**INACAP TEMUCO**

INGENIERÍA EN INFORMÁTICA

**Taller Integral de Proyecto Informático**

Sistema Web para la Gestión de mantenimientos y garantías para proyectos post-implementación para la empresa Lazos

inacap

**Nombre Alumno (a): Tracy Solange Padilla Sierra**

**Luis Alberto Lizama Bustos**

**Juan Carlos Vallejos** **Obreque**

Nombre Profesor: Marisel Angélica Prado González

Sección: 075 Vespertino

Fecha: 14 de Octubre del 2013

Índice de contenido

[CAPÍTULO I INTRODUCCIÓN 4](#_Toc369476697)

[1.1. Introducción 4](#_Toc369476698)

[1.2. Definición del Problema 4](#_Toc369476699)

[CAPÍTULO II OBJETIVOS 5](#_Toc369476700)

[2.1. Título 5](#_Toc369476701)

[2.2. Objetivos 5](#_Toc369476702)

[2.2.1 Objetivo General 5](#_Toc369476703)

[2.2.2 Objetivos Específicos 5](#_Toc369476704)

[2.3. Alcance y Limitaciones 6](#_Toc369476705)

[2.3.1 Alcance 6](#_Toc369476706)

[2.3.1 Limitaciones 7](#_Toc369476707)

[CAPÍTULO III MARCO TEÓRICO 8](#_Toc369476708)

[3.1. Fundamentos Teóricos 8](#_Toc369476709)

[3.2. Antecedentes del Proyecto de SW 11](#_Toc369476710)

[3.3. Desarrollo del Problema 15](#_Toc369476711)

[3.4. Arquitectura 18](#_Toc369476712)

[3.5. Lenguajes de Programación 21](#_Toc369476713)

[CAPÍTULO IV METODOLOGÍA DEL DESARROLLO 25](#_Toc369476714)

[4.1. Estudio de Factibilidad 25](#_Toc369476715)

[4.1.1 Alternativas de Solución 25](#_Toc369476716)

[4.1.2 Evaluación de Alternativas 25](#_Toc369476717)

[4.1.3 Selección de Alternativa de Solución 25](#_Toc369476718)

[4.2. Planificación de la solución (Carta Gantt) 25](#_Toc369476719)

[4.3. Desarrollo del Requerimiento 26](#_Toc369476720)

[4.3.1 Definición de Requerimientos 26](#_Toc369476721)

[4.3.2 Análisis del Requerimiento 26](#_Toc369476722)

[4.4. Diseño Lógico 26](#_Toc369476723)

[4.4.1 Modelo de Casos de Uso 26](#_Toc369476724)

[4.4.2 Definición de Actores 26](#_Toc369476725)

[4.4.3 Definición de Casos de Uso 26](#_Toc369476726)

[4.4.4 Descripción de Casos de Uso 26](#_Toc369476727)

[4.4.5 Grafico de Casos de Uso 26](#_Toc369476728)

[4.4.6 Diagrama de Secuencia 26](#_Toc369476729)

[4.5. Diseño Físico 27](#_Toc369476730)

[4.5.1 Diseño Arquitectónico 27](#_Toc369476731)

[4.5.2 Definición del Modelo de Datos 27](#_Toc369476732)

[4.5.3 Diseño de la Interfaz 27](#_Toc369476733)

[4.6. Implementación 27](#_Toc369476734)

[CAPÍTULO V CONCLUSIONES 27](#_Toc369476735)

[5.1. Bibliografía 27](#_Toc369476736)

Índice de Tablas

[Tabla 1 16](#_Toc368917033)

Índice de Figuras

[Ilustración 1 – Empresa Lazos 10](#_Toc369476737)

[Ilustración 2 - Arquitectura 18](#_Toc369476738)

[Ilustración 3 - Capas fisica 20](#_Toc369476739)

[Ilustración 4 - Arquitectura logica 21](#_Toc369476740)

# CAPÍTULO I INTRODUCCIÓN

## Introducción

La empresa Lazos S.A. Solicita un sistema para la gestión de la post-entrega de sus sistemas desarrollados. Actualmente existe un sistema de ticket para la atención al cliente, pero no un sistema que cubra la necesidad tanto para la empresa ni para sus propios clientes.

## Definición del Problema

La idea es implementar sistema que cumpla con los estándares de escalabilidad de la empresa, ya que para ésta puede llegar a ser un gran problema la gestión de la post-implementación de sus proyectos, la información que se encuentra hoy en día se encuentra muy dispersa, porque cada jefe de proyecto maneja la post-entrega de los sistemas implementados, y el medio de comunicación para la gestión de la post-entrega es sólo es por correo electrónico o llamadas telefonicas, lo cual provoca una desorganización en la gestión de los proyectos entregados.

# CAPÍTULO II OBJETIVOS

## Título

Sistema Web para la Gestión de mantenimientos y garantías para proyectos post-implementación para la empresa Lazos.

## Objetivos

### 2.2.1 Objetivo General

Crear un sistema Web para efectuar la gestión de la post-entrega de sistemas desarrollados por la empresa Lazos.

### 2.2.2 Objetivos Específicos

* Establecer un modelo de datos escalable para las funcionalidades del sistema.
* Categorizar la centralización de las solicitudes del cliente estableciendo tipos y estados para el ciclo de vida de las solicitudes.
* Registrar la información de la gestión histórica de los proyectos guardando los distintos cambios que va sufriendo la solicitud en el tiempo.
* Indicar vía correo electrónico a los usuarios de los eventos que se producen en el sistema, utilizando el registro de los usuarios que se encuentren registrados en éste.
* Representar la información estadísticamente para ayudar a la toma de decisiones.

## Alcance y Limitaciones

### 2.3.1 Alcance

Inicialmente el proyecto comenzará con la toma de requerimientos, ésta se procesará previó al análisis y desarrollo de este sistema. Se realizarán reuniones con la contraparte (Lazos) y nuestro equipo de trabajo, las cuales quedarán formalizadas según se especifique en la carta Gantt.

A continuación se solicitará la información y/o documentación necesaria para la implementación y documentación de las maquetas, las cuales son necesarias para definir el alcance de los componentes visuales que conformarán el sistema. Durante la ejecución del proceso de análisis, el cliente podrá evaluar y comentar las maquetas para poder analizar y comprender qué es realmente lo que quiere y necesita.

Al momento de tener las maquetas aprobadas por el cliente, se pasará a la siguiente etapa, que será la implementación del documento maqueta. Este documento será el primer entregable a cliente. Cuando el cliente apruebe el documento se pasará a la siguiente etapa, la cual será comenzar con la confección del documento de análisis o diseño funcional del sistema (UML, flujos, etc.), el cual será el segundo entregable, y posterior a su confección se comenzará con el desarrollo de los componentes gráficos para la implementación de este sistema.

Posteriormente se construirá un modelo de datos para la implantación de este sistema correspondiente al tercer entregable al cliente. Dicho modelo debe cumplir con la nomenclatura estándar que utilice la contraparte para sus sistemas, la cual deberá entregar con anticipación, de lo contrario se definirá una interna.

Una vez finalizado el desarrollo de los tres primeros entregables, se generarán planes de pruebas para poder evaluar que lo implementado realmente cumpla con las funcionalidades solicitadas por el cliente. Estos serán el cuarto entregable al cliente.

Posteriormente al desarrollo del sistema se generarán manuales de usuarios (definidos por perfiles), y también se entregará el código fuente del sistema. Éstos serán el quinto y sexto entregable a cliente.

Finalmente al cierre del proyecto con la contraparte se capacitará a todo el personal que utilizará el sistema, para que los usuarios tengan los conocimientos necesarios al momento de manipular el sistema.

### 2.3.1 Limitaciones

* Cualquier requerimiento posterior a la toma de requerimientos no será considerado dentro del alcance de este sistema.
* Las funcionalidades en esta etapa del proyecto no se integrarán con otros sistemas o módulos.
* No se incluirá mantenimiento para las funcionalidades del sistema, sólo implementación.

# CAPÍTULO III MARCO TEÓRICO

## Fundamentos Teóricos

**Empresa Lazos**

La marca Lazos nace en 1997 como respuesta a los altos requerimientos tecnológicos en servicios de conectividad y comunicaciones, en particular en la implantación y ejecución del proyecto Enlaces. Lazos fue pionero (1997 - 1999) en ofrecer comercialmente en la IX Región el servicio de acceso a Internet conmutado (ISP). En el 2009 se decide constituir Lazos S.A. como empresa privada independiente manteniendo la participación accionaria principal. El 2011, como prueba de su madurez, se adjudica dos convenios marcos para vender sus servicios al Gobierno de Chile:

* Convenio marco de desarrollo de software
* Convenio marco de webhosting

**Misión**

Mejorar la eficiencia y eficacia de los procesos de gestión y comunicación de las organizaciones, a través del desarrollo, operación y mantención de servicios de software Web, con fuerte orientación a la satisfacción del cliente.

**Objetivos**

* Diseñar y construir soluciones innovadoras para el mejoramiento de la gestión.
* Buscar altos niveles de satisfacción de sus clientes.
* Mejorar la rentabilidad de las organizaciones.
* Fortalecer las capacidades de gestión de sus clientes.

**Cargos**

* Gerente.
* Gerente de Proyectos.
* Jefe de Operaciones.
* Jefe de Proyecto.
* Jefe de Desarrollo.
* Ingeniera QA.
* Administrador de Bases de Datos.
* Webmaster.
* Ingeniero de Software.
* Administrador de Sistemas y Redes.
* Diseñador de Software.
* Ejecutiva Soporte Técnico.
* Ejecutivo Soporte Remoto.
* Asistente Administrativo.



Ilustración 1 – Empresa Lazos

## Antecedentes del Proyecto de SW

Debido a la importancia que se genera en la atención al cliente y la organización de la información, es necesario mejorar las condiciones en que la empresa en cuanto a gestión de sus proyectos.

Hoy en día en el mercado existen varios tipos de sistemas que realizan procesos y tareas similares a las que requiere la empresa, pero no las realizan en su totalidad como la empresa lo requiere.

A continuación se detallan algunos de los sistemas que existen en el mercado:

**SAAS barlotask**

Es un sistema que permite rápida y ágilmente gestionar las tareas de su personal, asignado prioridades, responsables, fechas límites, entre otros. Por Internet y desde cualquier lugar.

Este sistema permite:

* Dar finalizada una nueva Tarea
* Mantenerla en estado borrador hasta “cerrarla”
* Adjuntar documentos
* Verificar Tareas anteriores
* Perfiles:
  + Colaborador
  + Gerente
  + Administrador/Control
* Avisos por mail de las nuevas tareas Asignadas
* Listado de Tareas Pendientes, ordenadas por prioridad o fecha

Ventajas del Sistema Workflow de Gestión de Tareas

* Mejora la asignación y Control de Tareas
* Mejora los tiempos de las presentaciones
* Centralización de la Información
* Descentralización de la actualización
* Rápida puesta en marcha

**OsTicket**

Es un sistema automatizado de soporte al cliente, que integra discretamente todos los tickets creados vía email o por formulario web dentro de una interface web simple:

* Administra, organiza y archiva todas las solicitudes de soporte.
* Los clientes, al abrir una consulta recibirán un e-mail de auto-respuesta.
* Los clientes pueden ver el estado de los tickets que han abierto y su historial en línea, utilizando su número de consulta.
* OsTicket es una aplicación de código abierto escrita principalmente usando el lenguaje de programación PHP.
* Es gratis

**Mantis Bug Tracker**

Es un software que constituye una solución muy completa para gestionar tareas en un equipo de trabajo. Es una aplicación OpenSource hecha en php y mysql, destaca por su facilidad y flexibilidad de instalar y configurar. Esta aplicación se utiliza para testear soluciones, hacer un registro histórico de alteraciones y gestionar equipos remotamente.

Una de las características de Mantis es que cuenta con una gran cantidad de posibilidades para su configuración:

* Permitir configurar la transición de estados (abierto, encaminado, testear, devuelto, cerrado, reabierto...).
* Puede especificar un número indeterminado de estados para cada tarea (abierta, encaminada, testear, devuelta, cerrada, reabierta…).
* Permite introducir diferentes perfiles (programador, tester, coordinador, visualizador…).

Mantis incluye filtros, un sistema de búsqueda, tiene soporte para varios idiomas y también informa por e-mail la resolución de los errores de los tickets que se han informado.

**Projects and teamwork**

Es un sistema que da solución a la gestión de proyectos, tanto en entornos distribuidos como centralizados, que requieren de escalabilidad y flexibilidad, también incluye una nube. puede coordinar todo su equipo, planificar los proyectos, hacer seguimiento, establecer prioridades y aumentar su productividad, con una solución que se adapta a cada organización y no requiere largos periodos de formación o configuración.

Algunas características incluidas:

* Comunicación
* Editor Gantt avanzado
* Importación desde MS Project
* Equipo de Proyecto
* Multi-idioma
* Multi-moneda
* Estimación de esfuerzo
* Seguimiento de las tareas
* Coste de los recursos
* Estructura Organizativa
* Procesos de Negocio
* Seguimiento de Proyectos

En nuestra investigación no se encontraron antecedentes de que exista un sistema que cumpla en su totalidad con los requisitos o funciones que la empresa necesita, ya que nuestro sistema estara acorde a los requerimientos solicitados, cumpliendo la función de dos sistemas en uno solo, ya que en el mercado se encuentran por separado: La gestión del cliente o soporte para el cliente y la gestión de tareas para el equipo de trabajo.

Los sistemas mencionados anteriormente son una opción que existe, pero sus valores son altamente elevados, populares que son utilizados por organizaciones grandes y están fuera del presupuesto de la empresa. A excepción de osticket ya que es completamente gratuito.

Pero la diferencia que existe entre los sistemas expuestos anteriormente en relación con nuestro sistema son las siguientes:

* + Está hecho a medida.
  + Las funciones están directamente relacionadas a las tareas que día a día realiza la empresa.
  + Los procesos son solamente los que la empresa solicita, a diferencia de los demás sistemas que se enfocan en un entorno general para organizaciones con funciones comunes, ya que para Lazos no son necesarias, como por ejemplo: gestiones para un gerente, o entidades externas.
  + Costos elevados por el prestigio y las funcionalidades que integra el sistema.

## Desarrollo del Problema

La empresa Lazos, dentro de su ámbito de negocio, su principal componente es el desarrollo de software, en el cual entregan sistemas a distintos clientes para su implementación. Una vez que el ciclo de desarrollo concluye comienza un periodo denominado “Garantía”, en el cual los clientes pueden reportar errores que presente el sistema, o pueden presentar nuevos requerimientos para agregarle a éste.

Dicho levantamiento de requerimientos o reporte de errores, actualmente para los sistemas que se encuentran en periodo de garantía, o que ya han sido implementados, se realiza por medios tales como conversaciones telefónicas o envío de correos electrónicos. Por lo cual, no queda respaldo centralizado de dicha información de manera que todo el personal interesado de la empresa esté informado, ni tampoco dicha información queda disponible para rápido acceso en el futuro.

Considerando lo anterior, el problema consiste en no tener un sistema que gestione el flujo de trabajo posterior a la implementación de los sistemas desarrollados por Lazos. Por lo tanto se propone crear un sistema acorde a la problemática detectada, en el cual se deben incluir las siguientes funcionalidades agrupadas por perfiles:

**Perfil Administrador:**

* Ingresar proyectos.
* Crear a los usuarios y clientes.
* Aprobar o rechazar las solicitudes de los clientes.
* Asignar una tarea (se puede asignar a otro administrador o desarrollador).
* Puede desempeñar las mismas acciones del perfil desarrollador.

**Perfil Encargado de las solicitudes:**

* Ingresar proyectos.
* Aprobar o rechazar las solicitudes de los clientes.
* Asignar una tarea (se puede asignar a otro encargado o desarrollador).
* Aprobar o rechazar la resolución de solicitudes.
* Puede desempeñar las mismas acciones del perfil desarrollador.

**Perfil Desarrollador:**

* Revisar la solicitud y su historico.
* Resolver y cambiar de estado.
* Adjuntar archivo y comentarios.

**Funcionalidades Generales:**

* Notificación vía correo electrónico de los cambios de estados a los usuarios asignados.
* Súper usuario puede crear usuarios internos como externos y puede crear proyectos.
* Todos los usuarios deben agregar un comentario al cambiar de estado.

Considerando la separación lógica agrupada por funcionalidades, las cuales responden a los actores involucrados internamente en el negocio de la empresa, es que se pretende tener un sistema que gestione el flujo de trabajo posterior a la implementación de los sistemas desarrollados por Lazos. En este sistema existirá un registro histórico de la información de los proyectos, utilizará flujos de trabajo para poder saber en tiempo real y de manera centralizada el estado de las solicitudes y errores reportados para los proyectos que se encuentran en garantía.

## Arquitectura

Considerando las necesidades y la naturaleza Web de la solución propuesta, es que en terminos de arquitectura, el desarrollo del software se basará en el paradigma cliente servidor, más especificamente en la lógica por capas.



Ilustración 2 - Arquitectura

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

En terminos fisicos la arquitectura se dividira en tres capas, las cuales son: la capa del cliente o capa navegador, la cual representa todo lo que el usuario visualiza. La segunda capa corresponde al servidor Web, en el cual se ejecuta la logica de negocio y se procesan las peticiones del cliente. Y la tercera capa corresponde al servidor de base de datos, donde se guardan los datos de manera persistente.



Ilustración 3 - Capas fisica

En terminos logicos la aplicación se separa en cuatro capas, las cuales son: la capa de presentación, que corresponde a todo lo que visualiza el usuario, es decir las vistas. La capa de negocio, en la cual se ejecuta toda la logica especifica del sistema, como por ejemplo las validaciones y reglas del negocio internas de la empresa. La capa de acceso a datos, en la cual se encapsulan las llamadas y transacciones que se realizan con el servidor de base de datos. Y finalmente la capa de entidades de negocio, en la cual se definen los objetos comunes entre las tres primeras capas para trasnportar la información entre estas de manera coherente.



Ilustración 4 - Arquitectura logica

## Lenguajes de Programación

A continuacion se describen los lenguajes de programación más popularos o accesibles para desarrollar el software para la solución propuesta, de manera que se pueda elegir uno considerando sus caracteristicas de acuerdo a las necesidades del proyecto.

**PHP**

PHP es un lenguaje que nació orientado a la programación web dinámica, se utiliza principalmente en aplicaciones de servidor, y el código puede ser también incrustado en contenido HTML. PHP es un lenguaje ampliamente utilizado en el desarrollo de aplicaciones Web debido a su naturaleza basada en software libre.

Su sintaxis se asemeja a la de los lenguajes C, Java y Perl. La incrustación de código dentro del contenido HTML se realiza al utilizar las etiquetas <?php (código) ?>, es aquí donde se definen las clases en el caso de la orientación a objetos que implementa PHP desde las últimas versiones.

**.NET**

Microsoft.net es una plataforma de desarrollo y ejecución de aplicaciones que se incluye en Windows Server 2008, Windows Vista y Windows 7, o se puede instalar en Windows Xp, Windows Server 2003, y sus principales características son:

* Las aplicaciones de .NET se ejecutan en un entorno aislado del sistema operativo denominado runtime, característica que las hace ser portables.
* Es orientado a objetos y multi lenguaje.
* Esta orientado al desarrollo de aplicaciones empresariales complejas.
* Proporciona un modelo de desarrollo para distintos dispositivos de hadware (pc, tablets, etc.) y software (Windows, Web, consola, móviles, etc.).
* Se puede integrar con aplicaciones desarrolladas en COM y viceversa
* Integra compatibilidad con otros sistemas operativos al implementar estándares como xml, soap, wsdl, etc.

La plataforma de .NET esta formada principalmente por:

* El Framework, que es el componente sobre el cual se desarrollan y ejecutan las aplicaciones.
* El runtime, que se denomina CLR (common language runtime) cuya función es administrar memoria, aislar el funcionamiento de las aplicaciones, brindar seguridad en la ejecución de estas y generar código nativo o compilado.
* El BLC (biblioteca de clase base) que proporciona todas las clases necesarias para implementar las funcionalidades mas comunes.
* Lenguajes de programación y compiladores, ya que .NET es multilenguaje, los lenguajes deben seguir un estándar llamado CLS (common language specification) y así son compatible entre si, ejemplos: visual basic.net, c-sharp, net.cobol, delphi, c++, python, etc.
* Herramientas y documentación, .NET proporciona utilidades para documentación y otras tareas de forma mas practica.

Actualmente .NET soporta desarrollo de aplicaciones Web montadas en servidores IIS utilizando la arquitectura ASP.NET, en la cual se puede desarrollar en los lenguajes C# o VB. Las aplicaciones .NET no estan basadas en software libre, por lo que se deben costear las licencias de los servidores y otros componentes.

**JAVA**

Java es un lenguaje orientado a objetos desarrollado por Sun Microsystems en los 90, nace principalmente del lenguaje C y C++, pero simplifica el modelo de objetos, tiene como filosofía la reutilización de código y le quita la responsabilidad al programador de manejar la memoria a través de punteros, ya que java discrimina cuando usar el paso por referencia o por valor, así como también el programador ya no debe escribir los destructores porque java libera memoria de forma automática con el garbage collector.

Las características de Java es que sus aplicaciones están compiladas en un bytecode el cual corre en una maquina virtual específica para una plataforma, por lo cual el código fuente será el mismo independiente del sistema en el que se ejecute lo que lo hace ser portable. Java también soporta programación estructurada al implementar el método main() como en C, y soporta de la orientación a objetos también las clases, polimorfismo, etc. pero a diferencia de C++ no existe la herencia múltiple, lo cual se trata de contrarrestar con la implementación de las interfaces. También en Java los tipos de datos primitivos existen al igual que en C y también existen sus equivalentes en objeto con métodos y propiedades asociadas que heredan de Object.

Java al tener estas características de automatización como el no manejar directamente la memoria, el garbage collector, mas la maquina virtual, hace que sea un lenguaje menos eficiente en términos de velocidad y recursos, pero más potente en el desarrollo de aplicaciones empresariales complejas.

Actualmente Java es muy utilizado en aplicaciones Web, siendo estas desarrollladas bajo distintos Frameworks existentes como por ejemplo Servlets, JSP, Struts, o Spring. Java no esta basado en codigo libre y se deben costear las licencias implicadas.

# CAPÍTULO IV METODOLOGÍA DEL DESARROLLO

## Estudio de Factibilidad

### 4.1.1 Alternativas de Solución

### 4.1.2 Evaluación de Alternativas

#### 4.1.2.1 Factibilidad Técnica

#### 4.1.2.2 Factibilidad Económica

#### 4.1.2.3 Factibilidad Operacional

#### 4.1.2.4 Comparativa de las Alternativas

### 4.1.3 Selección de Alternativa de Solución

#### 4.1.3.1 Beneficios de la Solución

## Planificación de la solución (Carta Gantt)

## Desarrollo del Requerimiento

### 4.3.1 Definición de Requerimientos

### 4.3.2 Análisis del Requerimiento

#### 4.3.2.1 Requisitos Funcionales

#### 4.3.2.2 Requisitos No Funcionales

## Diseño Lógico

### 4.4.1 Modelo de Casos de Uso

### 4.4.2 Definición de Actores

### 4.4.3 Definición de Casos de Uso

### 4.4.4 Descripción de Casos de Uso

### 4.4.5 Grafico de Casos de Uso

### 4.4.6 Diagrama de Secuencia

## Diseño Físico

### 4.5.1 Diseño Arquitectónico

### 4.5.2 Definición del Modelo de Datos

#### 4.5.2.1 Modelo de Datos

#### 4.5.2.2 Diccionario de Datos

### 4.5.3 Diseño de la Interfaz

#### 4.5.3.1 Interfaz 1

#### 4.5.3.2 Interfaz 2

#### 4.5.3.n Interfaz n

## Implementación

# CAPÍTULO V CONCLUSIONES

|  |  |
| --- | --- |
|  |  |
|  |  |

Tabla 1

## Bibliografía