if (!$users->errors()) {
            if ($this->Users->save($users)) {
                $result=$this->Noti->success('Usuario
'.'.users['username'].', guardado correctamente',true,$users);
            } else {
                $result=$this->Noti->error('No se pudo guardar el
Usuario '.'.users['username'], $users);
            }
        } else {
            $errores = $users->errors();
            $result=$this->Noti->warning($errores,$users);
        }
        echo json_encode($result);
    }
}

/**
 * Delete method
 *
 * @param string|null $id User id.
 * @return \Cake\Network\Response|null Redirects to index.
 * @throws \Cake\Datasource\Exception\RecordNotFoundException When record
not found.
 */
public function delete($id = null)
{
    $this->request->allowMethod(['post', 'delete']);
    $user = $this->Users->get($id);
    if ($this->Users->delete($user)) {
        $this->Flash->success(__('The user has been deleted.'));
    } else {
        $this->Flash->error(__('The user could not be deleted. Please,
try again.'));
    }
}

```

```

    }

    return $this->redirect(['action' => 'index']);
}

public function deleteJson($id=null)
{
    $result=null;
    $this->autoRender = FALSE;
    $this->request->allowMethod(['post', 'delete']);
    $users=$this->Users->get($id);
    try {
        if ($this->Users->delete($users)) {
            $result=$this->Noti->success('Usuario
'.$users['username'].' , borrado correctamente',true,$users);
        } else {
            $result=$this->Noti->error('No se pudo borrar el Usuario
'.$users['username'],$users);
        }
        echo json_encode($result);
    }catch (\PDOException $e) {
        $result=$this->ErrorPDO->codigoError($e->errorInfo[0]);
        echo json_encode($result);
    }
}

public function login() {
    if ($this->request->is('post')) {
        $user = $this->Auth->identify();
        if ($user) {
            $this->Auth->setUser($user);
            return $this->redirect($this->Auth->redirectUrl());
        }
        $this->Flash->error(__('Your username or password was
incorrect.'));
    }
}

public function logout() {
    $this->Flash->success(__('Good-Bye'));
    $this->redirect($this->Auth->logout());
}

public function initialize()
{
    parent::initialize();

    $this->Auth->allow();
}
}

```

4.4 Transición

Esta fase asegura la disponibilidad de la aplicación para los usuarios finales, efectuando pruebas con el fin de verificar que el producto cumpla con los requerimientos establecidos y detectar oportunamente errores en el funcionamiento.

4.4.1 Pruebas de caja blanca

También llamadas pruebas estructurales ya que se realizan sobre funciones internas de la aplicación y se evalúa si el código fuente proporciona el resultado esperado.

La finalidad de las pruebas de caja blanca es detectar y corregir errores dentro del código fuente.

Prueba del camino básico

El método del camino básico permite establecer la complejidad lógica del diseño. En esta prueba examina la función asegurando que se ejecuten todas las sentencias del código al menos una vez con los diferentes caminos que se pueden generar gracias a las estructuras condicionales con sus respectivos estados.

La finalidad de esta prueba es garantizar lo siguiente:

- Se ejecuten los bucles repetitivos en sus límites.
- Todos los caminos se ejecuten al menos una vez independiente de cada módulo.
- Se ejecuten las decisiones en su parte verdadera y falsa.
- Se utilicen todas las estructuras de datos.

Se realiza la prueba con la función para agregar un nuevo usuario:

Camino	Función
0	public function add() {
1	\$a = array('estado' => FALSE, 'datos'=>null, 'titulo'=>null,'mensaje'=>null, 'tipomsg'=>'blackgloss');

2	<code>if (\$this->request->is('post')) {</code>
3	<code> \$this->autoRender = FALSE; \$this->User->set(\$this->request->data);</code>
4	<code> if(\$this->User->validates() {</code>
5	<code> \$this->User->create();</code>
6	<code> if (\$this->User->save(\$this->request->data) {</code>
7	<code> \$a['titulo']='Correcto.'; \$a['mensaje'][0]=' Cliente guardado correctamente'; \$a['estado']=true; \$a['tipormsg']='success'; }</code>
8	<code> else {</code>
9	<code> \$a['titulo']='Error.'; \$a['mensaje'][0]=' No se pudo guardar el Cliente'; \$a['tipormsg']='danger'; }</code>
10	<code> }</code>
11	<code> else {</code>
12	<code> \$errores = \$this->Cliente->validationErrors; \$acu=0; foreach (\$errores as \$indice => \$subarray) { \$a['mensaje'][\$acu++] = \$subarray[0]; } \$a['tipormsg']='warning'; \$a['titulo']='Aviso.'; }</code>
13	<code> return json_encode(\$a); }</code>
14	<code>}</code>

Tabla 4.40: Prueba de camino básico de función agregar usuario
Elaborado por: Rubén Jiménez – Investigador

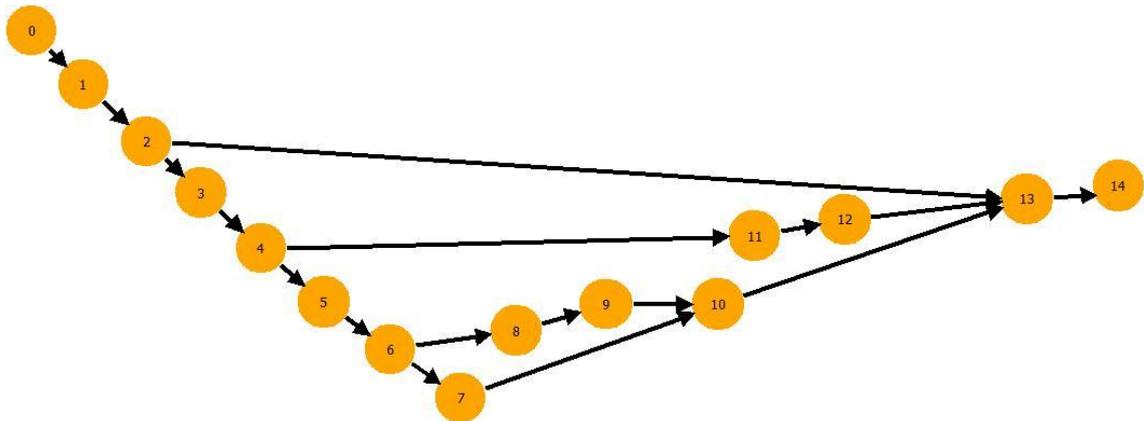


Figura 4.47: Grafo prueba del camino básico
Elaborado por: Rubén Jiménez – Investigador

Complejidad ciclomática

$$V(G) = A \text{ (arista)} - N \text{ (nodos)} + 2$$

$$V(G) = 16 - 14 + 2$$

$$V(G) = 4$$

Número	Nodos	Número de nodos
1	0-1-2-13-14	5
2	0-1-2-3-4-11-12-13-14	9
3	0-1-2-3-4-5-6-8-9-10-13-14	12
4	0-1-2-3-4-5-6-7-10-13-14	11

Tabla 4.41: Caminos básicos
Elaborado por: Rubén Jiménez – Investigador

Para la comprobación se tomará el camino número 4:

Camino	Función
0	Inicio de la función, no recibe parámetros.
1	Declaración de un array que almacenará los mensajes de validación que se mostraran al usuario.
2	Comprobación de tipo de petición para el ingreso (post o put).
3	Se desactiva el autorenderizado. Asignación del modelo en el que se guardarán los datos.
4	Comprobación si se han validado los datos de User.
5	Creación del objeto para el registro ingresado.
6	Comprobación si los datos han sido guardados correctamente.
7	Guardado de parámetros para el mensaje de error (Título, Mensaje y Tipo).
10	Fin de comprobación
13	Retorno del mensaje de error en una sola variable.

14	Fin del camino.
----	-----------------

Tabla 4.42: Comprobación de camino básico
Elaborado por: Rubén Jiménez – Investigador

4.4.2 Prueba de caja negra

Las pruebas de caja negra son pruebas funcionales realizadas sobre la interfaz que se enfocan en el cumplimiento de los requerimientos iniciales y funcionalidad de la aplicación. Estas pruebas son indiferentes de la estructura y comportamiento interno del programa ya que no considera el código fuente entre sus parámetros a evaluar.

Se analiza el sistema en un ambiente real revisando procedimientos de rutina para encontrar fallos, poniendo énfasis en los siguientes aspectos:

- Correcto funcionamiento del menú de navegación.
- Visualización de la información de manera correcta y entendible para el usuario.
- Correcta validación de ingreso de usuarios.
- Controles debidamente validados y habilitados según el proceso, para evitar el ingreso de información errónea.
- La información innecesaria permanece oculta para el usuario.
- Búsquedas deben devolver la información requerida.

Prueba de caja negra: tabla usuarios

1) Presentación entendible

Se verifica que la información sea presentada en pantalla de manera clara y entendible para el usuario.

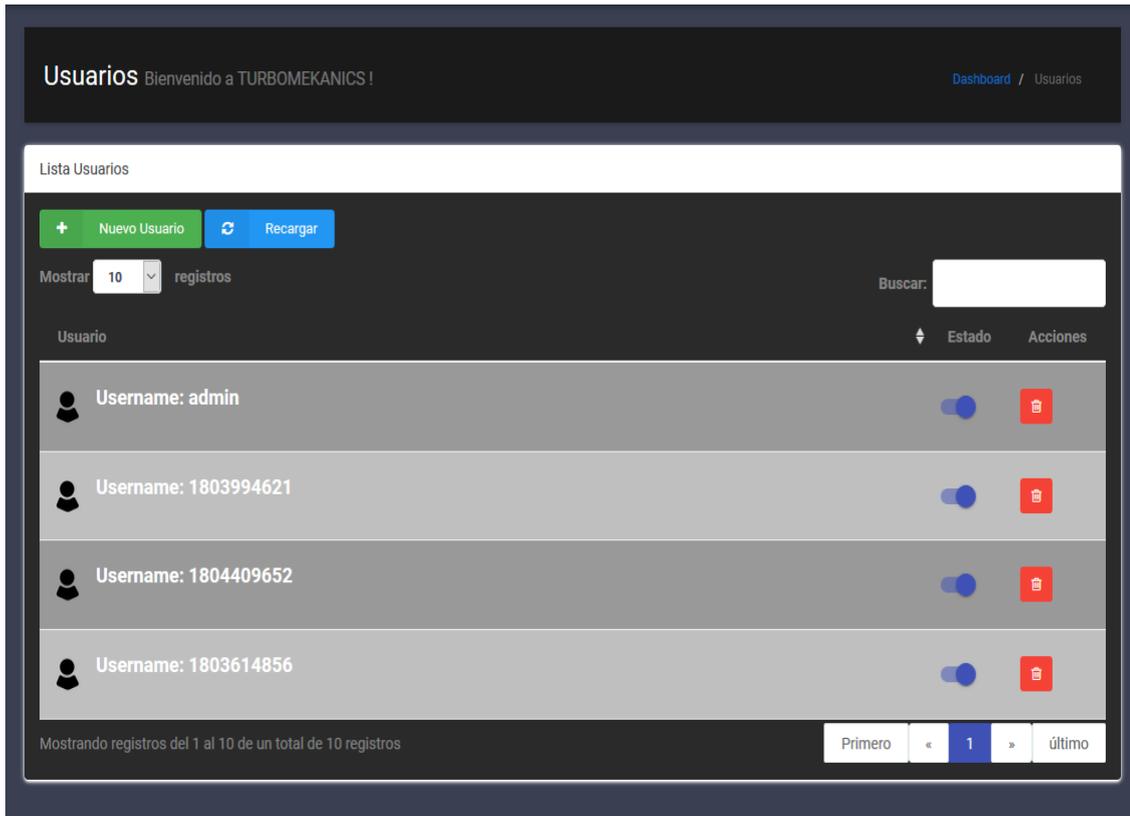


Figura 4.48: Prueba de caja negra: interfaz entendible
Elaborado por: Rubén Jiménez – Investigador

2) Campo requerido

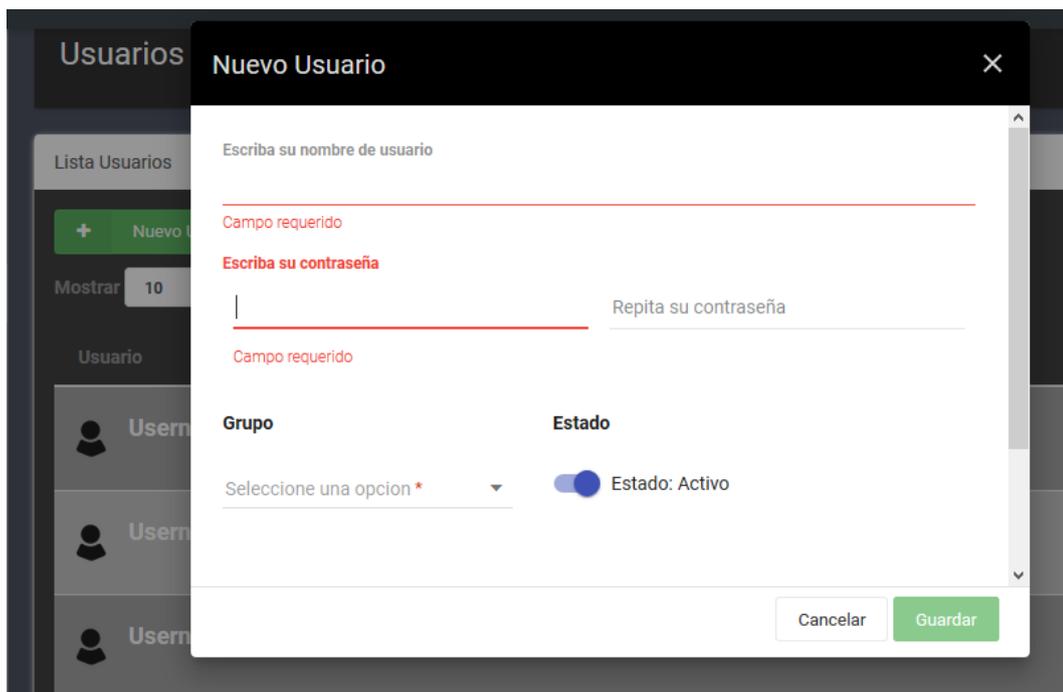
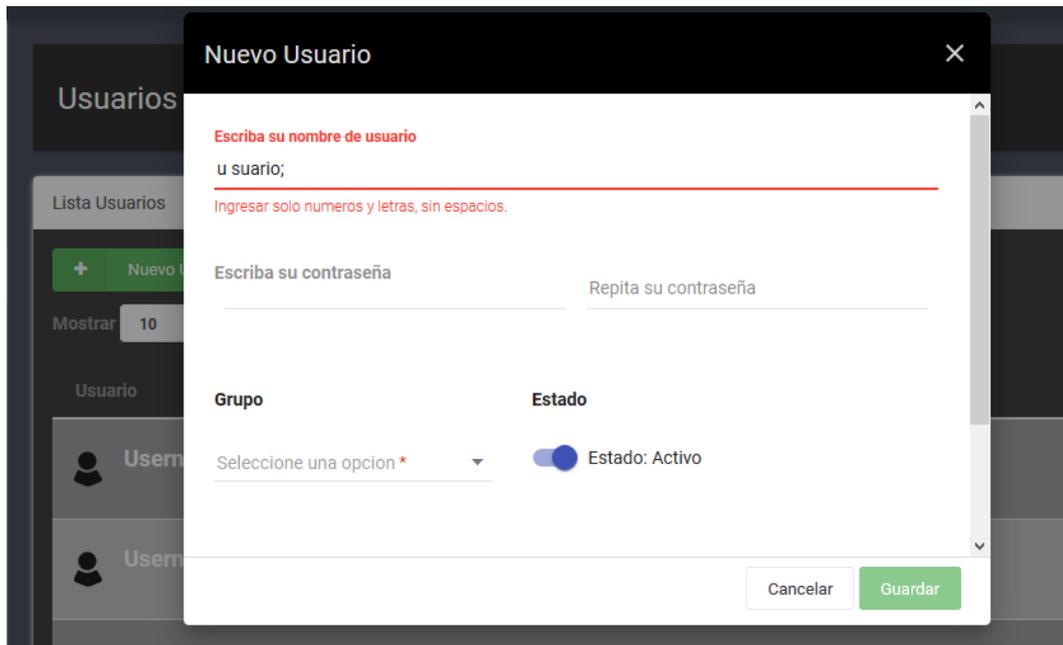


Figura 4.49: Prueba de caja negra: campo requerido
Elaborado por: Rubén Jiménez – Investigador

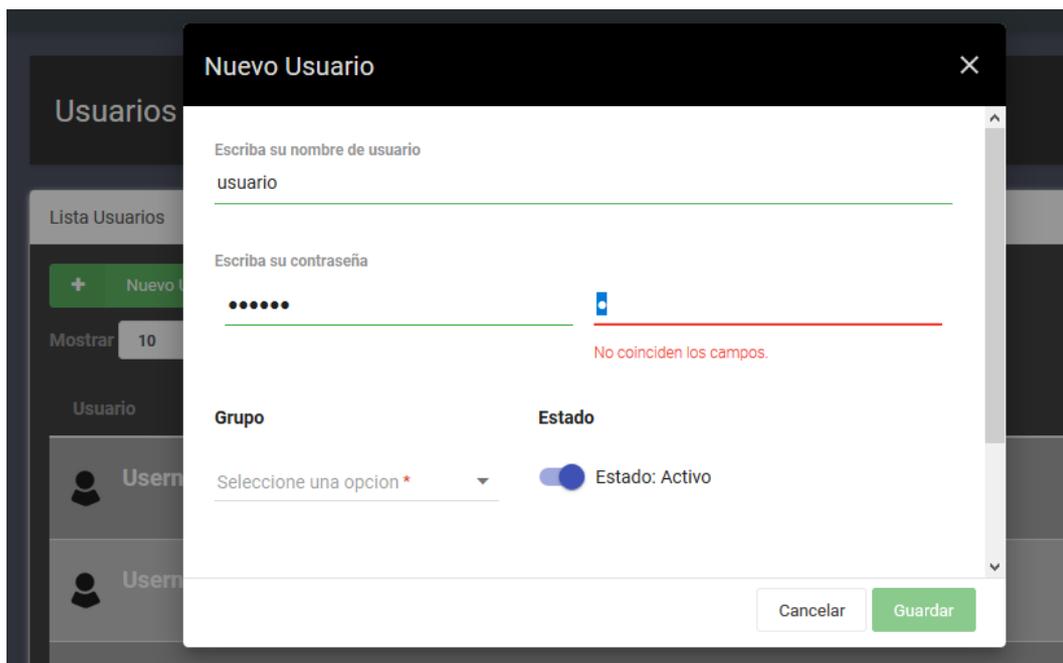
3) Tipo de datos incorrecto



The screenshot shows a 'Nuevo Usuario' (New User) form. The 'Escriba su nombre de usuario' (Enter your username) field contains the text 'u suario;'. A red error message below the field reads: 'Escriba su nombre de usuario u suario; Ingresar solo numeros y letras, sin espacios.' (Enter your username u suario; Only enter numbers and letters, no spaces). The 'Escriba su contraseña' (Enter your password) field is empty, and the 'Repita su contraseña' (Repeat your password) field is also empty. The 'Grupo' (Group) dropdown is set to 'Seleccione una opcion *'. The 'Estado' (Status) toggle is turned on, labeled 'Estado: Activo'. At the bottom right, there are 'Cancelar' (Cancel) and 'Guardar' (Save) buttons.

Figura 4.50: Prueba de caja negra: tipo de datos incorrecto
Elaborado por: Rubén Jiménez – Investigador

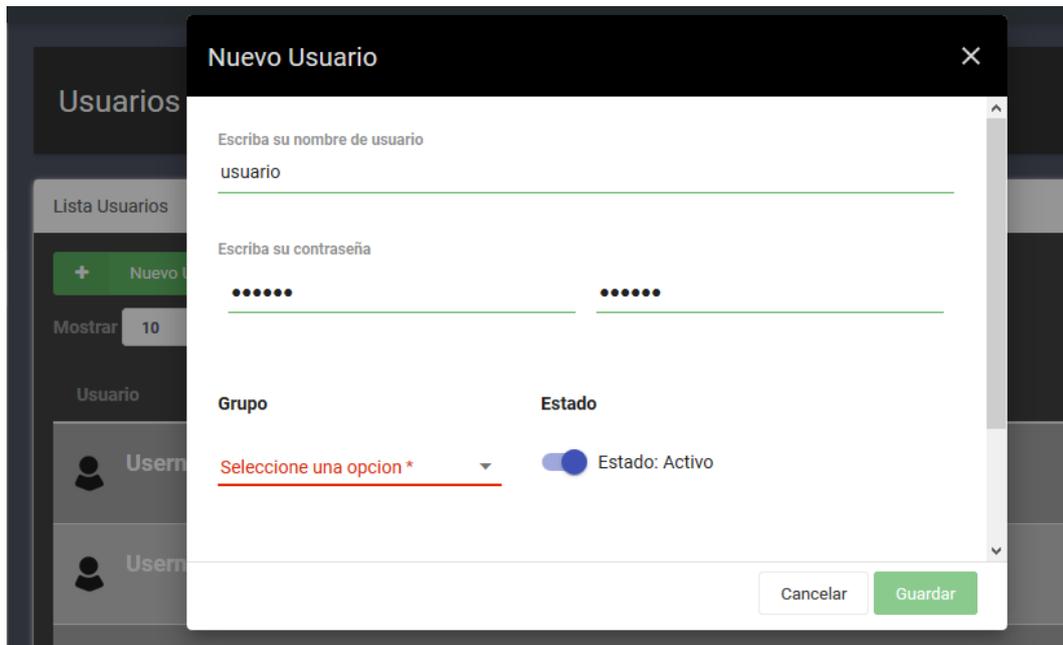
4) Inconsistencia de contraseñas



The screenshot shows the same 'Nuevo Usuario' form. The 'Escriba su nombre de usuario' field contains 'usuario'. The 'Escriba su contraseña' field contains six dots, and the 'Repita su contraseña' field contains a blue square. A red error message below the password fields reads: 'No coinciden los campos.' (Fields do not match). The 'Grupo' dropdown is 'Seleccione una opcion *' and the 'Estado' toggle is 'Estado: Activo'. 'Cancelar' and 'Guardar' buttons are at the bottom right.

Figura 4.51: Prueba de caja negra: inconsistencia de contraseñas
Elaborado por: Rubén Jiménez – Investigador

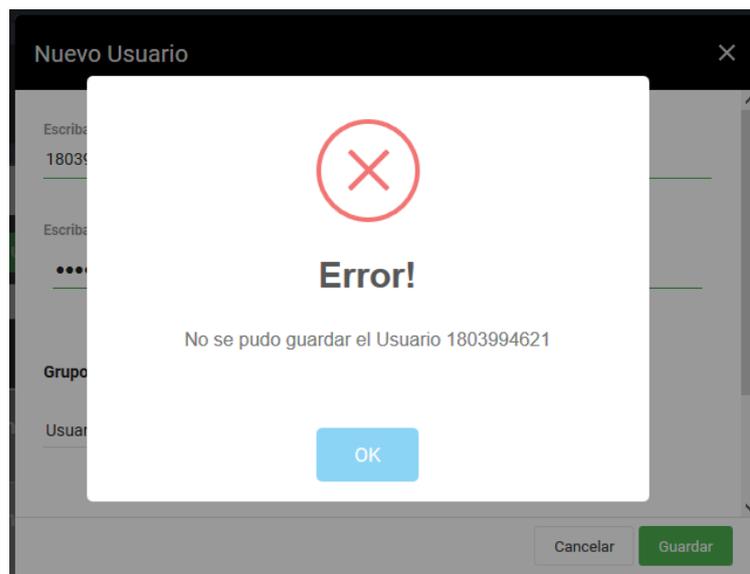
5) Opción no seleccionada



The image shows a web application interface for creating a new user. The main window is titled "Nuevo Usuario" and contains several input fields: "Escriba su nombre de usuario" (with "usuario" entered), "Escriba su contraseña" (with two masked password fields), "Grupo" (a dropdown menu), and "Estado" (a toggle switch labeled "Estado: Activo"). The "Grupo" dropdown menu is currently open, displaying the error message "Seleccione una opción *" in red text. At the bottom right of the form, there are "Cancelar" and "Guardar" buttons.

Figura 4.52: Prueba de caja negra: opción no seleccionada
Elaborado por: Rubén Jiménez – Investigador

6) Registro existente



The image shows the same "Nuevo Usuario" form as in Figure 4.52, but with an error dialog box overlaid in the center. The dialog box has a red "X" icon and the text "Error!" followed by "No se pudo guardar el Usuario 1803994621". There is an "OK" button at the bottom of the dialog box. The "Cancelar" and "Guardar" buttons from the form are visible at the bottom right of the screen.

Figura 4.53: Prueba de caja negra: registro existente
Elaborado por: Rubén Jiménez – Investigador

4.5 Implantación

Terminada la etapa de programación se procede a instalar la aplicación siguiendo el siguiente cronograma incluyendo la migración de la información desde los sistemas anteriores:

Actividad	Tiempo estimado
Instalación y configuración de servidor web	4 horas
Configuración de PHP y MySQL	4 horas
Migración y replicación de datos	1 día
Realización de pruebas de funcionamiento	1 día
Corrección y verificación de errores	2 días
Entrega de la aplicación	1 día
Capacitación al personal	2 días

Tabla 4.43: Plan de Implantación
Elaborado por: Rubén Jiménez – Investigador

4.5.1 Instalación y configuración del servidor web

Para la instalación y configuración del servidor web se siguen los mismos pasos que en la implementación, que consta en el Anexo 2.

4.5.2 Configuración de PHP y MySQL

Para la configuración de PHP primero se debe editar la configuración de apache para

```
# Especificar un archivo por defecto
DirectoryIndex index.php index.php3 index.html index.htm

# Asignarle el tipo MIME al documento PHP y su extensión
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
AddType application/x-httpd-php .php
AddType application/x-httpd-php3 .php3
```

que acepte el tipo de archivos, se edita el archivo `conf/httpd.conf`.

Para que el servidor se pueda conectar y trabajar adecuadamente con MySQL se edita el archivo de configuración de php: *php.ini*

4.5.3 Migración y replicación de datos

Actualmente la información de la empresa es manejada haciendo uso del sistema de

```
; Habilitar Open Tags (no recomendado)
short_open_tag = Off

; Tipo de Mensajes de error.
error_reporting = E_ALL
display_errors = On

; Path en relativo, en donde están las extensiones para PHP
extension_dir = "c:/wamp64/bin/php/php7.0.10/ext/"

; Carpeta temporal donde se copian los archivos subidos por medio de formularios
upload_tmp_dir = "c:/wamp64/tmp"

; Cargar la extensión de MySQL
extension=php_mysqli.dll

; Directorio donde se guardan los valores de sesiones
session.save_path = "c:/wamp64/tmp"
```

gestión de base de datos Sybase Adaptive Server Anywhere versión 9.

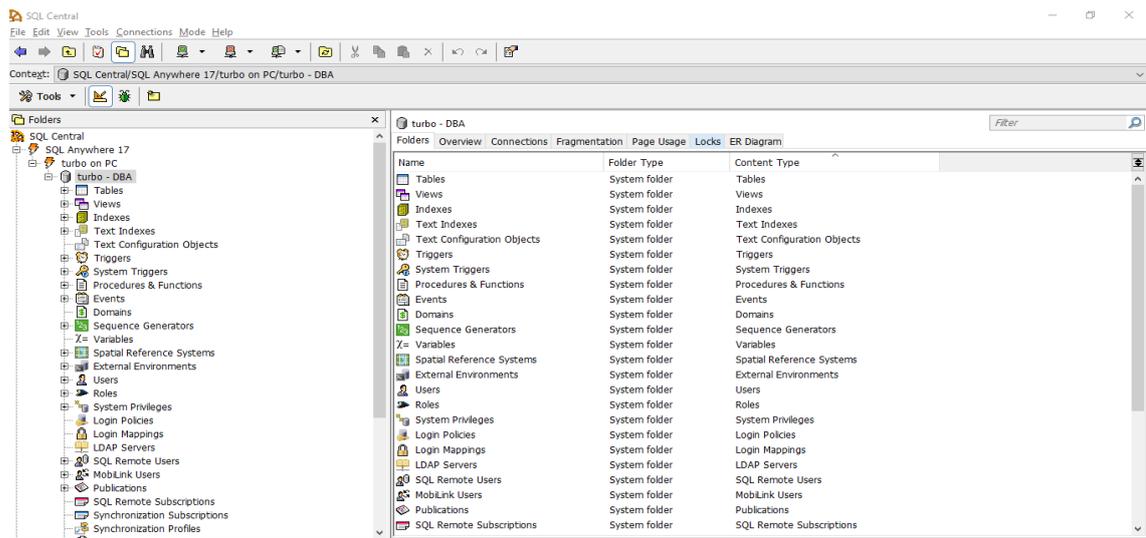


Figura 4.54: SGBD usado actualmente
Elaborado por: Rubén Jiménez – Investigador

Para implementar la nueva aplicación es necesario para la empresa migrar la información y que esta se replique haciendo que la aplicación se mantenga conectada a los sistemas de software que son usados actualmente.

Migración

“La migración de datos en el contexto de los sistemas de ficheros consiste en transformar un sistema de ficheros en otro sin la pérdida de datos, de modo que se pueda continuar trabajando con los datos en el nuevo sistema de datos” [36]

La migración de datos en un sistema es necesaria por estas razones:

- Mejora en el rendimiento y seguridad al utilizar nuevas versiones de sistemas de almacenamiento.
- Nuevos requisitos del usuario requieren el uso de soportes de almacenamiento nuevos que cumplan con las características acordes a las necesidades requeridas.
- Crecimiento en el volumen de información.
- Fusión de varios sistemas en uno solo para un determinado uso.

La migración se realiza siguiendo el siguiente esquema:

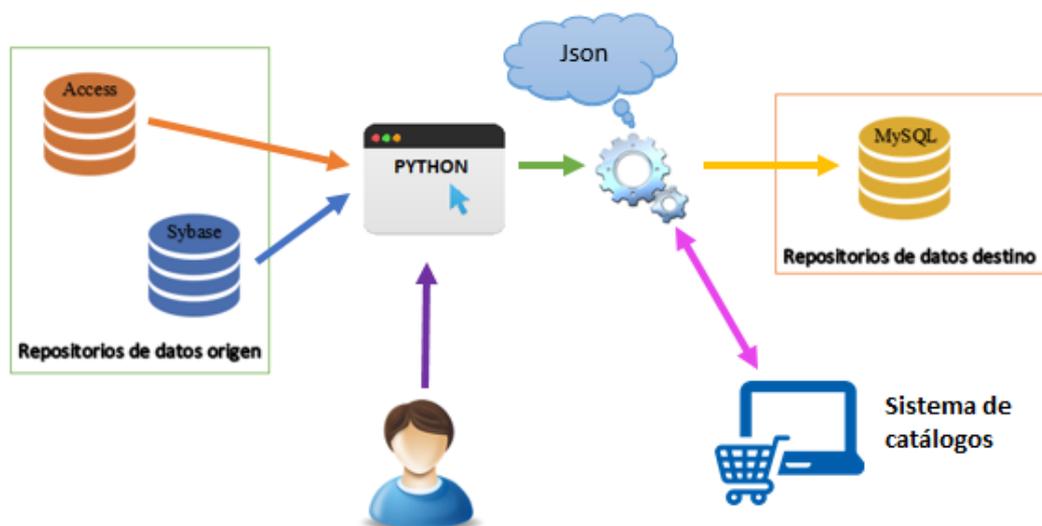


Figura 4.55: Esquema de migración
Elaborado por: Rubén Jiménez – Investigador

Para la migración se ha desarrollado un programa en Python que permite pasar los datos de Sybase Adaptive Server Anywhere y Access a MySQL. El código se encuentra en el Anexo 5.

El programa cuenta con seguridades, por ello pide la ubicación de los servicios web y la cuenta de usuario y contraseña del administrador.

“Los servicios web son un tipo de middleware mediante el que pueden comunicarse aplicaciones remotas.” [37]

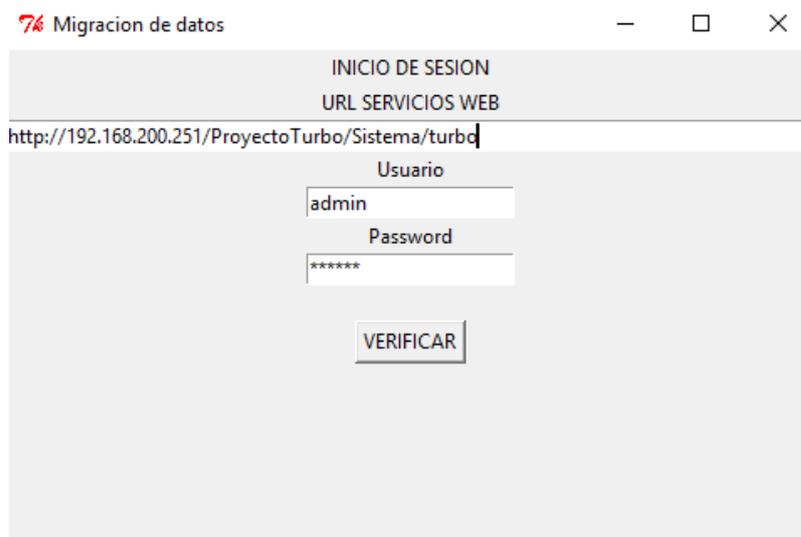


Figura 4.56: Interfaz inicio de sesión programa migración de datos
Elaborado por: Rubén Jiménez – Investigador

En seguida aparecerá una pantalla que por seguridad pedirá la ruta de la base de datos Sybase Adaptive Server Anywhere con el nombre de la base de datos, usuario y contraseña; y la ruta de la base de datos de Access con su respectiva contraseña.

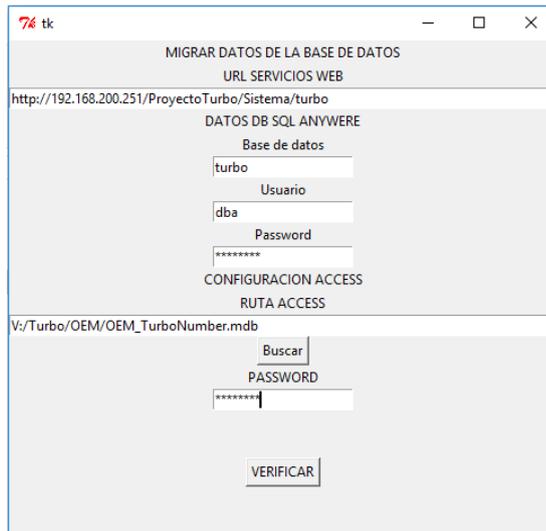


Figura 4.57: Interfaz conexión a bases de datos
Elaborado por: Rubén Jiménez – Investigador

Esta interfaz permite migrar los datos de las tablas clientes y artículos desde Sybase Adaptive Server Anywhere y de la tabla imagen desde Access haciendo uso de servicios web (web services).

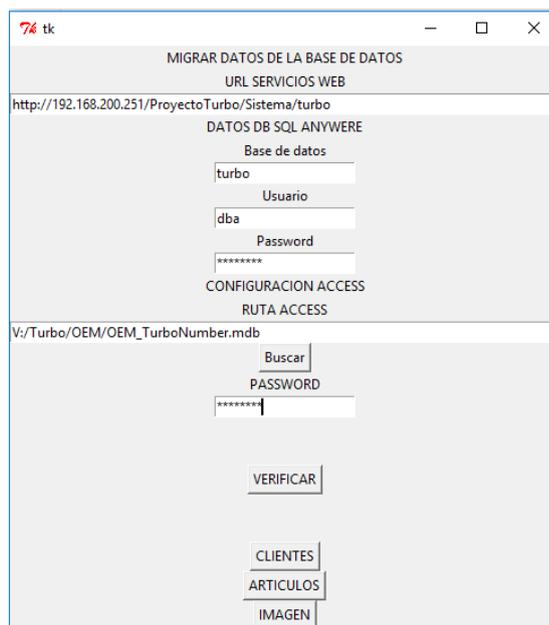


Figura 4.58: Interfaz migración
Elaborado por: Rubén Jiménez – Investigador

Al ejecutar el programa de migración elaborado en Python, éste actúa como intermediario estableciendo la conexión a las bases de datos Microsoft Access y Sybase Adaptive Server Anywhere con el servicio web al que realiza el llamado.

Al ejecutar una de las opciones de migración en el programa en Python se llama al servicio web que retorna métodos en formato Json para la inserción, lectura, actualización y eliminación de los datos; consumiéndolos desde el servidor de datos origen hacia el servidor de datos destino y transportando dicha información mediante el uso del protocolo http.

Para ello se crea una tabla en el sistema gestor de base datos con el catálogo maestro de datos que contiene la ruta del servicio web.

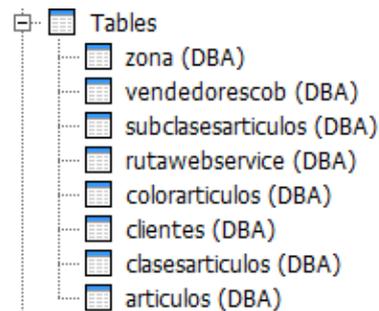


Figura 4.59: Tablas de catálogo de datos maestro
Elaborado por: Rubén Jiménez – Investigador

rutawebsevice (DBA)								
Columns	Constraints	Referencing Constraints	Indexes	Text Indexes	Triggers	Privileges	Dependent Views	Data
	id	path						
1	1	http://192.168.200.251/ProyectoTurbo/Sistema/turbo						

Figura 4.60: Tabla con la ruta del web service
Elaborado por: Rubén Jiménez – Investigador

Cabe destacar que el programa además permitirá agregar nuevos registros evitando datos repetidos.

```
C:\WINDOWS\system32\cmd.exe - python migracion.py
C:\Users\mondogo\Desktop\Ruben\Turbo\pruebas>python migracion.py
{"msg":{"unique":"RUC O CED ya registrado."},"tipomsg":"warning","titulo":"Advertencia!","datos":{"status":"","fax":"","listaprecio_id":1,"sexo":"","nombre":"Wilson Perez Nata","estado":"","direccion":"Pinillo ggg","estadocivil":"","decuento":1,"mail":"","contacto":"","telefono":"032851150"}}}
{"msg":{"unique":"RUC O CED ya registrado."},"tipomsg":"warning","titulo":"Advertencia!","datos":{"status":"","fax":"","listaprecio_id":2,"sexo":"","nombre":"Alex Lopez\r\n","estado":"","direccion":"Pinillo","estadocivil":"","decuento":5,"mail":"","contacto":"","telefono":"032854562"}}}
{"msg":{"unique":"RUC O CED ya registrado."},"tipomsg":"warning","titulo":"Advertencia!","datos":{"status":"","fax":"","listaprecio_id":4,"sexo":"","nombre":"ggg","estado":"","direccion":"Pin","estadocivil":"","decuento":1,"mail":"","contacto":"","telefono":"454211"}}}
{"msg":{"unique":"RUC O CED ya registrado."},"tipomsg":"warning","titulo":"Advertencia!","datos":{"status":"","fax":"","listaprecio_id":3,"sexo":"","nombre":"Pato too","estado":"","direccion":"fhdgh","estadocivil":"","decuento":2,"mail":"","contacto":"","telefono":"45121212"}}}
{"msg":{"unique":"RUC O CED ya registrado."},"tipomsg":"warning","titulo":"Advertencia!","datos":{"status":"","fax":"","listaprecio_id":4,"sexo":"","nombre":"Lunnn","estado":"","direccion":"fdhgjk","estadocivil":"","decuento":2,"mail":"","contacto":"","telefono":"2323"}}}
{"msg":{"unique":"RUC O CED ya registrado."},"tipomsg":"warning","titulo":"Advertencia!","datos":{"status":"","fax":"","listaprecio_id":2,"sexo":"","nombre":"Nose","estado":"","direccion":"dsgfhj","estadocivil":"","decuento":4,"mail":"","contacto":"","telefono":"4521212"}}}
{"msg":{"unique":"RUC O CED ya registrado."},"tipomsg":"warning","titulo":"Advertencia!","datos":{"status":"","fax":"","listaprecio_id":2,"sexo":"","nombre":"fdhgj","estado":"","direccion":"gdsfh","estadocivil":"","decuento":1,"mail":"","contacto":"","telefono":"12132132"}}}
{"msg":{"unique":"RUC O CED ya registrado."},"tipomsg":"warning","titulo":"Advertencia!","datos":{"status":"","fax":"","listaprecio_id":2,"sexo":"","nombre":"Ruben Jime","estado":"","direccion":"Amabato","estadocivil":"","decuento":1,"mail":"","contacto":"","telefono":"032457878"}}}
```

Figura 4.61: Programa para migración de datos funcionando
Elaborado por: Rubén Jiménez – Investigador

Replicación

“La replicación es un conjunto de tecnologías destinadas a la copia y distribución de datos y objetos de base de datos desde una base de datos a otra, para luego sincronizar ambas bases de datos y mantener su coherencia.” [38]

Mediante la replicación es posible distribuir los datos en diferentes ubicaciones y/o entre usuarios remotos a través de conexiones de área local o extensa.

El modelo que utiliza la replicación es de publicación y suscripción, en el que el servidor principal (publicador) distribuye sus datos a los servidores secundarios (suscriptores).

Los tipos de replicación que existen son los siguientes:

- **Replicación transaccional:** se realiza cuando se producen cambios en las tablas (en tiempo real) garantizando la coherencia transaccional (usado en entornos de servidor a servidor).

- **Replicación instantánea (snapshot):** transfiere toda la tabla con los datos exactos (nombres, campos y contenido) para su acceso de solo lectura, para su modificación se debe acceder al servidor de origen.
- **Replicación mezcla:** en ambos lados, tanto en el origen como en el destino se actualiza la información.

La replicación se realiza siguiendo el siguiente esquema:

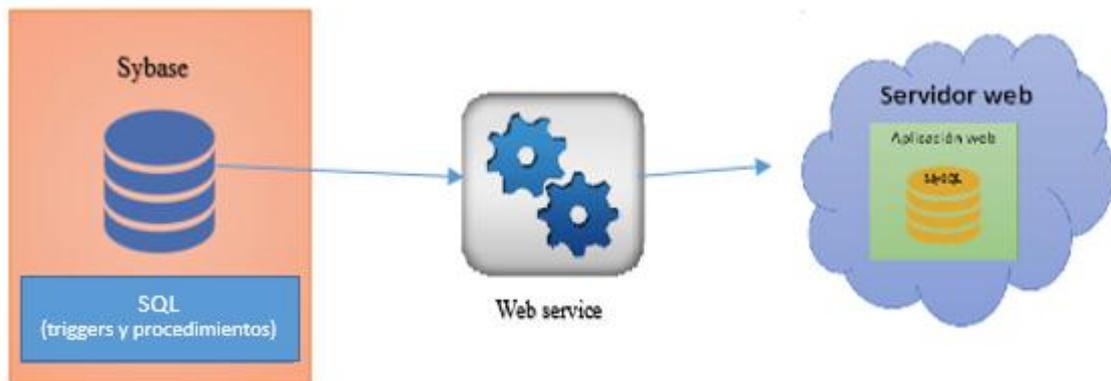


Figura 4.62: Esquema de replicación
Elaborado por: Rubén Jiménez – Investigador

Para la aplicación se realizó la replicación instantánea mediante la implantación de triggers y procedimientos que se ejecutan cuando en la base de datos maestra sucede algún evento como el ingreso, eliminación o actualización de registros.

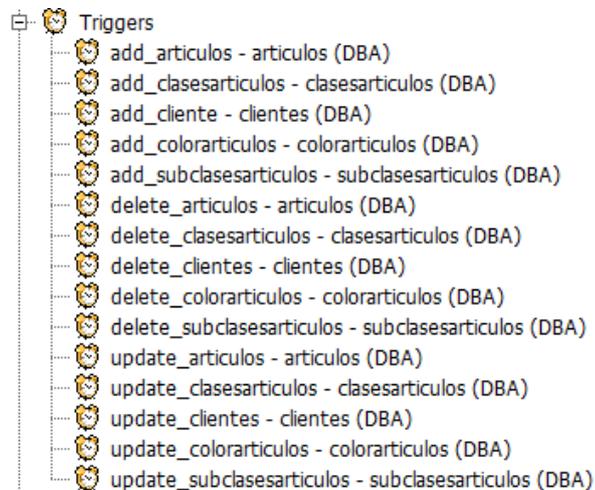


Figura 4.63: Triggers para la replicación de datos
Elaborado por: Rubén Jiménez – Investigador

El código de los triggers se encuentra en el Anexo 6.

4.5.4 Pruebas de funcionamiento

Las pruebas de funcionamiento finales se han elaborado un cuadro para evaluar que se cumplan todas las condiciones requeridas por el usuario en la ejecución de la aplicación:

Prueba	Cumplimiento	
	SI	NO
Correcto inicio de sesión	X	
Visualización adecuada del menú de acuerdo a los privilegios	X	
Visualización de catálogo según el usuario que ha iniciado sesión	X	
Correcta administración de usuarios (administrador)	X	

Tabla 4.44: Pruebas finales de funcionamiento
Elaborado por: Rubén Jiménez – Investigador

4.5.5 Capacitación al personal

Objetivo: brindar a los usuarios finales conocimiento necesario para el uso operativo del sistema.

Involucrados: administrador y ayudante de sistemas.

Descripción: la capacitación se realizará en base al siguiente cuadro con los temas a tratar y su respectiva duración para su cumplimiento.	
Tema	Duración
Inicio de sesión	20 minutos
Administración de cuentas de usuario y grupos	2 horas
Explicación general de la interfaz	3 hora
Migración de datos	3 horas

Tabla 4.45: Plan de capacitación
Elaborado por: Rubén Jiménez – Investigador

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- El levantamiento de requerimientos fue una etapa de gran importancia en el proyecto ya que ayudó a entender las necesidades de la empresa y automatizarlas.
- El uso de Angular, CakePhp y Bootstrap como frameworks de desarrollo permitió la creación de un sistema robusto y dinámico.
- Con la utilización de un programa de migración se logró centralizar la información guardada en las base de datos Sybase Adaptive Server Anywhere y Microsoft Access en una sola base de datos MySQL reduciendo la redundancia y facilitando el control de los datos para que sean fácilmente consultables al encontrarse estos almacenados en un solo lugar.
- La arquitectura Modelo – Vista – Controlador (MVC) facilitó el desarrollo ya que mantiene el código fuente ordenado y permite su reutilización, facilitando la administración y entendimiento
- La implementación de un nuevo catálogo web provee acceso rápido y eficiente a los clientes con los precios que se adaptan a sus necesidades mejorando la atención que brinda la empresa.
- La aplicación se desarrolló con la ayuda de la metodología RUP que ordena los pasos a seguir para su elaboración, permite cumplir con los requerimientos y generar sistemas de calidad.

5.2 Recomendaciones

- Conocer los requerimientos reales tanto de los usuarios como del personal de la empresa para la realización de mejoras a futuro en la aplicación y brindar un mejor servicio a los clientes.
- Realizar mantenimiento frecuentemente y mantener actualizadas las herramientas tanto de desarrollo como de almacenamiento después de ser implantada la aplicación.
- Revisar que los triggers de replicación se encuentren funcionando óptimamente y de igual manera realizar backups de la base de datos con frecuencia para salvaguardar los datos en caso de algún desastre informático o problemas con el servidor.
- Se recomienda que la empresa implemente y mantenga políticas de desarrollo adecuadas en base a las últimas tecnologías existentes en el mercado.
- Se recomienda socializar el manual de funcionamiento con los usuarios involucrados en el uso del sistema para que de esta manera la aplicación sea manejada de mejor manera.
- Se recomienda ampliar el conocimiento acerca de los Servicios Web y de nuevos modelos de seguridad que pueden implementarse a futuro para cubrir vulnerabilidades.

Bibliografía

- [1] Instituto de Promoción de Exportaciones e Inversiones, «Análisis Sectorial Automotriz,» Pro Ecuador, Quito, 2017.
- [2] Garner Espinosa C.A, "Garner Espinosa C.A Inicio," [Online]. Available: <http://www.garner.com.ec/es/inicio>. [Accessed 27 Abril 2016].
- [3] TurboMaster, "TurboMaster inicio," UIO Media Innovation, [Online]. Available: <http://www.turbomaster.com.ec/>. [Accessed 27 Abril 2016].
- [4] TurboTech, "TurboTech inicio," [Online]. Available: <http://www.turbotech.ec/>. [Accessed 27 Abril 2016].
- [5] N. A. Moyolema Morales, "Diseño e implantación del sitio web para el control de pedidos para la empresa Comercial Yolanda Salazar Cia. Ltda," Ambato, 2006.
- [6] I. V. Villacís Palacios, "Diseño e implementación de una aplicación Web, para realizar pedidos on - line en la empresa Disama Cía. Ltda.," Ambato, 2007.
- [7] P. A. Aguayo Eldredge and D. P. Salcedo Mena, "Desarrollo de un sistema web ventas por catálogo para Agroimzoo Cía. Ltda.," Quito, 2014.
- [8] Editorial Vértice, Organización del proceso de venta, Editorial Vértice, 2011.
- [9] J. Heizer and B. Render, Principios de administración de operaciones, Pearson Educación, 2004 .
- [10] J. Niño, Aplicaciones web, Editex, 2010.
- [11] Template monster blog, "Template monster blog," [Online]. Available: <http://www.templatemonsterblog.es/2013/07/17/como-se-puede-crear-su-primera-aplicacion-web/>. [Accessed 30 Abril 2016].
- [12] J. Sánchez Asenjo, "Jorge Sanchez. net," 2012. [Online]. Available: <http://www.jorgesanchez.net/web/iaw/iaw1.pdf>. [Accessed 30 Abril 2016].
- [13] E. Ballester Gómez, P. Barco Martínez, P. Moreda Pozo, A. Suárez Cueto, A. Montoyo Guijarro and E. Saquete Boro, "Bases de Datos 1," [Online]. Available: <http://rua.ua.es/dspace/bitstream/10045/2990/1/ApuntesBD1.pdf>. [Accessed 30 Abril 2016].
- [14] I. Ramos Salavert y M. D. Lozano Pérez, Ingeniería del software y bases de datos: tendencias actuales, Ediciones de la Universidad de Castilla - La Mancha, 2000.
- [15] V. J. Eslava Muñoz, El nuevo PHP. Conceptos avanzados., Bubok Publishing S.L, 2013.

- [16] P. Freddi, Windows Server 2008: Los servicios de red TCP/IP, Ediciones ENI, 2010.
- [17] Inteco, Ingeniería de Software: Metodologías y Ciclos de vida, 2009.
- [18] E. Leyva, J. I. Prieto, M. d. I. Á. Sampalo y M. L. Garzón, Sistema y aplicaciones informáticas, Sevilla: MAD-Eduforma, 2006.
- [19] S. Briones, 2014 Mayo 8. [En línea]. Available: <https://es.slideshare.net/lowees/cuadro-comparativo-34456077>.
- [20] 1stWebDesigner Editorial Team, «1stWebDesigner.com,» 11 Enero 2016. [En línea]. Available: <https://1stwebdesigner.com/php-vs-ruby-vs-python/>.
- [21] The PHP Group, «<http://www.php.net/>,» [En línea]. Available: <http://www.php.net/manual/en/refs.database.vendors.php>.
- [22] M.-A. Lemburg, «<https://wiki.python.org/>,» 7 Junio 2014. [En línea]. Available: <https://wiki.python.org/moin/DatabaseInterfaces>.
- [23] Á. Cobo, P. Gómez, D. Pérez y R. Rocha, PHP y MySQL: Tecnología para el desarrollo de aplicaciones web., Madrid: Ediciones Díaz de Santos, 2005.
- [24] ADR Infor S.L, «Curso de Desarrollo de Páginas Web con PHP,» 2017. [En línea]. Available: <https://www.adrformacion.com/cursos/php2/leccion1/tutorial1.html>.
- [25] E. Brueggeman, «The Website of Elliott Brueggeman,» 29 Abril 2008. [En línea]. Available: <http://www.ebrueggeman.com/blog/php-mysql-performance>.
- [26] C. Thibaud, MySQL 5: instalación, implementación, administración, programación, Ediciones ENI, 2006.
- [27] M. Maier, D. Emery y R. Hilliard, Software Architecture: Introducing IEEE Standard 1471, Hopkinton: West Virginia University.
- [28] J. Barranco de Areba, Metodología del análisis estructurado de sistemas, Madrid: Universidad Pontificia Comillas, 2001.
- [29] S. R. Gómez Palomo y E. Moraleda Gil, Aproximación a la ingeniería del software, Madrid: Editorial Universitaria Ramon Areces, 2014.
- [30] M. C. Ortega Toledo, «Aplicación multimedia para visualización de datos estadísticos gestión de ventas,» Universidad del Azuay, Cuenca, 2013.
- [31] K. Rodriguez, «WebApache,» 5 Septiembre 2013. [En línea]. Available: <http://webapache.blogspot.com/2013/09/introduccion.html>.
- [32] Oracle Corporation , «MySQL Workbench,» 2017. [En línea]. Available: <https://www.mysql.fr/products/workbench/>.

- [33] A. Walsh, Getting Started with AngularJS, BookRix, 2016.
- [34] J. Acedo, «Apuntes de programación,» 4 Mayo 2015. [En línea]. Available: <http://programacion.jias.es/2015/05/web-%C2%BFque-es-el-framework-bootstrap-ventajas-desventajas/>.
- [35] J. Gube, «Design Instruct,» 5 Agosto 2014. [En línea]. Available: <http://designinstruct.com/web-design/flexible-grid-system/>.
- [36] C. Caballero González y J. A. Clavero García, Sistemas de almacenamiento UF1466, Madrid: Ediciones Paraninfo, S.A., 2016.
- [37] M. P. Usaola, «Introducción a los Servicios web,» Escuela Superior de Informática UCLM .
- [38] M. Pérez Marqués, SQL Server 2008 R2 : motor de base de datos y administración, Madrid: RC Libros, 2011.

GLOSARIO DE TÉRMINOS Y ACRÓNIMOS

Backup: copia de seguridad o de respaldo de los datos que se realiza con el fin de disponer de un medio para recuperarlos en caso de pérdida debido a catástrofes informáticas, ataque de virus informáticos, eliminado accidental u otras causas. Permite también el traslado de ubicación de los datos.

CSS: hojas de estilo en cascada es un lenguaje que define la presentación de un documento de una interfaz de usuario escrita en HTML.

Framework: estructura tecnológica y conceptual que sirve como base para el desarrollo de un programa informático, generalmente incluye herramientas, librerías de programación y un lenguaje interpretado.

Hosting: servicio que permite almacenar contenido accesible vía web (imágenes, videos, información).

HTML: lenguaje estándar de marcado de hipertexto para la elaboración de páginas web definiendo un código con una estructura básica de su contenido.

HTTP: protocolo de transferencia de hipertexto, es un protocolo de comunicación que permite la transmisión de información en la web definiendo la sintaxis y semántica que utilizan los elementos de la arquitectura web (clientes y servidores).

Javascript: es un lenguaje de programación interpretado orientado a objetos que permite mejoras en la interfaz de usuario y que las páginas web sean dinámicas.

Marketing: es un proceso que comprende la identificación de necesidades y formulación de objetivos orientados al consumidor

MVC: patrón de arquitectura de software que separa datos, lógica de negocio e interfaz de usuario en tres componentes.

Open source: software que es desarrollado y distribuido libremente y del cual se tiene acceso a su código fuente.

Replicación: servicio en que los datos almacenados se duplican en tiempo real.

Script: es un programa simple generalmente almacenado en un archivo de texto que realiza diversas tareas tales como interactuar con el usuario, con el sistema operativo y combinar componentes.

SQL: lenguaje de consulta estructurada, es un lenguaje que permite realizar diversos tipos de operaciones en un sistema de gestión de base de datos relacionales.

Triggers: son objetos que se almacenan en la base de datos y se asocian con las tablas al ejecutarse cuando sucede algún evento en ellas. Son empleados para implementar las reglas de negocio.

Turbocompresor: también llamado turbo, es un sistema de sobrealimentación utilizado en motores de combustión interna alternativos (motores diesel) que usa una turbina centrífuga para comprimir gases.

Web service: tecnología que usa estándares y protocolos para intercambiar datos entre aplicaciones.

ANEXOS Y APENDICES

ANEXO 1

ENTREVISTAS

Entrevista al administrador de sistemas de la empresa Turbomekanics S.A.

ENTREVISTADO:

ENTREVISTADOR:

FECHA:

Objetivo: Recolectar información sobre al manejo de catálogos de la empresa

Indicaciones: lea las siguientes preguntas y contéstelas con absoluta sinceridad de una manera clara en base a su experiencia.

1. ¿Cómo se registra actualmente la información de los productos que oferta la empresa?

.....
.....
.....

2. ¿A qué clientes van dirigidos los productos que ofrece la empresa?

.....
.....
.....

3. ¿Considera usted que facilitaría el manejo de la información la creación de una aplicación web?

.....
.....
.....

4. ¿Qué documentos de salida se requiere generar en la aplicación web?

.....
.....

.....
Entrevista al ayudante de sistemas de la empresa Turbomekanics S.A.

ENTREVISTADO:

ENTREVISTADOR:

FECHA:

Objetivo: Recolectar información sobre al manejo de catálogos de la empresa

Indicaciones: lea las siguientes preguntas y contéstelas con absoluta sinceridad de una manera clara en base a su experiencia.

1. ¿Considera usted que facilitaría el manejo de la información la creación de una aplicación web?

.....
.....
.....

2. ¿Qué contenidos deben figurar en la aplicación web?

.....
.....
.....

3. ¿Considera que un manual de usuario le ayudaría a mejorar el servicio?

.....
.....
.....

ANEXO 2

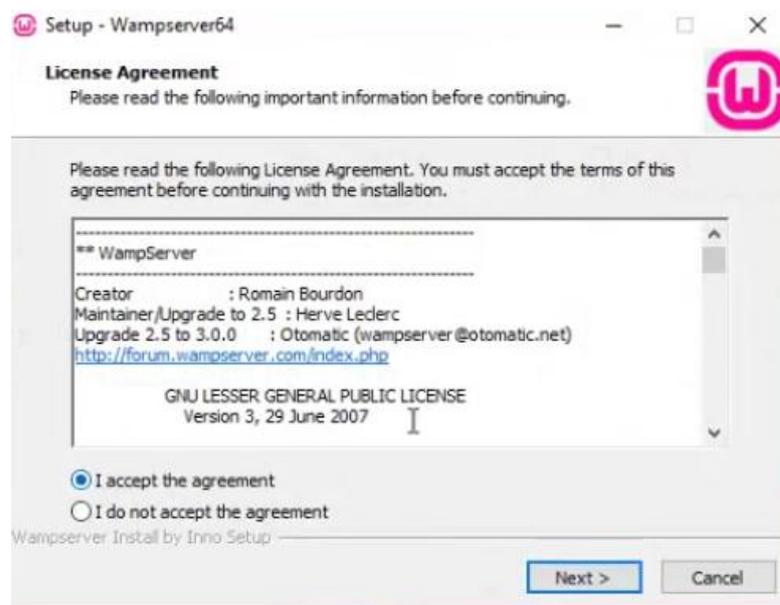
INSTALACIÓN DE WAMP SERVER

1) Obtener el archivo de instalación desde la web <http://www.wampserver.com/en/>

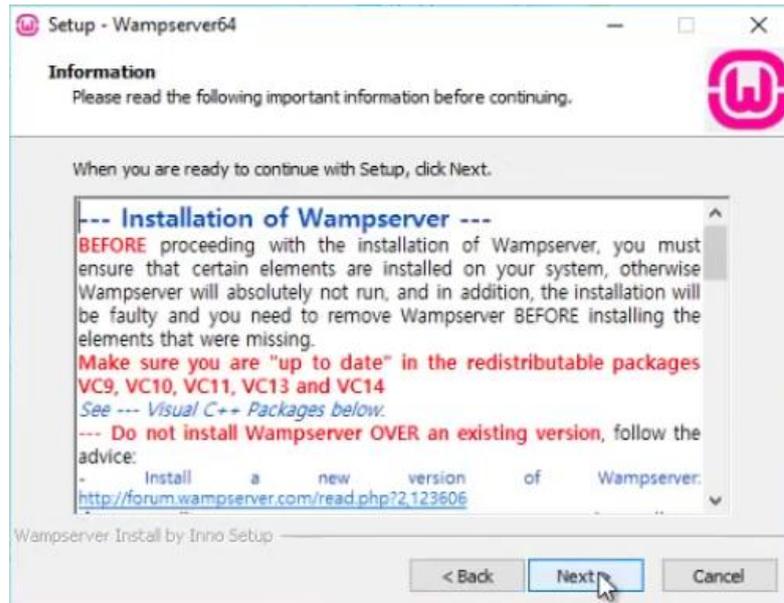
La última versión disponible al momento de realizar el proyecto de tesis es Wampserver 3.0.6 con las siguientes características:

- Apache 2.4.23
- PHP 5.6.25/7.0.10
- MySQL 5.7.14
- PhpMyAdmin 4.6.4
- Adminer 4.2.5
- PhpSysInfo 3.2.5

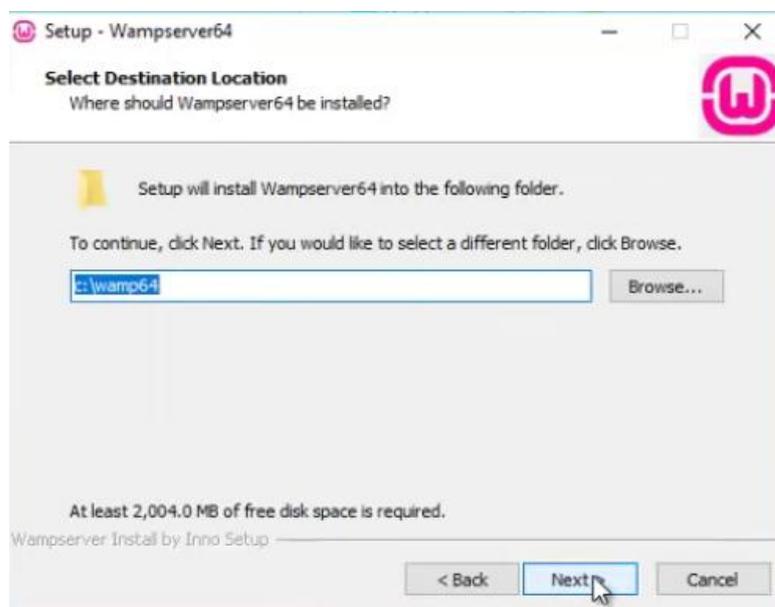
2) Iniciar el proceso de instalación.



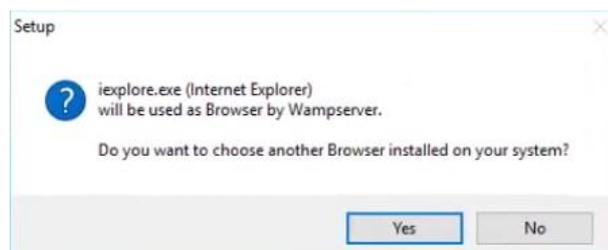
3) Aceptamos los términos y condiciones y seleccionamos siguiente (next).



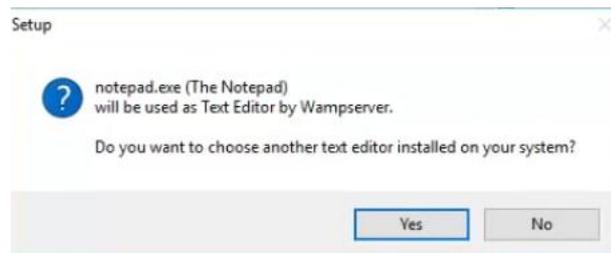
4) Seleccionamos el directorio en el que se quiere instalar el servidor



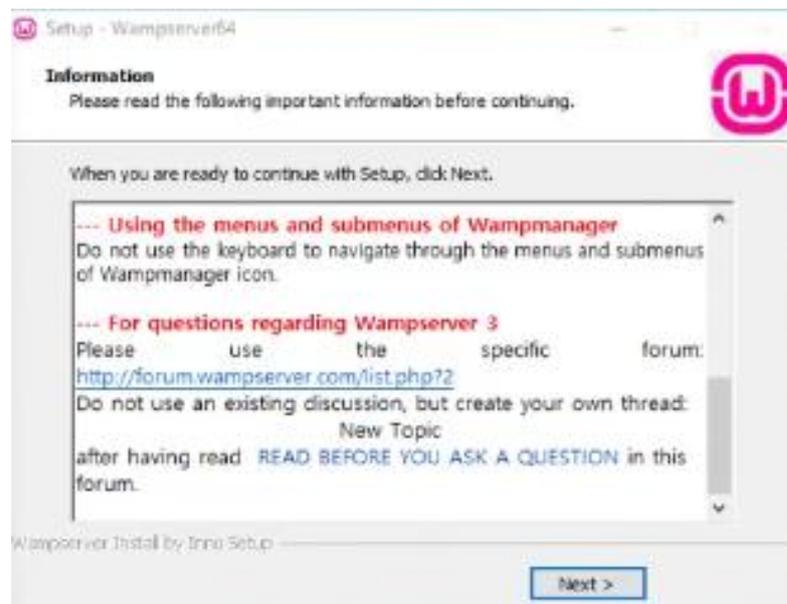
5) Después de copiar los archivos, nos pregunta el navegador por defecto que deseamos para el uso del wampserver



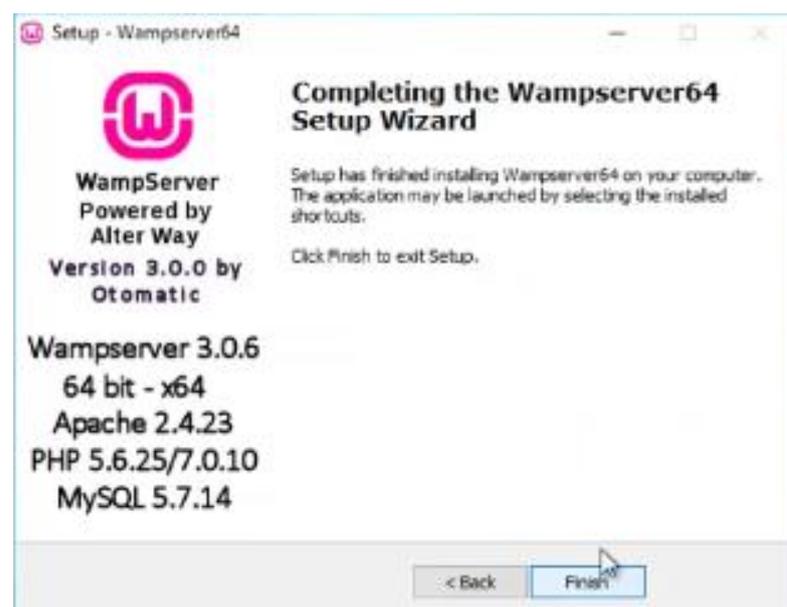
6) También nos pregunta cual editor de texto dejamos por defecto



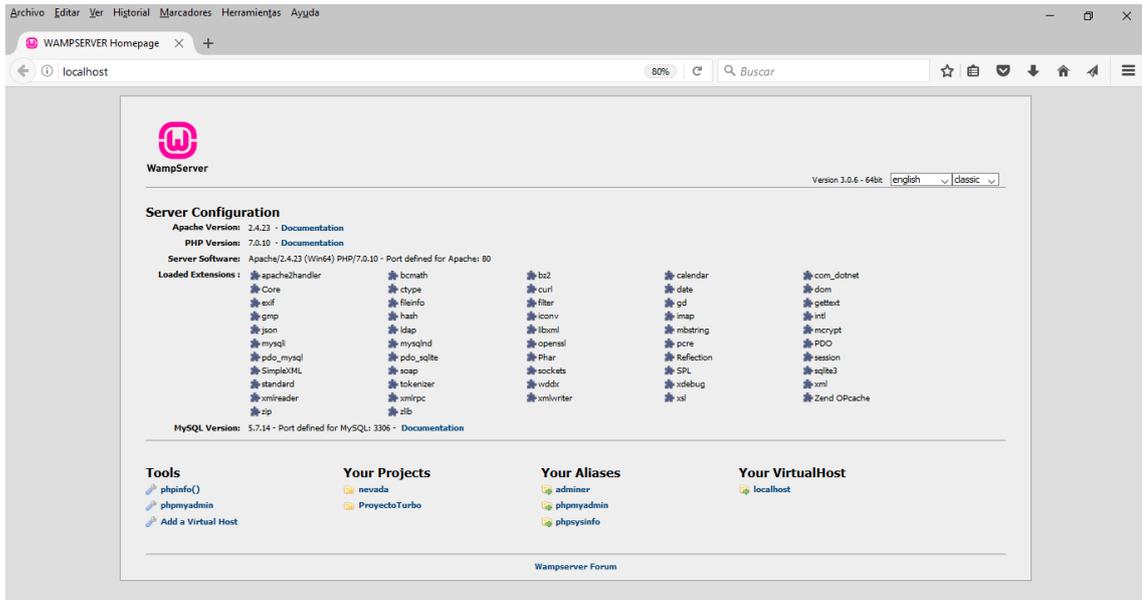
7) A continuación nos presenta un resumen de la instalación



8) Y finalizará la instalación



Se accede desde el navegador con la dirección de localhost y se muestra que WAMPserver se encuentra funcionando.



ANEXO 3

CÓDIGO SQL

```
/*=====
=*/

/* Table: articulos */

/*=====
=*/

create table articulos
(
    id          int not null auto_increment,
    codigo      varchar(20),
    nombre      varchar(60),
    costo       numeric,
    obsevacion  varchar(70),
    costofob    numeric,
    colorarticulo_id int,
    clasesarticulo_id int,
    subclasesarticulo_id int,
    primary key (id)
);

/*=====
=*/

/* Table: clasesarticulos */

/*=====
=*/

create table clasesarticulos
```

```
(
    id            int not null auto_increment,
    codigo        varchar(5),
    nombre        varchar(30),
    estado        varchar(1),
    primary key (id)
);
```

```
/*=====
=*/
```

```
/* Table: clientes */
```

```
/*=====
=*/
```

```
create table clientes
```

```
(
    id            int not null auto_increment,
    codigo        varchar(8),
    nombre        varchar(200),
    cederuc       varchar(13),
    direccion     varchar(70),
    telefono      varchar(20),
    fax           varchar(20),
    contacto      varchar(100),
    descuento     numeric,
    mail          varchar(200),
    sexo          varchar(1),
    estadocivil   varchar(1),
    status        varchar(10),
    estado        varchar(1),
```

```

        listaprecio_id    int,
        primary key (id)
    );

/*=====
=*/

/* Table: colorarticulos          */

/*=====
=*/

create table colorarticulos
(
    id            int not null auto_increment,
    codigo        varchar(10),
    nombre        varchar(10),
    primary key (id)
);

/*=====
=*/

/* Table: fotousers              */

/*=====
=*/

create table fotousers
(
    id            int not null auto_increment,
    filename      varchar(255),
    dir           varchar(255),
    mimetype      varchar(255),
    filesize      int,
    created       date,

```

```

        modified      date,
        file          blob,
        user_id       int,
        estado        int,
        primary key (id)
    );

/*=====
=*/

/* Table: groups                                     */

/*=====
=*/

create table groups
(
    id                int not null auto_increment,
    name              varchar(100) not null,
    created           date,
    modified          date,
    estado            int,
    primary key (id)
);

/*=====
=*/

/* Table: imagenes                                   */

/*=====
=*/

create table imagenes
(
    id                int not null auto_increment,

```

```
filename      varchar(255),
dir           varchar(255),
mimetype      varchar(255),
filesize      int,
created       date,
modified      date,
articulo_id   int4,
file          blob,
estado        int,
primary key (id)
```

```
);
```

```
/*=====
=*/
```

```
/* Table: listaprecios */
```

```
/*=====
=*/
```

```
create table listaprecios
```

```
(
```

```
id          int not null auto_increment,
```

```
codigo      varbinary(2),
```

```
estado      int,
```

```
primary key (id)
```

```
);
```

```
/*=====
=*/
```

```
/* Table: menugrups */
```

```
/*=====
=*/
```

```
create table menugrups
```

```
(  
  id          int not null auto_increment,  
  menu_id     int4,  
  group_id    int4,  
  user_id     int,  
  estado      int,  
  primary key (id)  
);
```

```
/*=====
```

```
/* Table: menus */
```

```
/*=====
```

```
create table menus
```

```
(  
  id          int not null auto_increment,  
  text        varchar(100),  
  heading     boolean,  
  icon        varchar(100),  
  sref        varchar(250),  
  alert       double,  
  label       varchar(250),  
  translate   varchar(250),  
  parent_id   int,  
  priority    int,  
  estado      int,  
  primary key (id)
```

```
);
```

```
/*=====
=*/
```

```
/* Table: precios */
```

```
/*=====
=*/
```

```
create table precios
```

```
(
```

```
  id          int not null auto_increment,
```

```
  listaprecio_id  int,
```

```
  valor         numeric,
```

```
  estado        int,
```

```
  articulo_id   int,
```

```
  primary key (id)
```

```
);
```

```
/*=====
=*/
```

```
/* Table: subclasesarticulos */
```

```
/*=====
=*/
```

```
create table subclasesarticulos
```

```
(
```

```
  id          int not null auto_increment,
```

```
  codigo      varchar(5),
```

```
  nombre      varchar(30),
```

```
  estado      varchar(1),
```

```
  primary key (id)
```

```
);
```

```
/*=====
=*/
/* Table: users */
/*=====
=*/
```

create table users

```
(
  id          int not null auto_increment,
  username    varchar(255) not null,
  password    char(250) not null,
  group_id    int not null,
  created     date,
  modified    date,
  estado      int,
  identificador varchar(10),
  primary key (id)
);
```

```
alter table fotousers add constraint fk_reference_3 foreign key (user_id)
  references users (id) on delete restrict on update restrict;
```

```
alter table users add constraint fk_reference_2 foreign key (group_id)
  references groups (id) on delete restrict on update restrict;
```

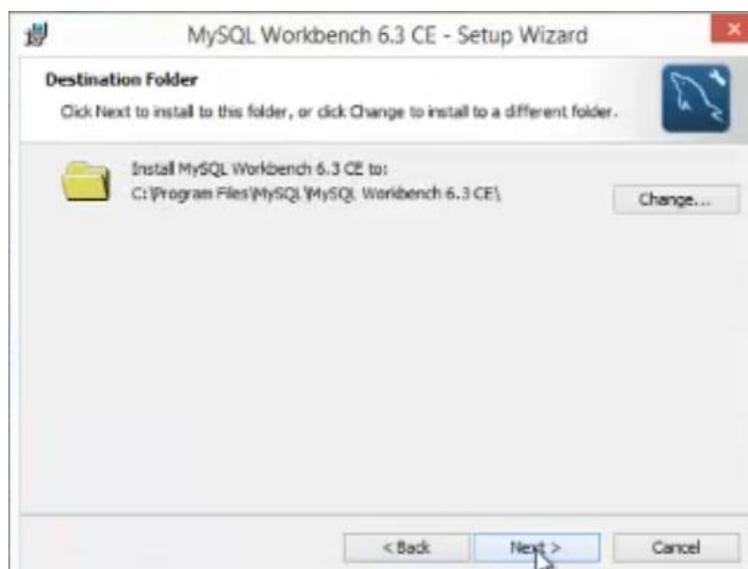
ANEXO 4

INSTALACIÓN Y CONFIGURACIÓN DE MYSQL WORKBENCH

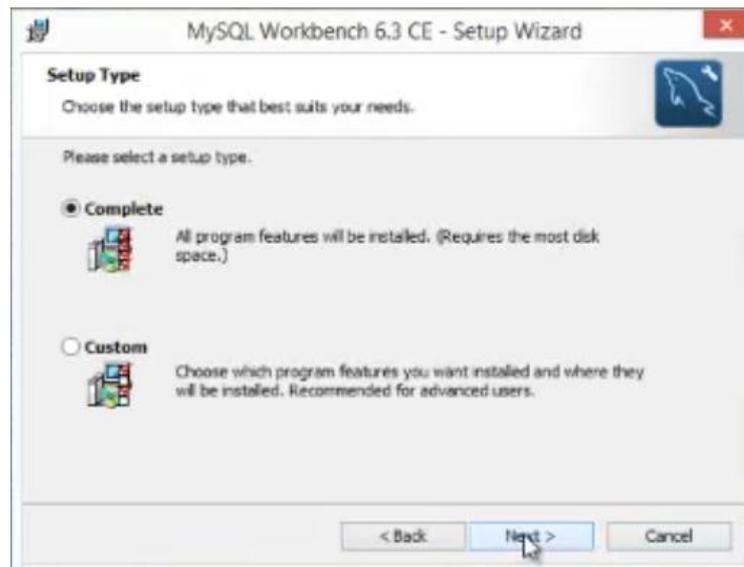
- 1) Obtener el archivo de instalación desde la web
<https://dev.mysql.com/downloads/workbench/>
- 2) Aparece la pantalla de bienvenida de la instalación



- 3) Al dar clic en siguiente nos pide especificar la carpeta de instalación



4) No pide el tipo de instalación que deseamos realizar



5) Seleccionar siguiente y ya está listo para comenzar la instalación



6) Y finalizará la instalación



ANEXO 5

CÓDIGO DEL PROGRAMA DE REPLICACIÓN ELABORADO EN PYTHON

```
#!/usr/bin/python
# -*- coding: iso-8859-1 -*-

import Tkinter
from User import User
from Articulo import Articulo
from Catalogos import Catalogos

class simpleapp_tk(Tkinter.Tk):
    def __init__(self,parent):
        Tkinter.Tk.__init__(self,parent)
        self.parent = parent
        self.minsize(width=300, height=300)
        self.initialize()

    def initialize(self):
        self.grid()

        self.labelVariable = Tkinter.StringVar()
        label = Tkinter.Label(self,textvariable=self.labelVariable,
                               anchor="center",fg="black")
        label.grid(column=0,row=0,columnspan=2,sticky='EW')
        self.labelVariable.set(u"MIGRAR DATOS DE LA BASE DE DATOS")

        self.grid_columnconfigure(0,weight=1)
        self.resizable(True,False)

        button = Tkinter.Button(self,text=u"CLIENTES",
                                command=self.OnButtonClick)
        button.grid(column=0,row=1)
        button1 = Tkinter.Button(self,text=u"ARTICULOS",
                                command=self.OnButtonClickArticulo)
        button1.grid(column=0,row=2)
        button2 = Tkinter.Button(self,text=u"IMAGEN",
                                command=self.OnButtonClickImagen)
        button2.grid(column=0,row=3)

    def OnButtonClick(self):
        User().mgrUser()

    def OnButtonClickImagen(self):
```

```
Catalogos().mgrDetalleArticulos()

def OnButtonClickArticulo(self):
    Articulo().mgrArticulos()
    Articulo().mgrSubClaseArticulos()
    Articulo().mgrColorArticulos()
    Articulo().mgrINArticulos()

if __name__ == "__main__":
    app = simpleapp_tk(None)
    app.title('Migracion de datos')
    app.mainloop()
```

ANEXO 6

TRIGGERS DE REPLICACIÓN

Agregar artículo

```
ALTER TRIGGER "add_articulos" AFTER INSERT
ORDER 1 ON "DBA"."articulos"
REFERENCING OLD AS old_articulo NEW AS new_articulo
FOR EACH ROW
BEGIN
    DECLARE datos LONG VARCHAR;
    DECLARE codigo LONG VARCHAR;
    DECLARE nombre LONG VARCHAR;
    DECLARE costo LONG VARCHAR;
    DECLARE observacion LONG VARCHAR;
    DECLARE costofob LONG VARCHAR;
    DECLARE color LONG VARCHAR;
    DECLARE clase LONG VARCHAR;
    DECLARE subclase LONG VARCHAR;
    DECLARE precio1 LONG VARCHAR;
    DECLARE precio2 LONG VARCHAR;
    DECLARE precio3 LONG VARCHAR;
    DECLARE precio4 LONG VARCHAR;
    DECLARE rutapath LONG VARCHAR;

    /* Type the trigger statements here */

    set codigo=new_articulo.codart;
    set nombre=new_articulo.nomart;
```

```
set costo=new_articulo.cospro;
set observacion=new_articulo.observ;
set costofob=new_articulo.cosfob;
set color=new_articulo.codcolor;
set clase=new_articulo.codcla;
set subclase=new_articulo.subcodcla;
set precio1=new_articulo.prec01;
set precio2=new_articulo.prec02;
set precio3=new_articulo.prec03;
set precio4=new_articulo.prec04;
```

```
SELECT path into rutapath FROM "DBA"."rutawebsservice" where id=1;
```

```
set datos=STRING('codigo=',codigo,
                '&nombre=',nombre,
                '&costo=',costo,
                '&obsevacion=',observacion,
                '&costofob=',costofob,
                '&color=',color,
                '&clase=',clase,
                '&subclase=',subclase,
                '&precio1=',precio1,
                '&precio2=',precio2,
                '&precio3=',precio3,
                '&precio4=',precio4);
```

```
BEGIN TRY
```

```
call "DBA"."add_articulos"(datos,rutapath);
```

```
END TRY
```

```
END
```

Eliminar artículo

```
ALTER TRIGGER "delete_articulos" before DELETE
ORDER 3 ON "DBA"."articulos"
REFERENCING OLD AS old_articulo
FOR EACH ROW
BEGIN
    DECLARE codigo LONG VARCHAR;
    DECLARE rutapath LONG VARCHAR;

    SELECT path into rutapath FROM "DBA"."rutawebsservice" where id=1;
    /* Type the trigger statements here */
    set codigo=old_articulo.codart;

    BEGIN TRY
        call "DBA"."delete_articulos"(codigo,rutapath);
    END TRY
    BEGIN CATCH

    END CATCH;

END
```

Actualizar artículo

```
ALTER TRIGGER "update_articulos" AFTER UPDATE
ORDER 2 ON "DBA"."articulos"
REFERENCING OLD AS old_articulo NEW AS new_articulo
FOR EACH ROW
```

BEGIN

DECLARE datos LONG VARCHAR;

DECLARE codigo LONG VARCHAR;

DECLARE codigoan LONG VARCHAR;

DECLARE nombre LONG VARCHAR;

DECLARE costo LONG VARCHAR;

DECLARE observacion LONG VARCHAR;

DECLARE costofob LONG VARCHAR;

DECLARE color LONG VARCHAR;

DECLARE clase LONG VARCHAR;

DECLARE subclase LONG VARCHAR;

DECLARE precio1 LONG VARCHAR;

DECLARE precio2 LONG VARCHAR;

DECLARE precio3 LONG VARCHAR;

DECLARE precio4 LONG VARCHAR;

DECLARE rutapath LONG VARCHAR;

/* Type the trigger statements here */

set codigo=old_articulo.codart;

set codigoan=codigo;

if old_articulo.codart<>new_articulo.codart then

 SET codigo=new_articulo.codart;

end if;

set nombre=old_articulo.nomart;

if old_articulo.nomart<>new_articulo.nomart then

 SET nombre=new_articulo.nomart;

end if;

```
set costo=old_articulo.cospro;  
if old_articulo.cospro<>new_articulo.cospro then  
    SET costo=new_articulo.cospro;  
end if;
```

```
set observacion = old_articulo.observ;  
if old_articulo.observ <> new_articulo.observ then  
    SET observacion = new_articulo.observ;  
end if;
```

```
set costofob=old_articulo.cosfob;  
if old_articulo.cosfob<>new_articulo.cosfob then  
    SET costofob=new_articulo.cosfob;  
end if;
```

```
set color=old_articulo.codcolor;  
if old_articulo.codcolor<>new_articulo.codcolor then  
    SET color=new_articulo.codcolor;  
end if;
```

```
set clase=old_articulo.codcla;  
if old_articulo.codcla<>new_articulo.codcla then  
    SET clase=new_articulo.codcla;  
end if;
```

```
set subclase=old_articulo.subcodcla;  
if old_articulo.subcodcla<>new_articulo.subcodcla then  
    SET subclase=new_articulo.subcodcla;
```

```
end if;
```

```
set precio1=old_articulo.precio1;
```

```
if old_articulo.precio1<>new_articulo.precio1 then
```

```
    SET precio1=new_articulo.precio1;
```

```
end if;
```

```
set precio2=old_articulo.precio2;
```

```
if old_articulo.precio2<>new_articulo.precio2 then
```

```
    SET precio2=new_articulo.precio2;
```

```
end if;
```

```
set precio3=old_articulo.precio3;
```

```
if old_articulo.precio3<>new_articulo.precio3 then
```

```
    SET precio3=new_articulo.precio3;
```

```
end if;
```

```
set precio4=old_articulo.precio4;
```

```
if old_articulo.precio4<>new_articulo.precio4 then
```

```
    SET precio4=new_articulo.precio4;
```

```
end if;
```

```
SELECT path into rutapath FROM "DBA"."rutawebsevice" where id=1;
```

```
set datos=STRING('codigo=',codigo,
```

```
    '&nombre=',nombre,
```

```
    '&costo=',costo,
```

```
    '&obsevacion=',observacion,
```

```
    '&costofob=',costofob,
```

```
'&color=',color,  
'&clase=',clase,  
'&subclase=',subclase,  
'&precio1=',precio1,  
'&precio2=',precio2,  
'&precio3=',precio3,  
'&precio4=',precio4);  
  
BEGIN TRY  
    call "DBA"."edit_articulos"(datos,codigoan,'application/x-www-form-  
urlencodedx',rutapath);  
END TRY  
  
BEGIN CATCH  
  
END CATCH;  
  
END
```