**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 S.A.

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: 02 de Diciembre del 2013

Índice de contenido

[CAPÍTULO I INTRODUCCIÓN 5](#_Toc373614336)

[1.1. Introducción 6](#_Toc373614337)

[1.2. Definición del Problema 6](#_Toc373614338)

[CAPÍTULO II OBJETIVOS 7](#_Toc373614339)

[2.1. Título 7](#_Toc373614340)

[2.2. Objetivos 7](#_Toc373614341)

[2.2.1 Objetivo General 7](#_Toc373614342)

[2.2.2 Objetivos Específicos 7](#_Toc373614343)

[2.3. Alcance y Limitaciones 7](#_Toc373614344)

[2.3.1 Alcance 8](#_Toc373614345)

[2.3.1 Limitaciones 9](#_Toc373614346)

[CAPÍTULO III MARCO TEÓRICO 9](#_Toc373614347)

[3.1. Fundamentos Teóricos 10](#_Toc373614348)

[3.2. Antecedentes del Proyecto de SW 13](#_Toc373614349)

[3.3. Desarrollo del Problema 17](#_Toc373614350)

[3.4. Arquitectura 20](#_Toc373614351)

[3.5. Lenguajes de Programación 24](#_Toc373614352)

[CAPÍTULO IV METODOLOGÍA DEL DESARROLLO 27](#_Toc373614353)

[4.1. Estudio de Factibilidad 28](#_Toc373614354)

[4.1.1 Alternativas de Solución 28](#_Toc373614355)

[4.1.2 Evaluación de Alternativas 30](#_Toc373614356)

[4.1.3 Selección de Alternativa de Solución 45](#_Toc373614357)

[4.2. Planificación de la solución (Carta Gantt) 47](#_Toc373614358)

[4.3. Desarrollo del Requerimiento 51](#_Toc373614359)

[4.3.1 Definición de Requerimientos 51](#_Toc373614360)

[4.3.2 Análisis del Requerimiento 52](#_Toc373614361)

[4.4. Diseño Lógico 55](#_Toc373614362)

[4.4.1 Modelo de Casos de Uso 56](#_Toc373614363)

[4.4.2 Definición de Actores 58](#_Toc373614364)

[4.4.3 Definición de Casos de Uso 59](#_Toc373614365)

[4.4.4 Descripción de Casos de Uso 60](#_Toc373614366)

[4.4.5 Gráfico de Casos de Uso 76](#_Toc373614367)

[4.4.6 Diagrama de Secuencia 78](#_Toc373614368)

[4.5. Diseño Físico 85](#_Toc373614369)

[4.5.1 Diseño Arquitectónico 85](#_Toc373614370)

[4.5.2 Definición del Modelo de Datos 85](#_Toc373614371)

[4.5.3 Diseño de las Interfaces 104](#_Toc373614372)

[4.6. Implementación 117](#_Toc373614373)

[CAPÍTULO V CONCLUSIONES 117](#_Toc373614374)

[5.1. Bibliografía 117](#_Toc373614375)

Índice de Tablas

[Tabla 1 - Recurso Humano 33](#_Toc373614376)

[Tabla 2 - Recursos Consumibles 34](#_Toc373614377)

[Tabla 3 - Recursos Teconológicos 34](#_Toc373614378)

[Tabla 4 - Recursos Teconológicos Depreciación 35](#_Toc373614379)

[Tabla 5 - Recursos de Operación 36](#_Toc373614380)

[Tabla 6 - Resumen del presupuesto del proyecto 37](#_Toc373614381)

[Tabla 7 - Costos del proyecto 40](#_Toc373614382)

[Tabla 8 - CU-01 60](#_Toc373614383)

[Tabla 9 - CU-02 61](#_Toc373614384)

[Tabla 10 - CU-03 62](#_Toc373614385)

[Tabla 11 - CU-04 63](#_Toc373614386)

[Tabla 12 - CU-05 64](#_Toc373614387)

[Tabla 13 - CU-06 65](#_Toc373614388)

[Tabla 14 - CU-07 66](#_Toc373614389)

[Tabla 15 - CU-08 67](#_Toc373614390)

[Tabla 16 - CU-09 67](#_Toc373614391)

[Tabla 17 - CU-10 68](#_Toc373614392)

[Tabla 18 - CU-11 69](#_Toc373614393)

[Tabla 19 - CU-12 70](#_Toc373614394)

[Tabla 20 - CU-13 71](#_Toc373614395)

[Tabla 21 - CU-14 73](#_Toc373614396)

[Tabla 22 - CU-15 74](#_Toc373614397)

[Tabla 23 - CU-16 75](#_Toc373614398)

[Tabla 24 - CU-17 76](#_Toc373614399)

Índice de Figuras

[Ilustración 1 – Empresa Lazos 12](#_Toc373614509)

[Ilustración 2 - Arquitectura 20](#_Toc373614510)

[Ilustración 3 - Capas fisica 22](#_Toc373614511)

[Ilustración 4 - Arquitectura logica 23](#_Toc373614512)

[Ilustración 5 - Diseño en cascada 28](#_Toc373614513)

[Ilustración 6: Gantt, Parte 1 48](#_Toc373614514)

[Ilustración 7: Gantt, Parte 2 49](#_Toc373614515)

[Ilustración 8: Gantt, Parte 3 50](#_Toc373614516)

[Ilustración 9: Gantt, Parte 4 51](#_Toc373614517)

[Ilustración 10 - Casos de uso 1 56](#_Toc373614518)

[Ilustración 11 - Casos de uso 2 56](#_Toc373614519)

[Ilustración 12 - Casos de uso 3 57](#_Toc373614520)

[Ilustración 13 - Casos de uso 4 57](#_Toc373614521)

[Ilustración 14 - Gráfico de Casos de Uso 77](#_Toc373614522)

[Ilustración 15 - Diagrama de secuencia 1 78](#_Toc373614523)

[Ilustración 16 - Diagrama de secuencia 2 78](#_Toc373614524)

[Ilustración 17 - Diagrama de secuencia 3 79](#_Toc373614525)

[Ilustración 18 - Diagrama de secuencia 4 79](#_Toc373614526)

[Ilustración 19 - Diagrama de secuencia 5 80](#_Toc373614527)

[Ilustración 20 - Diagrama de secuencia 6 80](#_Toc373614528)

[Ilustración 21 - Diagrama de secuencia 7 81](#_Toc373614529)

[Ilustración 22 - Diagrama de secuencia 8 81](#_Toc373614530)

[Ilustración 23 - Diagrama de secuencia 9 81](#_Toc373614531)

[Ilustración 24 - Diagrama de secuencia 10 82](#_Toc373614532)

[Ilustración 25 - Diagrama de secuencia 11 82](#_Toc373614533)

[Ilustración 26 - Diagrama de secuencia 12 83](#_Toc373614534)

[Ilustración 27 - Diagrama de secuencia 13 83](#_Toc373614535)

[Ilustración 28 - Diagrama de secuencia 14 84](#_Toc373614536)

[Ilustración 29 - Diagrama de secuencia 15 84](#_Toc373614537)

[Ilustración 30 - Acceso al sistema 104](#_Toc373614538)

[Ilustración 31 - Ingreso al sistema 105](#_Toc373614539)

[Ilustración 32 - Mis datos 105](#_Toc373614540)

[Ilustración 33 - Cambio de contraseña 106](#_Toc373614541)

[Ilustración 34 - Registrar proyecto 107](#_Toc373614542)

[Ilustración 35 - Búsqueda básica 108](#_Toc373614543)

[Ilustración 36 - Búsqueda avanzada 109](#_Toc373614544)

[Ilustración 37 - Modificar proyecto 110](#_Toc373614545)

[Ilustración 38 - Agregar usuario 111](#_Toc373614546)

[Ilustración 39 - Búsqueda básica 112](#_Toc373614547)

[Ilustración 40 - Búsqueda avanzada 112](#_Toc373614548)

[Ilustración 41 - Modificar usuario 113](#_Toc373614549)

[Ilustración 42 - Asignar perfiles 113](#_Toc373614550)

[Ilustración 43 - Registrar solicitud 114](#_Toc373614551)

[Ilustración 44 - Búsqueda básica 114](#_Toc373614552)

[Ilustración 45 - Búsqueda avanzada 115](#_Toc373614553)

[Ilustración 46 - Búsqueda básica 116](#_Toc373614554)

[Ilustración 47 - Búsqueda avanzada 117](#_Toc373614555)

# 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, ya que cada jefe de proyecto maneja la post-entrega de sus sistemas implementados, y el medio de comunicación para la gestión de la post-entrega es sólo es por correo electrónico o llamadas telefónicas, 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 S.A.

## 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 S.A.

### 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 se van efectuando en la solicitud durante el tiempo.
* Indicar vía correo electrónico a los usuarios registrados en el sistema de los eventos que sufre la(s) solicitud(es), 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 S.A.) y nuestro equipo de trabajo, las cuales se encuentran estipuladas según la especificación que se detalla en la carta Gantt, ver las siguientes ilustraciones: Ilustración 6, Ilustración 7, Ilustración 8, Ilustración 9.

A continuación se solicitará la información y/o documentación necesaria para la implementación y documentación de las maquetas (pantallas), 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 sé incluirá mantenimiento para las funcionalidades del sistema, sólo implementación.

# CAPÍTULO III MARCO TEÓRICO

## Fundamentos Teóricos

**Empresa Lazos S.A.**

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 [[1]](#footnote-1):

* 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 sitio[[2]](#footnote-2).

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[[3]](#footnote-3):

* 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 un sistema gratuito.

**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[[4]](#footnote-4).

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[[5]](#footnote-5).

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 (Lazos S.A). necesita, ya que nuestro sistema estará 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 esta 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 S.A. 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 S.A, 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 y/o envío de correos electrónicos, es decir, no queda respaldo centralizado de dicha información de manera que todo el personal interesado de la empresa esté informado, como tampoco esta 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 la empresa. 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 histórico.
* 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 decir, se pretende tener un sistema que gestione el flujo de trabajo posterior a la implementación de los sistemas desarrollados para esta empresa.

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 términos de arquitectura, el desarrollo del software se basará en el paradigma cliente servidor, más específicamente 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 términos físicos la arquitectura se dividirá 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 lógica 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 física

En términos lógicos 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 lógica 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.
* La capa de entidades de negocio, en la cual, se definen los objetos comunes entre las tres primeras capas para transportar la información entre éstas de manera coherente.



Ilustración 4 - Arquitectura logica

## Lenguajes de Programación

A continuación se describen los lenguajes de programación más populares o accesibles para desarrollar el software, es decir, la solución propuesta de manera que se pueda elegir uno considerando sus características de acuerdo a las necesidades de este 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[[6]](#footnote-6).

Su sintaxis se asemeja a 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 hacen 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 está 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 éstas 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, como por ejemplo: 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 práctica.

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 están basadas en software libre, por lo que se deben costear las licencias de los servidores y otros componentes[[7]](#footnote-7).

**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 máquina virtual específica para una plataforma, es decir, el código fuente será el mismo independiente del sistema en el que se ejecute por lo cual 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, más la máquina 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 éstas desarrolladas bajo distintos Frameworks existentes, como por ejemplo: Servlets, JSP, Struts, o Spring. Java no está basado en código libre y se deben costear las licencias implicadas[[8]](#footnote-8).

# CAPÍTULO IV METODOLOGÍA DEL DESARROLLO

Considerando que el alcance del desarrollo del software para la solución es limitado, se optará por la metodología de desarrollo en cascada. Esta metodología de desarrollo es útil cuando los requerimientos son fijos, ya que el avance del desarrollo será de manera lineal.

El paradigma del modelo en cascada determina un enfoque sistemático y ordenado, secuencial, donde se inicia con la especificación de los requerimientos, luego la planificación y análisis, modelado, construcción y finalmente el despliegue del sistema con su periodo de pruebas y soporte.

Un inconveniente importante en este paradigma es que el cliente no puede visualizar o tener una idea de la funcionalidad real del sistema hasta que éste se encuentre terminado, desventaja la cual pretendemos suplir con un riguroso análisis inicial de la toma de requerimiento y confección de las maquetas del sistema.



Ilustración 5 - Diseño en cascada

## Estudio de Factibilidad

### 4.1.1 Alternativas de Solución

Considerando los requisitos a cubrir por el sistema, en base a lo solicitado por la contraparte (Lazos S.A), las alternativas de solución a analizar las diferenciaremos en dos opciones:

* La primera, es optar por una alternativa de software que se pueda adquirir en el mercado.
* La segunda, es optar por desarrollar el software a medida.

**Software existente**

Para esta alternativa de solución se deberá tener en consideración los productos ya existentes en el mercado actual, descritos en el CAPÍTULO III MARCO TEÓRICO, se debe considerar sus características, costos, y si cumplen o no con suplir los requerimientos que abarca el sistema.

**Software a medida**

Para esta alternativa de solución se deberá tener en consideración los lenguajes de programación, descritos en el CAPÍTULO III MARCO TEÓRICO, y la metodología de desarrollo de software mencionados en la introducción de este capítulo.

A continuación se detallan las alternativas de solución que se consideran las más óptimas para el desarrollo de este proyecto:

* + Alternativa N° 1: .NET y SLQ Server. Desarrollo de un sistema web en .net con el gestor de base de datos “SLQ Server”. De esta forma generar un sistema de interfaz sencilla y amigable.
  + Alternativa N° 2: PHP con MySQL. Desarrollo de un sistema web en PHP con el gestor de base de datos “MySQL”, generando un sistema seguro, amigable y óptimo integrando diseños y estilos para una mejor organización de la información.

### 4.1.2 Evaluación de Alternativas

A continuación se describe la evaluación y especificación de las dos alternativas descritas anteriormente para la solución, considerando los aspectos técnicos, económicos y operacionales de cada una de éstas para finalmente realizar una comparación.

#### 4.1.2.1 Factibilidad Técnica

La factibilidad técnica tiene por objetivos evaluar si se cuenta con los recursos, conocimientos, herramientas, experiencia y el personal para llevar a cabo el diseño, implementación y operación de la soluciones propuestas.

Evaluando la factibilidad técnica de la primera alternativa (Software existente) encontramos que técnicamente es más complejo para nuestro equipo de trabajo analizar software de terceros, lo que incrementaría los tiempos de desarrollo si se desea modificar, por ejemplo: El sistema OsTicket. En cambio si se opta por un sistema de código cerrado o de pago no existirían incovenientes técnicos ya que el desarrollo sólo se limitaria a instalar y configurar dichas soluciones.

Evaluando la factibilidad técnica de la segunda alternativa (Software a medida) encontramos que técnicamente es posible llevar a cabo el desarrollo del sistema, ya que nuestro equipo de trabajo tiene experiencia desarrollando sistemas en código libre y código licenciado.

También es importante destacar que cliente final, es decir, la empresa Lazos S.A, se dedica al desarrollo de software, por lo que ellos mismos también poseen el conocimiento técnico para posteriormente mantener el sistema.

Respecto al hardware, como se ha definido en el alcance del proyecto, ya estará definido por el cliente, por lo que no existirá impedimento de este tipo para la implementación de la primera o segunda solución.

En relación a las tecnologías, actualmente los software de desarrollo o documentación asociada a las tecnologías son de disposición libre, y no existen impedimentos para manejarlos, ya que nuestro equipo de trabajo cuenta con los conocimientos necesarios para el manejo de herramientas de desarrollo.

Para la implementación y desarrollo de las alternativas propuestas, se necesitará lo siguiente:

**Alternativa N° 1 .NET con SQL** **Server**

|  |  |
| --- | --- |
| Detalle | Cantidad |
| Servidor Procesador I3 2.7 GHZ, RAM 8 Gb DDR3 HD 2,0 TB | 1 |
| Licencia SLQ Server | 1 |
| Antivirus para Servidor | 1 |
| Licencia Windows Server 2008 r2 | 1 |
| Licencia Visual Studio | 1 |

**Alternativa N° 2 PHP con MySQL**

|  |  |
| --- | --- |
| Detalle | Cantidad |
| Servidor Procesador I3 2.7 GHZ, RAM 8 Gb DDR3 HD 2,0 TB | 1 |
| Licencia MySQL | 1 |
| Antivirus para Servidor | 1 |
| Licencia Windows Server 2008 r2 | 1 |

En la alternativa N° 1 nos encontramos con varios puntos a favor para el desarrollo de este proyecto, pero como la empresa Lazos S.A cuenta con servidores para la implementación del sistema y soporte en MySQL, optaremos por la opción N°2.

#### 4.1.2.2 Factibilidad Económica

En relación a las alternativas de solución. A continuación se evalúa la factibilidad económica para cada alternativa propuesta.

**Alternativa N° 1 .net con SQL Server**

|  |  |  |
| --- | --- | --- |
| Detalle | Cantidad | Costo $ |
| Servidor Procesador I3 2.7 GHZ, RAM 8 Gb DDR3, HD 2,0 TB | 1 | 500.000 |
| Licencia SQL Server | 1 | 163.000 |
| Antivirus para Servidor | 1 | 203.000 |
| Licencia Windows Server 2008 r2 | 1 | 522.000 |
| Licencia Visual Studio | 1 | 253.000 |
| Total | | $1.641.000 |

**Alternativa N° 2 PHP con MySQL**

|  |  |  |
| --- | --- | --- |
| Detalle | Cantidad | Costo $ |
| Servidor Procesador I5 2.7 GHZ, RAM 8 Gb DDR3, HD 2,0 TB | 1 | 500.000 |
| Licencia MySQL | 1 | 153.000 |
| Antivirus para Servidor | 1 | 203.000 |
| Licencia Windows Server 2008 r2 | 1 | 522.000 |
| Total | | $1.378.000 |

Nota:

* + Cabe mencionar que la empresa cuenta en su totalidad con el equipamiento ya sea en hardware y software para la alternativa N°2, ya que es una empresa dedicada a este rubro del desarrollo de sistemas, por lo tanto, ellos se harán cargo de la implementación y mantenimiento del sistema.

**Costos de Desarrollo del Proyecto**

**Recurso Humano**

En este estudio se incluye el costo del recurso humano para dar una aproximación más real del valor del proyecto.

La mano de obra es el recurso más importante y a la vez el más costoso que se ve involucrado en la realización de cualquier tipo de proyecto, esto no excluye a los proyectos de tipo informático para determinar el salario mensual de cada uno de los desarrolladores se ha decidido tomar como referencia el equivalente al sueldo otorgado a los empleados de la empresa Lazos S.A. El cual se especificará a continuación:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Recurso | Cantidad | Meses | Salario Mensual ($) | Total ($) |
| Jefe de Proyecto | 1 | 2.5 | 800.000 | 2.000.000 |
| Testing | 1 | 1 | 350.000 | 350.000 |
| Programadores | 2 | 2.5 | 400.000 | 1.000.000 |
| Total | | | | $3.350.000 |

Tabla 1 - Recurso Humano

**Recursos Consumibles**

Los recursos consumibles que serán utilizados para la realización del proyecto se detallarán a continuación: en base a los requerimientos solicitados por parte de la empresa en base a proyectos realizados anteriormente y los precios se obtuvieron de cotizaciones realizadas.

|  |  |  |  |
| --- | --- | --- | --- |
| Detalle | Cantidad | Costo Unitario ($) | Total ($) |
| Fotocopias | 1000 | 25 | 2.500 |
| Resmas de papel | 1 | 2.990 | 2.990 |
| Cartuchos de tinta de color | 1 | 12.990 | 12.990 |
| Cartucho de tinta negro | 2 | 9.990 | 19.980 |
| Gastos Varios |  | 15.000 | 15.000 |
| Total | | | $53.460 |

Tabla 2 - Recursos Consumibles

Nota:

* + Una cantidad aproximada de 1000 fotocopias para lo que es la entrega de manuales e informes que tiene un valor aproximado de $2.500.
  + Una resmas de papel para la impresión del informe, documentos y anexos.
  + Cartuchos de tinta para la impresora de color y negro para realizar la impresión del informe, documentos y anexos.
  + Gastos varios incluye: anillado, carpetas para la entrega de manuales e informes, entre otros.

**Recursos Teconológicos**

Para el desarrollo del proyecto es necesario la adquisición de hardware que se detallará en el siguiente cuadro:

|  |  |  |  |
| --- | --- | --- | --- |
| Detalle | Cantidad | Costo Unitario ($) | Total ($) |
| Servidores (web y aplicaciones) | - | - | - |
| Notebook | 3 | 450.000 | 1.350.000 |
| Hosting | 1 | - | - |
| Dominio | 1 | 18.900 | 18.900 |
| Total | | | $1.368.900 |

Tabla 3 - Recursos Teconológicos

Nota:

* + Servidores (web y aplicaciones). La empresa cuenta con sus propios servidores web y aplicaciones con sus licencias respectivas, de esta forma se obtiene un ahorro significativo en este punto, ya que son recursos altamente costosos para una implementación propia. Los costos disminuyen al contratar servicios externos, pero no es necesario invertir en estos.
  + El hosting es otorgado por la misma empresa donde no se debe realizar inversiones en la implementación del proyecto.
  + El dominio tiene un valor aproximado de $18.900 por 2 años. Fue cotizado en ww.nic.cl.

Los notebook que serán utilizadas para el desarrollo del proyecto se detalla en el punto Factibilidad Técnica, los valores correspondientes a la depreciación del equipo se pueden apreciar en el siguiente cuadro que muestra los costos de los recursos tecnológicos utilizados para el proyecto.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Detalle | Meses | Costos ($) | Total ($) | |
| Depreciación del Equipo | 2.5 | 45.000 | 84.375 | |
| Total | | | | $84.375 |

Tabla 4 - Recursos Teconológicos Depreciación

Nota:

* + Cabe destacar la depreciación del equipo a utilizar en el desarrollo de este proyecto es equivalente al 30% y nuestro proyecto se realizará en aproximadamente 2.5 meses, donde el total del valor de los equipos ($1.350.000), se debe multiplicar por el 30%, luego se debe dividir en 12 meses que es equivalente a un 1 año y el resultado se multiplica por los meses de uso del equipo que son 2.5 meses y esto nos da una depreciación total de los 3 equipos equivalente a $84.375.

**Recursos de Operación**

* + **Energía Eléctrica**

El cálculo de la energía eléctrica se realiza tomando en cuenta el consumo de energía utilizada por los equipos para el desarrollo del proyecto, es decir, que la cantidad de energía consumida por cada uno de los equipos dependerá de las características propias de cada uno de ellos.

* + La siguiente tabla muestra un resumen de la energía aproximada que será utilizada en el desarrollo de este proyecto.

|  |  |  |  |
| --- | --- | --- | --- |
| Equipo | Cantidad | Consumo (watts) | Consumo Total (watts) |
| Notebook | 3 | 90 | 270 |
| Iluminación Ampolletas (Estación de trabajo) | 3 | 15 | 45 |
| Total |  |  | 315 |

Tabla 5 - Recursos de Operación

Convirtiendo a kilowatts: (315 w/h)/ 1000 = 0,315 kw/h

Consumo Mensual días hábiles: 0,315 x 4 Horas x 5 días x 4 semanas = 25,2 kw/h aproximados que se consumen al mes.

Consumo Mensual fin de semana: 0,315 x 8 Horas x 2 días x 4 semanas = 20,16 kw/h aproximados que se consumen al mes.

El total del consumo mensual aproximado es de 45,36 kw/h al mes.

El costo del kilowatts por hora del servicio de energía eléctrica vigente hasta la fecha para un consumo mensual menor a 220 kw/h es de $86,964.

Así: 45,36 kw/h \* $86,964 = $3944,68 de consumo mensual.

El total del consumo de energía aproximado en el proyecto es de: $3944,68 de consumo mensual \* 2.5 meses = $9861,7.

Finalmente en el estudio de factibilidad se presenta un cuadro resumen de los costos de desarrollo del proyecto en el que se consideraron los costos de Recurso Humano, Recursos Consumibles y Recursos Tecnológicos.

|  |  |
| --- | --- |
| Recurso | Total ($) |
| Recurso Humano | $3.350.000 |
| Recurso Consumible | $53.460 |
| Recursos Tecnológicos | $1.368.900 |
| Recursos de Operación | $9.862 |
| Total | $4.782.222 |

Tabla 6 - Resumen del presupuesto del proyecto

Como se observa en la tabla anterior, el costo del desarrollo del proyecto asciende a la cantidad de $4.782.222

A continuación se incorporan las alternativas de solución al presupuesto general del proyecto detallado anteriormente.

**Alternativa N° 1 .net con MySQL**

|  |  |  |
| --- | --- | --- |
| Detalle equipamiento | Cantidad | Costo $ |
| Servidor Procesador I3 2.7 GHZ, RAM 8 Gb DDR3, HD 2,0 TB | 1 | 500.000 |
| Licencia SQL Server | 1 | 163.000 |
| Antivirus para Servidor | 1 | 203.000 |
| Licencia Windows Server 2008 r2 | 1 | 522.000 |
| Licencia Visual Studio | 1 | 253.000 |
| Total | | $1.641.000 |

|  |  |
| --- | --- |
| Recurso (Desarrollo) | Total ($) |
| Recurso Humano | $3.350.000 |
| Recurso Consumible | $53.460 |
| Recursos Tecnológicos | $1.368.900 |
| Recursos de Operación | $9.862 |
| Total | $4.782.222 |

El costo total de la alternativa N° 1 incluyendo Equipamiento y desarrollo, esto es un total de $6.423.222.

**Alternativa N° 2 PHP con MySQL**

|  |  |  |
| --- | --- | --- |
| Detalle | Cantidad | Costo $ |
| Servidor Procesador I5 2.7 GHZ, RAM 8 Gb DDR3, HD 2,0 TB | 1 | 500.000 |
| Licencia MySQL | 1 | 153.000 |
| Antivirus para Servidor | 1 | 203.000 |
| Licencia Windows Server 2008 r2 | 1 | 522.000 |
| Total | | $1.378.000 |

|  |  |
| --- | --- |
| Recurso (Desarrollo) | Total ($) |
| Recurso Humano | $3.350.000 |
| Recurso Consumible | $53.460 |
| Recursos Tecnológicos | $1.368.900 |
| Recursos de Operación | $9.862 |
| Total | $4.782.222 |

El costo total de la alternativa N° 2 incluyendo Equipamiento y desarrollo, esto es un total de $ 6.160.222.

**Costos del Proyecto**

Se deben analizar los costos del proyecto con el fin de obtener un monto total de la inversión, para poder visualizar si el proyecto es rentable para la contraparte.

La institución debe mantener una comunicación constante con sus clientes y en la actualidad sólo existe un sistema de ticket que genera principalmente una limitación en la comunicación y gestión entre la empresa y sus clientes.

* Este sistema pretende obtener principalmente los siguientes beneficios:
* Tener una mayor organización con la información de los proyectos.
* Mantener registros históricos de las solicitudes del cliente estableciendo, tipos y estados en las solicitudes.
* Representar la información estadísticamente para ayudar a la toma de decisiones.

El cuadro que se aprecia a continuación son los recursos necesarios para llevar a cabo el desarrollo a implementar:

|  |  |  |  |
| --- | --- | --- | --- |
| Descripción | Cantidad | Costo Unitario ($) | Total ($) |
| Licencia MySQL | 1 | - | - |
| Antivirus para Servidor | 1 | - | - |
| Licencia Windows Server 2008 r2 | 1 | - | - |
| Servidor (web y aplicaciones) | 1 | - | - |
| Costo realización del proyecto | - | $4.782.222 | $4.782.222 |
| Total Costo del Proyecto | | | $4.782.222 |

Tabla 7 - Costos del proyecto

La contraparte del sistema a implementar, ya cuenta con varios de los bienes a utilizar, los cuales se detallarán a continuación:

* Hosting
* Servidor Web y aplicaciones
* Licencia MySQL
* Antivirus para servidor
* Licencia Windows server 2008 r2

Como hoy en día existen varios sistemas que realizan procesos similares, pero no cumplen en su totalidad con lo solicitado por la contraparte. Los costos de éstos son altamente elevados y no satisfacen las necesidades de la empresa. Nuestro proyecto es ampliamente favorable sobre las prioridades de la institución, por lo tanto, consideramos que este proyecto es factiblemente económico y generará grandes beneficios.

#### 4.1.2.3 Factibilidad Operacional

En la ejecución del proyecto, consideramos necesario el siguiente personal:

* 1 Jefe de Proyecto
* 1 Testing
* 2 Programadores

Para operar el sistema, se requiere capacitar a todo el personal que dará uso del sistema, de esta forma la utilización de éste será más óptima como también sea administrado sin ningún inconveniente a la hora de operar.

En este proyecto se interactua directamente con la contraparte. Esto nos beneficia a ambos al momento de realizar la gestión de ésta, como también, administrar las solicitudes de los clientes, organizando la información, mejorando ampliamente el sistema actual de la empresa.

Al tener una buena comunicación y apoyo con la comunidad empresarial favorece al proyecto, ya que la toma de requerimientos se realiza de forma óptima. Los antecedentes y problemáticas que surgen en la empresa son vistas y analizadas ampliamente por la abundancia de información entregada por parte de la empresa y sus trabajadores. De esta forma se generan más alternativas de solución acorde a lo que la empresa realmente necesita.

De esta forma el sistema propuesto no causará perjuicios ni producirá resultados pobres para alguna área, será de fácil acceso para los usuarios del sistema, mejora ampliamente la productividad fortaleciendo la toma de decisiones.

Se deben considerar 3 aspectos fundamentales para el uso de su potencial:

**Complejidad del sistema**

Principalmente la idea principal es desarrollar un sistema con una interfaz amigable que sea fácil y sencilla de usar.

**Resistencia al cambio**

Hoy en día los usuarios se resisten al cambio, es decir, no son tolerantes a usar sistemas nuevos prefieren seguir usando sistemas antiguos o seguir trabajando manualmente (planillas, Excel, hojas, etc).

**Adaptación**

Para que los usuarios se adapten fácilmente al sistema se crearán manuales de usuarios y también se les capacitará a todo el personal lo cual tendremos total disposición en resolver sus preguntas o dudas con respecto al sistema.

#### 4.1.2.4 Comparativa de las Alternativas

A continuación procederemos a comparar las alternativas de solución para el desarrollo del proyecto de esta manera, poder apreciar cuál es la opción más fiable, óptima y eficiente para llevar a cabo el proyecto.

**Alternativa N° 1 .NET con MySQL**

La plataforma .NET es una de las alternativas que se quiere utilizar, ya que facilita el proceso de construcción de programas multipropósito. Esta herramienta consta con una gran cantidad de tecnologías que facilitan las tareas y a la vez agilizan la implementación. Es una alternativa eficiente.

Cabe mencionar que el desarrollo del proyecto con esta alternativa genera las siguientes ventajas y desventajas.

**Ventajas:**

* Aplicaciones con componentes en múltiples lenguajes.
* Ofrece mucha documentación de ayuda (herramientas, debuggers, editores).
* Rendimiento: Todos los códigos son compilados.
* Rápido aprendizaje por parte de los desarrolladores.
* Movilidad: Las aplicaciones pueden ser desplegadas en una amplia variedad de dispositivos.
* Escalabilidad y flexibilidad.
* Seguridad: .NET da respaldo para ejecutar código no seguro.

**Desventajas:**

* Mantenimiento en múltiples lenguajes, mantener un proyecto en múltiples lenguajes es costoso. Si una aplicación está realizada en varios lenguajes se necesitan expertos en varios lenguajes para entenderla y mantenerla, aumentando los costos.
* .NET no es multiplataforma. La plataforma .NET, sólo está disponible para la familia Windows.
* Licencias: Es un código cerrado, no hay licencias libres. La infraestructura para desarrollar en .NET, representa un alto costo para las empresas.

Los costos de equipamiento y desarrollo para esta alternativa son los siguientes:

|  |  |
| --- | --- |
| Detalle | Costo ($) |
| Desarrollo del proyecto | $4.782.222 |

|  |  |
| --- | --- |
| Detalle | Costo ($) |
| Equipamiento | $1.641.000 |

El costo total de la alternativa N° 1 incluyendo Equipamiento y desarrollo, el total de esto es de: $ $6.423.222.

**Alternativa N° 2 PHP con MySQL**

PHP es un lenguaje muy eficiente para el desarrollo de este proyecto, porque es completamente libre no requiere licencias para su utilización, es un lenguaje multiplataforma, compatible con la gran mayoría y principales gestores de base de datos.

El soporte para MySQL, el servidor y el hosting lo otorga la contraparte, por lo que los gastos de implementación se mantienen y no alteran los valores estipulados desde un principio en la factibilidad económica.

Cabe mencionar que el desarrollo del proyecto con esta alternativa genera las siguientes ventajas y desventajas.

**Ventajas:**

* Fácil de aprender.
* Lenguaje muy rápido.
* Lenguaje multiplataforma.
* Capacidad de conexión con la mayoría de los manejadores de bases de datos.
* Capacidad de expandir su potencial utilizando módulos.
* Es libre, no requiere licencias para la utilización.
* Incluye gran cantidad de funciones.

**Desventajas:**

* La principal desventaja es la necesidad de tener un servidor web.

Los costos de equipamiento y desarrollo para esta alternativa, son los siguientes:

|  |  |
| --- | --- |
| Detalle | Costo ($) |
| Desarrollo del proyecto | $4.782.222 |

|  |  |
| --- | --- |
| Detalle | Costo $ |
| Equipamiento | $1.378.000 |

El costo total de la alternativa N° 2 incluyendo Equipamiento y desarrollo, esto es un total de: $ 6.160.222.

Como la empresa Lazos S.A. cuenta con el equipamiento la inversión total sería sólo la contemplada en el desarrollo con un monto total de: $4.782.222.

### 4.1.3 Selección de Alternativa de Solución

Según los antecedentes, conocimientos y ventajas otorgadas hacía el proyecto en la implementación y desarrollo, hemos concluido que la alternativa de solución más acorde a los requerimientos establecidos es PHP con MySQL, ya que cumple en su totalidad en lo solicitado por la contraparte, generando ventajas significativas económicas y operacionales para ésta.

Como es un lenguaje libre, no necesita licencias para su desarrollo y utilización, es multiplataforma, por lo tanto, no habrán problemas de incompatibilidades al momento de utilizar el sistema. También es compatible con la gran mayoría y principales gestores de base de datos.

Económicamente es rentable utilizar esta alternativa porque disminuye significativamente el presupuesto. El rubro de la empresa Lazos S.A. está situado al desarrollo de sistemas, por lo que ellos cuentan con el soporte para MySQL, como también cuentan con servidor web y el hosting, por lo que los gastos de implementación se mantienen y no alteran los valores estipulados desde un principio en la factibilidad económica.

Por otra parte la experiencia que existe por parte del equipo hacia el lenguaje es intermedia, la cual no genera grande dificultades para el desarrollo del proyecto.

#### 4.1.3.1 Beneficios de la Solución

Como ya se ha mencionado en la alternativa N°2, ésta otorga beneficios significativos para la empresa, a continuación se detallan las alternativas:

* La inversión total del proyecto es menor ya que la empresa Lazos S.A. Cuenta en su totalidad con el equipamiento necesario para la implementación del sistema.
* Utilizar PHP con MySQL genera ventajas hacia nuestro equipo de trabajo, porque no se necesitará una gran capacitación para la utilización de este lenguaje.
* PHP, no requiere licencias para ser utilizado. Existe gran cantidad de información en la Web y foros que entregan ayuda.

## Planificación de la solución (Carta Gantt)



Ilustración 6: Gantt, Parte 1



Ilustración 7: Gantt, Parte 2



Ilustración 8: Gantt, Parte 3



Ilustración 9: Gantt, Parte 4

## Desarrollo del Requerimiento

### 4.3.1 Definición de Requerimientos

Tras una ardua revisión con varias interaciones con el cliente, se llego a determinar que la solución óptima para solucionar este problema era un sistema desarrollados a medidas, que sea capaz de suplir todas aquellas necesidades relacionadas con la parte de la post-entrega de sus proyectos terminados.

### 4.3.2 Análisis del Requerimiento

Posteriormente al punto anterior, se concluyó que se optará por un sistema robusto que satisfasga todas las necesidades del cliente para poder cumplir con el flujo de información de las solicitudes, las cuales son solicitudes posteriores a la implementación de sus sistemas, por lo tanto, se debe considerar lo siguiente:

* Se debe entregar el análisis de las solicitudes en base a estadísticas.
* Se debe gestionar el flujo de las solicitudes en el tiempo.
* Se debe gestionar usuarios para el manejo de las solicitudes.
* Debe existir un registro histórico de la gestión de las solicitudes.
* Se debe notificar vía correo electrónico a los interesados sobre el flujo de las solcitudes.
* Se debe centralizar la información en base a los proyectos a los que pertenezcan las solcitides.

#### 4.3.2.1 Requisitos Funcionales

Este sistema cumplirá una gama de requisitos funcionales los cuales se detallarán a continuación:

**REQ-F-01**: Gestión de Proyectos

Se debe poder crear proyectos, los cuales estarán asociados a los clientes que posea la empresa. Los proyectos se deben listar en una interfaz de búsqueda en la cual se pueda visualizar la información más relevante de éstos, y se debe poder realizar búsquedas con filtros básicos y también avanzados.

**REQ-F-02**: Gestión de Solicitudes

Se debe poder crear solicitudes, seleccionando un tipo, y un proyecto al cual pertenezcan. La solicitud debe quedar asignada siempre a un usuario durante su ciclo de vida. Los usuarios podrán avanzar la solicitud en el tiempo cambiando su estado, agregando observaciones, y agregando documentos.

**REQ-F-03**: Gestión de Usuarios

Se debe poder crear usuarios, los cuales tendrán acceso al sistema mediante una pantalla de login, y éstos podrán tener asignado uno o más perfiles de los existentes, estos determinan a que interfaces tendrán acceso y a los cuales no podrá acceder.

**REQ-F-04**: Alertas

Se debe envíar correos electrónicos informando el cambio de estado de las solicitudes a los usuarios involucrados en ésta.

**REQ-F-05**: Estadísticas

Se debe poder visualizar la información de los proyectos de manera estadística respecto a sus solicitudes, para lo cual se contabilizarán las cantidades de solicitudes y su estado para ciertos periodos de tiempo.

**REQ-F-06**: Gestión de Clientes

Se debe poder crear clientes, los cuales serán quienes tengan proyectos asociados y se deberán listar en los filtros de búsqueda relacionados.

#### 4.3.2.2 Requisitos No Funcionales

Este sistema cumplirá una gama de requisitos no funcionales los cuales se detallarán a continuación:

**REQ-NF-01**: Carga de datos inicial

Se debe realizar una carga de datos inicial antes de poner en marcha el sistema, para tener configurado los perfiles y el acceso a las interfaces correspondientes. Además se debe crear un super usuario que tenga asociado todos los perfiles.

**REQ-NF-02**: Registro de estadísticas

Se debe crear un proceso almacenado que tenga una periodicidad acorde a la información estadística que se guardará de las solicitudes de los proyectos.

**REQ-NF-03**: Usabilidad

Se debe considerar la usabilidad de las interfaces del sistema, de manera que sean de facil manejo y amigables.

**REQ-NF-04**: Seguridad

Se debe contemplar que para acceder al sistema se este ingresando con un usuario válido, y con el perfil indicado para acceder a cada interfaz. No se debe permitir el acceso sin el paso por el login a ningún componente del sistema.

**REQ-NF-05**: Mantenibilidad

El sistema debe considerar ser dinámico para la creación del menú de acceso a las interfaces y la asociación de los perfiles a los usuarios, de manera que se puedan integrar a futuro nuevos componentes en el sistema. Así también debe ser mantenible el modelo de datos utilizando integridad referencial.

## Diseño Lógico

A continuación se describe el diseño lógico de la aplicación, el cual consiste en representar las funcionalidades del sistema a través del lenguaje unificado de modelamiento (UML), utilizando casos de uso y diagramas de secuencia.

### 4.4.1 Modelo de Casos de Uso



Ilustración 10 - Casos de uso 1



Ilustración 11 - Casos de uso 2



Ilustración 12 - Casos de uso 3



Ilustración 13 - Casos de uso 4

### 4.4.2 Definición de Actores

* **Administrador:** El actor Administrador representa un perfil de la empresa que tiene un alto grado de responsabilidad, sus funcionalidades son la gestión de proyectos, clientes, y usuarios. También puede visualizar la información estadística.
* **Encargado:** El actor Encargado representa un perfil de la empresa que tiene una responsabilidad de gestión, como los jefes de proyecto y gerentes. Sus funcionalidades son la gestión de proyectos, y las solicitudes asociadas, para esto pueden visualizar la información estadística, asignar solicitudes a usuarios y modificar el estado de éstas para gestionar el flujo de trabajo.
* **Desarrollador:** El actor Desarrollador representa un perfil de la empresa que tiene una responsabilidad de operador, sus funcionalidades son revisar las solicitudes que tenga asignadas y avanzarlas en el flujo de trabajo agregando observaciones o documentos.

### 4.4.3 Definición de Casos de Uso

* **Inicia sesión:** Los usuarios deben primero iniciar sesión para ingresar al sistema.
* **Modifica datos personales:** Los usuarios pueden modificar su información básica.
* **Crea Proyectos:** Creación de proyectos en el sistema, asociados a un cliente.
* **Modifica Proyectos:** Modificación de la información de un proyecto.
* **Busca Proyectos:** Búsqueda de proyectos bajo determinados filtros.
* **Crea Usuarios:** Creación de usuarios en el sistema.
* **Asigna Perfiles:** Asignación de perfiles de acceso a los usuarios del sistema.
* **Asigna Solicitudes:** Asignación de solicitudes a los usuarios del sistema.
* **Crea Clientes:** Creación de clientes en el sistema.
* **Modifica Clientes:** Modificación de la información de un cliente.
* **Busca Clientes:** Búsqueda de clientes bajo determinados filtros.
* **Visualiza estadísticas:** Despliegue de la información estadística de las solicitudes de los proyectos.
* **Crea Solicitudes:** Creación de solicitudes en el sistema, asociadas a un tipo y un proyecto.
* **Busca Solicitudes:** Búsqueda de solicitudes bajo determinados filtros.
* **Cambia estado de Solicitudes:** Cambio de estado de las solicitudes asignadas.
* **Agrega Observaciones/Documentos a Solicitudes:** Agregación de observaciones y documentos al historial de las solicitudes.
* **Envía correo electrónico:** Notificación vía correo electrónico de los cambios de estado de las solicitudes.

### 4.4.4 Descripción de Casos de Uso

#### 4.4.4.1 Inicia sesión

|  |  |
| --- | --- |
| **Código:** | CU-01 |
| **Requisito:** | REQ-F-03, REQ-NF-04 |
| **Fecha:** | 10-11-2013 |
| **Actores:** | Administrador, Encargado, Desarrollador |
| **Precondiciones:** | N/A |
| **Flujo Normal:** | 1. El usuario ingresa su identificador de usuario y contraseña. 2. El usuario presiona el botón Iniciar sesión. 3. El sistema válida las credenciales del usuario. 4. El sistema inicia la sesión del usuario en el sistema y despliega la página inicial. 5. El usuario visualiza que ha ingresado al sistema. 6. Fin del caso de uso. |
| **Flujo Alternativo:** | * Flujo alternativo al paso 4:   + El sistema determina que las credenciales son inválidas.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 1. |
| **Poscondiciones:** | El usuario accede al sistema. |

Tabla 8 - CU-01

#### 4.4.4.2 Modifica datos personales

|  |  |
| --- | --- |
| **Código:** | CU-02 |
| **Requisito:** | REQ-F-03 |
| **Fecha:** | 10-11-2013 |
| **Actores:** | Administrador, Encargado, Desarrollador |
| **Precondiciones:** | Usuario registrado en el sistema. |
| **Flujo Normal:** | 1. El usuario ingresa a la opción de su información personal. 2. El sistema despliega los siguientes datos:    1. Rut (alfa-numérico)    2. Nombre (texto)    3. Apellido (texto)    4. Teléfono (numérico)    5. Dirección (texto)    6. Contraseña (alfa-numérico)    7. Correo (formato e-mail) 3. El usuario puede modificar los siguientes datos:    1. Nombre (Obligatorio)    2. Apellido (Obligatorio)    3. Teléfono (Opcional)    4. Dirección (Opcional)    5. Contraseña (Opcional)    6. Correo (Obligatorio) 4. El usuario presiona el botón Guardar. 5. El sistema válida los datos y muestra mensaje de éxito del proceso. 6. Fin caso de uso. |
| **Flujo Alternativo:** | * Flujo alternativo al paso 5:   + El sistema determina que los datos son inválidos.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 2. |
| **Poscondiciones:** | Se actualizan los datos del usuario. |

Tabla 9 - CU-02

#### 4.4.4.3 Crea Proyectos

|  |  |
| --- | --- |
| **Código:** | CU-03 |
| **Requisito:** | REQ-F-01 |
| **Fecha:** | 10-11-2013 |
| **Actores:** | Administrador, Encargado |
| **Precondiciones:** | N/A |
| **Flujo Normal:** | 1. El usuario ingresa a la opción para crear proyectos. 2. El sistema despliega los siguientes campos:    1. Nombre (texto)    2. Descripción (texto)    3. Jefe de proyecto (combo-box)    4. Duración (texto)    5. Fecha de inicio (fecha)    6. Fecha de término (fecha)    7. Fecha garantía (fecha)    8. Cliente (combo-box)    9. Tipo de proyecto (combo-box) 3. El usuario presiona el botón Crear. 4. El sistema válida los datos y muestra mensaje de éxito del proceso. 5. Fin caso de uso. |
| **Flujo Alternativo:** | * Flujo alternativo al paso 4:   + El sistema determina que los datos son inválidos.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 2. |
| **Poscondiciones:** | Se crea el proyecto en el sistema con estado vigente. |

Tabla 10 - CU-03

#### 4.4.4.4 Modifica Proyectos

|  |  |
| --- | --- |
| **Código:** | CU-04 |
| **Requisito:** | REQ-F-01 |
| **Fecha:** | 16-11-2013 |
| **Actores:** | Administrador, Encargado |
| **Precondiciones:** | N/A |
| **Flujo Normal:** | 1. El usuario ingresa a la opción de modificar proyecto. 2. El sistema despliega los siguientes datos:    1. Nombre (texto)    2. Descripción (texto)    3. Jefe de proyecto (combo-box)    4. Duración (texto)    5. Fecha de inicio (fecha)    6. Fecha de término (fecha)    7. Fecha garantía (fecha)    8. Cliente (combo-box)    9. Tipo de proyecto (combo-box) 3. El usuario puede modificar los siguientes datos:    1. Descripción (Obligatorio)    2. Jefe de proyecto (Obligatorio)    3. Duración (Obligatorio)    4. Fecha de inicio (Obligatorio)    5. Fecha de término (Obligatorio)    6. Fecha garantía (Obligatorio) 4. El usuario presiona el botón Guardar. 5. El sistema válida los datos y muestra mensaje de éxito del proceso. 6. Fin caso de uso. |
| **Flujo Alternativo:** | * Flujo alternativo al paso 5:   + El sistema determina que los datos son inválidos.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 2. * Flujo alternativo paso 4:   + El usuario presiona el botón Eliminar.   + El sistema despliega mensaje de pregunta de confirmación.   + El usuario confirma la acción.   + El sistema cambia el estado del proyecto a inactivo.   + El flujo continúa en el paso 5. |
| **Poscondiciones:** | Se actualizan los datos del proyecto. |

Tabla 11 - CU-04

#### 4.4.4.5 Busca Proyectos

|  |  |
| --- | --- |
| **Código:** | CU-05 |
| **Requisito:** | REQ-F-01 |
| **Fecha:** | 16-11-2013 |
| **Actores:** | Administrador, Encargado |
| **Precondiciones:** | N/A |
| **Flujo Normal:** | 1. El usuario ingresa a la opción Búsqueda de Proyecto. 2. El sistema despliega los siguientes filtros:    1. Nombre de Proyecto (texto)    2. Jefe de Proyecto (Combo-box)    3. Cliente (Combo-box)    4. Fecha Inicio desde (fecha) (filtros avanzados)    5. Fecha Inicio hasta (fecha) (filtros avanzados)    6. Fecha Término desde (fecha) (filtros avanzados)    7. Fecha Término hasta (fecha) (filtros avanzados)    8. Fecha de garantía desde (fecha) (filtros avanzados)    9. Fecha de garantía hasta (fecha) (filtros avanzados)    10. Destacado (combo-box) (filtros avanzados) 3. El usuario puede ingresar ingresar uno o más filtros. 4. El usuario presiona el botón Buscar. 5. El sistema válida los datos y despliega los resultados de la búsqueda. 6. El usuario visualiza los siguientes campos en la tabla de resultados:    1. Nombre de Proyecto (opcional)    2. Destacado    3. Tipo de Proyecto    4. Jefe de Proyecto    5. Fecha Inicio    6. Fecha Término    7. Fecha Garantía    8. Cliente    9. Link al modificar 7. Fin caso de uso. |
| **Flujo Alternativo:** | * Flujo alternativo al paso 7:   + El usuario presiona el botón limpiar.   + El sistema limpia los filtros y los resultados de la búsqueda.   + El flujo continúa en el paso 3. * Flujo alternativo al paso 7:   + El usuario presiona el link al modificar.   + El sistema direcciona al caso de uso Modifica Proyecto. * Flujo alternativo al paso 5:   + El sistema determina que los datos son inválidos.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 3. * Flujo alternativo al paso 6:   + No se encuentra resultados con la búsqueda.   + El sistema despliega mensaje de que no se han encontrado resultado.   + El flujo continúa en el paso 3. |
| **Poscondiciones:** | N/A |

Tabla 12 - CU-05

#### 4.4.4.6 Crea Usuarios

|  |  |
| --- | --- |
| **Código:** | CU-06 |
| **Requisito:** | REQ-F-03 |
| **Fecha:** | 16-11-2013 |
| **Actores:** | Administrador |
| **Precondiciones:** | N/A |
| **Flujo Normal:** | 1. El usuario ingresa a la opción para crear proyectos. 2. El sistema despliega los siguientes campos:    1. Nombre (texto)    2. Apellido (texto)    3. Rut (texto)    4. Descripción (texto)    5. Telefóno (numérico)    6. Dirección (texto)    7. Correo (formato e-mail)    8. Contraseña (texto) 3. El usuario presiona el botón Crear Usuario. 4. El sistema válida los datos y muestra mensaje de éxito del proceso. 5. Fin caso de uso. |
| **Flujo Alternativo:** | * Flujo alternativo al paso 4:   + El sistema determina que los datos son inválidos.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 2. |
| **Poscondiciones:** | Se crea el Usuario en el sistema con estado vigente. |

Tabla 13 - CU-06

#### 4.4.4.7 Asigna Perfiles

|  |  |
| --- | --- |
| **Código:** | CU-07 |
| **Requisito:** | REQ-F-03 |
| **Fecha:** | 16-11-2013 |
| **Actores:** | Administrador |
| **Precondiciones:** | N/A |
| **Flujo Normal:** | 1. El usuario ingresa a la opción para Asignar perfiles. 2. El sistema despliega los siguientes campos:    1. Rut (Combo-box)    2. Nombre Usuario (Combo-box)    3. Fecha de creación (fecha) 3. El usuario presiona el botón Buscar. 4. El sistema válida los datos y despliega los resultados de la búsqueda. 5. El usuario visualiza los siguientes campos en la tabla resultado:    1. Rut    2. Nombre    3. Correo    4. Fecha de creación    5. Link de perfiles 6. El usuario presiona el link a los perfiles para un usuario. 7. El sistema visualiza los perfiles disponibles y los asignados al usuario. 8. El usuario modifica la asignación de los perfiles y presiona el botón guardar. 9. El sistema válida los datos y muestra mensaje de éxito del proceso. 10. Find del caso de uso. |
| **Flujo Alternativo:** | * Flujo alternativo al paso 4:   + El sistema determina que los datos son inválidos.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 2. * Flujo alternativo al paso 9:   + El sistema determina que los datos son inválidos.   + El sistema despliega un mensaje de errror.   + El flujo continúa en el paso 8. |
| **Poscondiciones:** | Se actualiza en el sistema la relación de los perfiles asociados al usuario. |

Tabla 14 - CU-07

#### 4.4.4.8 Asigna Solicitudes

|  |  |
| --- | --- |
| **Código:** | CU-08 |
| **Requisito:** | REQ-F-02 |
| **Fecha:** | 16-11-2013 |
| **Actores:** | Administrador |
| **Precondiciones:** | N/A |
| **Flujo Normal:** | 1. Si el usuario es Encargado, la solicitud queda asignada al usuario seleccionado en el caso de uso Cambia estado de solicitudes. 2. Si el usuario es desarrollador, la solicitud queda asignada al usuario creador de la solicitud. 3. Fin caso de uso. |
| **Flujo Alternativo:** | N/A |
| **Poscondiciones:** | La solicitud queda asignada al usuario. |

Tabla 15 - CU-08

#### 4.4.4.9 Crea Clientes

|  |  |
| --- | --- |
| **Código:** | CU-09 |
| **Requisito:** | REQ-F-06 |
| **Fecha:** | 10-11-2013 |
| **Actores:** | Administrador, Encargado, Desarrollador |
| **Precondiciones:** | N/A |
| **Flujo Normal:** | 1. El usuario ingresa a la opción para crear Clientes. 2. El sistema despliega los siguientes campos:    1. Nombre (texto)    2. Apellido (texto)    3. Correo (formato e-mail)    4. Empresa (texto)    5. Rut empresa (formato rut)    6. Dirección (texto) 3. El usuario presiona el botón Crear. 4. El sistema válida los datos y muestra mensaje de éxito del proceso. 5. Fin caso de uso. |
| **Flujo Alternativo:** | * Flujo alternativo al paso 4:   + El sistema determina que los datos son inválidos.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 2. |
| **Poscondiciones:** | Se crea el Cliente en el sistema con estado vigente. |

Tabla 16 - CU-09

#### 4.4.4.10 Modifica Clientes

|  |  |
| --- | --- |
| **Código:** | CU-10 |
| **Requisito:** | REQ-F-06 |
| **Fecha:** | 10-11-2013 |
| **Actores:** | Administrador |
| **Precondiciones:** | Crea Clientes |
| **Flujo Normal:** | 1. El usuario ingresa a la opción de modificar Cliente. 2. El sistema despliega los siguientes datos:    1. Nombre (texto)    2. Apellido (texto)    3. Correo (formato e-mail)    4. Empresa (texto)    5. Rut empresa (formato Rut)    6. Dirección (texto) 3. El usuario puede modificar los siguientes datos:    1. Nombre (Obligatorio)    2. Apellido (Obligatorio)    3. Correo (Obligatorio)    4. Empresa (Obligatorio)    5. Rut empresa (Obligatorio)    6. Dirección (Opcional) 4. El usuario presiona el botón Guardar. 5. El sistema válida los datos y muestra mensaje de éxito del proceso. 6. Fin caso de uso. |
| **Flujo Alternativo:** | * Flujo alternativo al paso 5:   + El sistema determina que los datos son inválidos.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 2. * Flujo alternativo al paso 4:   + El usuario presiona el botón Eliminar.   + El sistema despliega mensaje de pregunta de confirmación.   + El usuario confirma la acción.   + El sistema cambia el estado del cliente a inactivo.   + El flujo continúa en el paso 5. |
| **Poscondiciones:** | Se actualizan los datos del cliente. |

Tabla 17 - CU-10

#### 4.4.4.11 Busca Clientes

|  |  |
| --- | --- |
| **Código:** | CU-11 |
| **Requisito:** | REQ-F-06 |
| **Fecha:** | 16-11-2013 |
| **Actores:** | Administrador, Encargado |
| **Precondiciones:** | N/A |
| **Flujo Normal:** | 1. El usuario ingresa a la opción Búsqueda de Clientes. 2. El sistema despliega los siguientes filtros:    1. Nombre (texto)    2. Apellido (texto)    3. Empresa (texto)    4. Rut empresa (texto) 3. El usuario puede ingresar ingresar uno o más filtros. 4. El usuario presiona el botón Buscar. 5. El sistema válida los datos y despliega los resultados de la búsqueda. 6. El usuario visualiza los siguientes campos en la tabla de resultados:    1. Nombre    2. Apellido    3. Empresa    4. Rut empresa    5. Link al modificar (sólo Administrador)   Fin caso de uso. |
| **Flujo Alternativo:** | * Flujo alternativo al paso 7:   + El usuario presiona el botón limpiar.   + El sistema limpia los filtros y los resultados de la búsqueda.   + El flujo continúa en el paso 3. * Flujo alternativo al paso 7:   + El usuario presiona el link al modificar.   + El sistema direcciona al caso de uso Modifica Clientes. * Flujo alternativo al paso 5:   + El sistema determina que los datos son inválidos.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 3. |
| **Poscondiciones:** | N/A |

Tabla 18 - CU-11

#### 4.4.4.12 Visualiza estadísticas

|  |  |
| --- | --- |
| **Código:** | CU-12 |
| **Requisito:** | REQ-F-05, REQ-NF-02 |
| **Fecha:** | 16-11-2013 |
| **Actores:** | Administrador, Encargado |
| **Precondiciones:** | N/A |
| **Flujo Normal:** | 1. El usuario ingresa a la opción Estadísticas. 2. El sistema despliega los siguientes filtros:    1. Nombre de Proyecto (texto)    2. Tipo de Proyecto (combo-box)    3. Fecha de Inicio (fecha)    4. Fecha de Término (fecha)    5. Tipo de Solicitud (combo-box)    6. Estado Solicitud (combo-box)    7. Cliente (texto) 3. El usuario puede ingresar uno o más filtros. 4. El usuario presiona el botón Buscar. 5. El sistema válida los datos y despliega los resultados de la búsqueda. 6. El usuario visualiza la información estadística de acuerdo a los filtros seleccionados 7. Fin caso de uso. |
| **Flujo Alternativo:** | * Flujo alternativo al paso 3:   + El usuario presiona el botón limpiar.   + El sistema limpia los filtros y los resultados de la búsqueda.   + El flujo continúa en el paso 2. * Flujo alternativo al paso 4:   + El sistema determina que los datos son inválidos.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 2. * Flujo alternativo al paso 6:   + No se encuentra resultados con la búsqueda.   + El sistema despliega mensaje de que no se han encontrado resultados.   + El flujo continúa en el paso 3. |
| **Poscondiciones:** | N/A |

Tabla 19 - CU-12

#### 4.4.4.13 Crea Solicitudes

|  |  |
| --- | --- |
| **Código:** | CU-13 |
| **Requisito:** | REQ-F-02 |
| **Fecha:** | 10-11-2013 |
| **Actores:** | Encargado |
| **Precondiciones:** | N/A |
| **Flujo Normal:** | 1. El usuario ingresa a la opción para crear Solicitudes. 2. El sistema despliega los siguientes campos:    1. Nombre (texto)    2. Descripción (texto)    3. Proyecto (combo-box)    4. Prioridad (combo-box)    5. Categoría (combo-box)    6. Correo en copia (texto) 3. El usuario presiona el botón Crear. 4. El sistema válida los datos y muestra mensaje de éxito del proceso. 5. El sistema habilita la opción para asignar la solicitud. 6. Fin caso de uso. |
| **Flujo Alternativo:** | * Flujo alternativo al paso 4:   + El sistema determina que los datos son inválidos.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 2. |
| **Poscondiciones:** | La solicitud queda persistente en el sistema en estado creada. Se gatilla el caso de uso Envía correo electrónico. |

Tabla 20 - CU-13

#### 4.4.4.14 Busca Solicitudes

|  |  |
| --- | --- |
| **Código:** | CU-14 |
| **Requisito:** | REQ-F-02 |
| **Fecha:** | 10-11-2013 |
| **Actores:** | Administrador, Encargado, Desarrollador |
| **Precondiciones:** | N/A |
| **Flujo Normal:** | 1. El usuario ingresa a la opción Búsqueda de Solicitudes. 2. El sistema despliega los siguientes filtros:    1. Código (numérico)    2. Nombre (texto)    3. Estado (combo-box)    4. Proyecto (combo-box)    5. Categoría (combo-box)    6. Fecha creación desde (fecha) (filtro avanzado)    7. Fecha creación hasta (fecha) (filtro avanzado)    8. Usuario (combo-box) (filtro avanzado)    9. Destacado (combo-box) (filtro avanzado)    10. Cliente (combo-box) (filtro avanzado) 3. El usuario puede ingresar ingresar uno o más filtros. 4. El usuario presiona el botón Buscar. 5. El sistema válida los datos y despliega los resultados de la búsqueda. 6. El usuario visualiza los siguientes campos en la tabla de resultados:    1. Destacado    2. Código    3. Nombre    4. Estado    5. Proyecto    6. Categoría    7. Link al detalle 7. El usuario puede presionar el campo Destacado. 8. El sistema intercambia el estado del campo Destacado para la solicitud respectiva. 9. El usuario presionar el link al detalle. 10. El sistema direcciona al detalle de la solicitud. 11. Fin caso de uso. |
| **Flujo Alternativo:** | * Flujo alternativo al paso 11:   + El usuario presiona el botón limpiar.   + El sistema limpia los filtros y los resultados de la búsqueda.   + El flujo continúa en el paso 3. * Flujo alternativo al paso 4:   + El usuario presiona el botón Mis solicitudes.   + El sistema realiza la búsqueda considerando las que solicitudes asignadas al usuario.   + El flujo continúa en el paso 6. * Flujo alternativo al paso 5:   + El sistema determina que los datos son inválidos.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 3. * Flujo alternativo al paso 6:   + No se encuentra resultados con la búsqueda.   + El sistema despliega mensaje de que no se han encontrado resultado.   + El flujo continúa en el paso 3. |
| **Poscondiciones:** | El usuario accede al detalle de la solicitud. |

Tabla 21 - CU-14

#### 4.4.4.15 Cambia estado de Solicitudes

|  |  |
| --- | --- |
| **Código:** | CU-15 |
| **Requisito:** | REQ-F-02 |
| **Fecha:** | 10-11-2013 |
| **Actores:** | Encargado, Desarrollador |
| **Precondiciones:** | Busca Solicitudes. |
| **Flujo Normal:** | 1. El usuario accede al detalle de la solicitud. 2. El sistema despliega la siguiente información:    1. Código    2. Nombre    3. Descripción    4. Estado    5. Proyecto    6. Categoría    7. Fecha creación    8. Usuario asignado 3. El usuario puede presionar el botón ver Histórico. 4. El sistema despliega el listado histórico de la solicitud con los siguientes campos:    1. Nombre    2. Descripción    3. Estado    4. Usuario    5. Prioridad    6. Categoría    7. Proyecto 5. El usuario puede presionar el botón ver Observaciones/Documentos. 6. El sistema despliega el listado de las Obseraciones/Documentos con los siguientes campos:    1. Usuario    2. Fecha    3. Comentario    4. Nombre archivo    5. Link descargar 7. El usuario puede presionar el Link descargar. 8. El sistema descarga el archivo. 9. El usuario puede modificar los siguientes datos:    1. Nombre (solo Encargado)    2. Descripción (solo Encargado)    3. Estado    4. Usuario (solo Encargado)    5. Prioridad (solo Encargado)    6. Categoria (solo Encargado)    7. Proyecto (solo Encargado)    8. Correo en copia (solo Encargado, oculto para Desarrollador) 10. El usuario presiona el botón Guardar. 11. El sistema válida los datos y muestra mensaje de éxito del proceso. 12. Fin caso de uso. |
| **Flujo Alternativo:** | * Flujo alternativo al paso 11:   + El sistema determina que los datos son inválidos.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 9. * Flujo alternativo al paso 11:   + El usuario a cambiado el dato Estado.   + El sistema determina que no se ha ejecutado el caso de uso Agrega Observaciones/Documentos.   + El sistema despliega mensaje de error de que es obligatorio agregar observación para cambiar estado.   + El flujo continúa en el paso 9. |
| **Poscondiciones:** | Se guarda el histórico y los datos de la solicitud en el sistema. |

Tabla 22 - CU-15

#### 4.4.4.16 Agrega Observaciones/Documentos a Solicitudes

|  |  |
| --- | --- |
| **Código:** | CU-16 |
| **Requisito:** | REQ-F-02 |
| **Fecha:** | 10-11-2013 |
| **Actores:** | Encargado, Desarrollador |
| **Precondiciones:** | Cambia estado de Solicitudes |
| **Flujo Normal:** | 1. El usuario ingresa al detalle de una solicitud. 2. El usuario ingresa texto en el campo observaciones. 3. El usuario puede adjuntar un documento presionando el botón seleccionar archivo. 4. El usuario presiona el botón para guardar los cambios. 5. El sistema válida los datos y muestra mensaje de éxito del proceso. 6. Fin caso de uso. |
| **Flujo Alternativo:** | * Flujo alternativo al paso 5:   + El sistema determina que los datos de la obseración son inválidos.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 2. * Flujo alternativo al paso 5:   + El sistema determina que los datos del documento exceden el tamaño máximo o no corresponden a los formatos permitidos.   + El sistema despliega mensaje de error.   + El flujo continúa en el paso 2. |
| **Poscondiciones:** | Las observaciones y documentos quedan persistentes en el sistema. |

Tabla 23 - CU-16

#### 4.4.4.17 Envía correo electrónico

|  |  |
| --- | --- |
| **Código:** | CU-17 |
| **Requisito:** | REQ-F-04 |
| **Fecha:** | 10-11-2013 |
| **Actores:** | Encargado, Desarrollador |
| **Precondiciones:** | Crea Solicitudes, Asigna Solicitudes, Cambia estado de Solicitudes. |
| **Flujo Normal:** | 1. El sistema busca los correos asociados a la solicitud. 2. El sistema envía el correo de alerta del cambio de estado de la solicitud. 3. Fin caso de uso. |
| **Flujo Alternativo:** | N/A |
| **Poscondiciones:** | N/A |

Tabla 24 - CU-17

### 4.4.5 Gráfico de Casos de Uso



Ilustración 14 - Gráfico de Casos de Uso

### 4.4.6 Diagrama de Secuencia



Ilustración 15 - Diagrama de secuencia 1



Ilustración 16 - Diagrama de secuencia 2



Ilustración 17 - Diagrama de secuencia 3



Ilustración 18 - Diagrama de secuencia 4



Ilustración 19 - Diagrama de secuencia 5



Ilustración 20 - Diagrama de secuencia 6



Ilustración 21 - Diagrama de secuencia 7



Ilustración 22 - Diagrama de secuencia 8



Ilustración 23 - Diagrama de secuencia 9



Ilustración 24 - Diagrama de secuencia 10



Ilustración 25 - Diagrama de secuencia 11



Ilustración 26 - Diagrama de secuencia 12



Ilustración 27 - Diagrama de secuencia 13



Ilustración 28 - Diagrama de secuencia 14



Ilustración 29 - Diagrama de secuencia 15

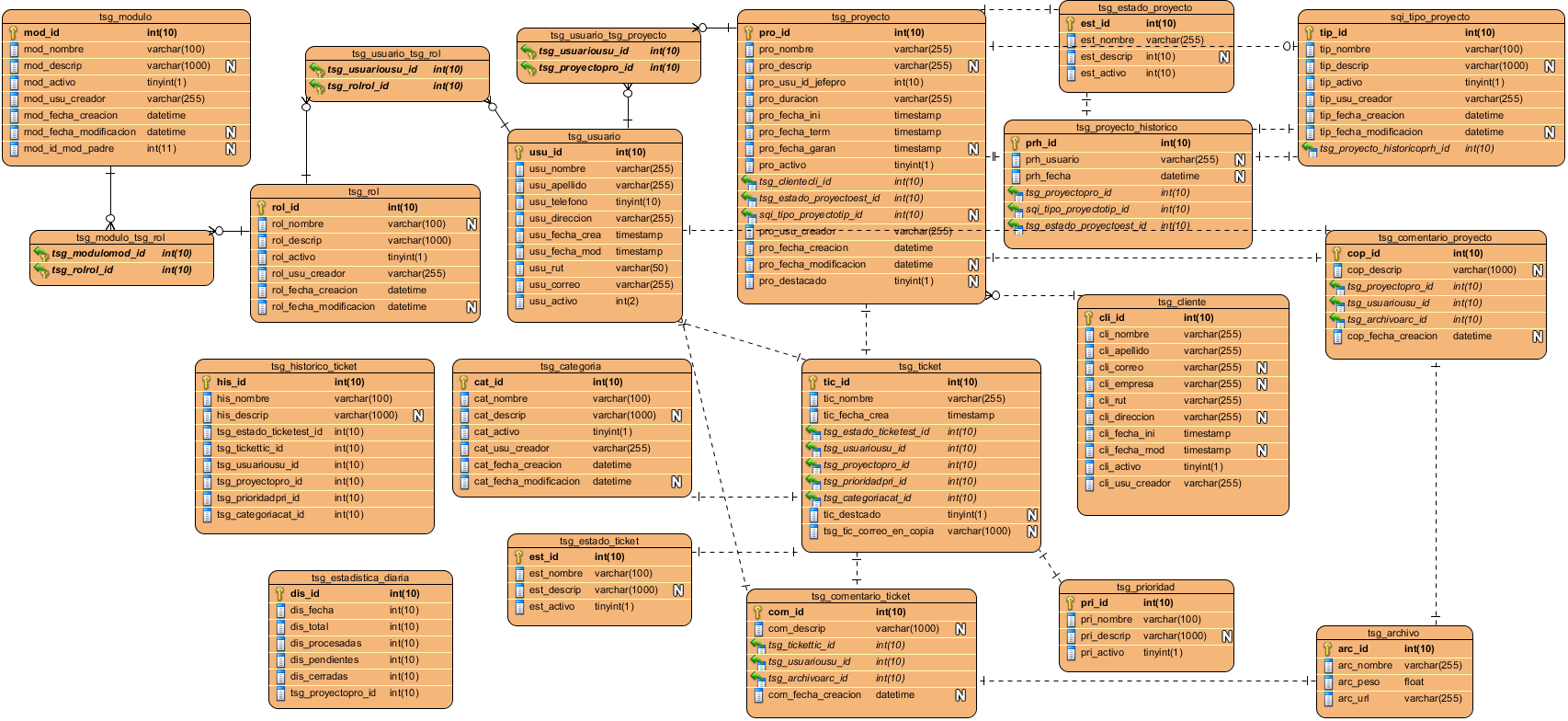
## Diseño Físico

### 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

#### 4.5.2.2.1 Resumen Tablas

|  |  |
| --- | --- |
| Nombre | Documentación |
| Image1.png [tsg\_estado\_proyecto](#.mEuioKGAqA8cQNj) | Tabla que corresponde al estado en que se encuentra el proyecto |
| Image1.png [tsg\_modulo](#3uSeioKGAqA8cQPh) | Tabla de módulos |
| Image1.png [tsg\_proyecto](#3PNOioKGAqA8cQM4) | Tabla de proyecto |
| Image1.png [tsg\_tipo\_proyecto](#ydHuioKGAqA8cQPA) | Tipo de proyecto |
| Image1.png [tsg\_usuario\_tsg\_proyecto](#SOORUYKGSugWDQUR) | Tabla de relación entre los usuarios y proyectos |
| Image1.png [tsg\_usuario\_tsg\_rol](#iFDVUYKGSugWDQe_) | Tabla de relación entre los usuarios y roles |
| Image1.png [tsg\_proyecto\_historico](#NZWXUYKGSugWDQkX) | Tabla de histórico del proyecto |
| Image1.png [tsg\_usuario](#Zmn2ioKGAqA8cQMi) | Tabla de Usuarios |
| Image1.png [tsg\_rol](#BkYeioKGAqA8cQPO) | Tabla de roles |
| Image1.png [tsg\_modulo\_tsg\_rol](#7D7VUYKGSugWDQfh) | Tabla de relación entre los módulos y roles |
| Image1.png [tsg\_comentario\_proyecto](#Fd6m0YKGSugWDQx4) | Tabla de observaciones del proyecto |
| Image1.png [tsg\_cliente](#qm3OioKGAqA8cQNZ) | Tabla sistema de gestión, esta tabla contiene los datos del cliente que interactuarán con el sistema |
| Image1.png [tsg\_historico\_ticket](#gE0eioKGAqA8cQPV) | Tabla de registro histórico del ticket |
| Image1.png [tsg\_categoria](#SMPuioKGAqA8cQPF) | Tabla de las categorías del proyecto |
| Image1.png [tsg\_ticket](#b2DOioKGAqA8cQNQ) | Tabla ticket, corresponde a los tickets que tiene asociado el proyecto |
| Image1.png [tsg\_estado\_ticket](#qVyuioKGAqA8cQN5) | Tabla del estado del ticket |
| Image1.png [tsg\_estadistica\_diaria](#RUueioKGAqA8cQP4) | Tabla de registro estadístico diario |
| Image1.png [tsg\_prioridad](#aBduioKGAqA8cQOz) | Tabla de prioridad del ticket |
| Image1.png [tsg\_comentario\_ticket](#k9RuioKGAqA8cQOM) | Tabla de comentario asociado al ticket |
| Image1.png [tsg\_archivo](#iXkASoKGAqA8cQWo) | Tabla de archivos adjuntos |

#### 4.5.2.2.2 Detalle de las Tablas

##### tsg\_estado\_proyecto

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | Tabla que corresponde al estado que se encuentra el proyecto |
| Autor | Tracy |
| Fecha de creación | 24-09-2013 03:31:33 PM |
| Fecha de modificación | 05-10-2013 10:43:40 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| est\_id | int(10) | PK | No | identificador único del estado del proyecto |
| est\_nombre | varchar(255) | N/A | No | nombre del estado del proyecto |
| est\_descrip | int(10) | N/A | Si | descripción del estado del proyecto |
| est\_activo | int(10) | N/A | No | eliminación lógica del sistema |

##### tsg\_modulo

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | Tabla de módulos |
| Autor | Tracy |
| Fecha de creación | 24-09-2013 03:36:36 PM |
| Fecha de modificación | 26-10-2013 07:28:08 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| mod\_id | int(10) | PK | No | Identificador único de la tabla |
| mod\_nombre | varchar(100) | N/A | No | Nombre |
| mod\_descrip | varchar(1000) | N/A | Si | Descripción del rol |
| mod\_activo | tinyint(1) | N/A | No | Indicador de registro activo |
| mod\_usu\_creador | varchar(255) | N/A | No | Usuario creador |
| mod\_fecha\_creacion | datetime | N/A | No | Fecha de creación |
| mod\_fecha\_modificacion | datetime | N/A | Si | Fecha de modificación |
| mod\_id\_mod\_padre | int(11) | N/A | Si | - |

##### tsg\_proyecto

|  |  |
| --- | --- |
| Nombre | Valor |
| Autor | Tracy |
| Fecha de creación | 24-09-2013 03:29:40 PM |
| Fecha de modificación | 05-10-2013 10:43:40 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| pro\_id | int(10) | PK | No | Clave primaria del proyecto, identificador único |
| pro\_nombre | varchar(255) | N/A | No | nombre del proyecto |
| pro\_descrip | varchar(255) | N/A | Si | descripción del proyecto |
| pro\_usu\_id\_jefepro | int(10) | N/A | No | Identificador del jefe de proyecto |
| pro\_duracion | varchar(255) | N/A | No | duración del proyecto |
| pro\_fecha\_ini | Timestamp | N/A | No | Fecha inicio del proyecto |
| pro\_fecha\_term | Timestamp | N/A | No | fecha termino del proyecto |
| pro\_fecha\_garan | Timestamp | N/A | Si | fecha de garantía del proyecto |
| pro\_activo | tinyint(1) | N/A | No | eliminación lógica del sistema. |
| tsg\_clientecli\_id | int(10) | FK ([tsg\_cliente.cli\_id](#am3OioKGAqA8cQNa)) | No | - |
| tsg\_estado\_proyectoest\_id | int(10) | FK ([tsg\_estado\_proyecto.est\_id](#.mEuioKGAqA8cQNk)) | No | - |
| sqi\_tipo\_proyectotip\_id | int(10) | FK ([sqi\_tipo\_proyecto.tip\_id](#KdHuioKGAqA8cQPB)) | Si | - |
| pro\_usu\_creador | varchar(255) | N/A | No | Usuario creador |
| pro\_fecha\_creacion | Datetime | N/A | No | Fecha creación |
| pro\_fecha\_modificacion | Datetime | N/A | Si | Fecha de modificación |
| pro\_destacado | tinyint(1) | N/A | Si | Destacado (Estrellita) |

##### tsg\_tipo\_proyecto

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | Tipo de proyecto |
| Autor | Tracy |
| Fecha de creación | 24-09-2013 03:34:51 PM |
| Fecha de modificación | 05-10-2013 10:43:40 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| tip\_id | int(10) | PK | No | Identificador único |
| tip\_nombre | varchar(100) | N/A | No | Nombre del tipo de proyecto |
| tip\_descrip | varchar(1000) | N/A | Si | Descripción del tipo de proyecto |
| tip\_activo | tinyint(1) | N/A | No | Registro activo |
| tip\_usu\_creador | varchar(255) | N/A | No | Usuario creador |
| tip\_fecha\_creacion | datetime | N/A | No | Fecha de creación |
| tip\_fecha\_modificacion | datetime | N/A | Si | Fecha de modificación |
| tsg\_proyecto\_historicoprh\_id | int(10) | FK ([tsg\_proyecto\_historico.prh\_id](#NZWXUYKGSugWDQkY)) | No | - |

##### tsg\_usuario\_tsg\_proyecto

|  |  |
| --- | --- |
| Nombre | Valor |
| Autor | Tracy |
| Fecha de creación | 05-10-2013 05:33:07 PM |
| Fecha de modificación | 05-10-2013 05:41:26 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| tsg\_usuariousu\_id | int(10) | PK/FK ([tsg\_usuario.usu\_id](#1mn2ioKGAqA8cQMj)) | No | - |
| tsg\_proyectopro\_id | int(10) | PK/FK ([tsg\_proyecto.pro\_id](#PPNOioKGAqA8cQM5)) | No | - |

##### tsg\_usuario\_tsg\_rol

|  |  |
| --- | --- |
| Nombre | Valor |
| Autor | Tracy |
| Fecha de creación | 05-10-2013 06:09:27 PM |
| Fecha de modificación | 26-10-2013 07:28:08 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| tsg\_usuariousu\_id | int(10) | PK/FK ([tsg\_usuario.usu\_id](#1mn2ioKGAqA8cQMj)) | No | - |
| tsg\_rolrol\_id | int(10) | PK/FK ([tsg\_rol.rol\_id](#hkYeioKGAqA8cQPP)) | No | - |

##### tsg\_proyecto\_historico

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | Tabla de histórico del proyecto |
| Autor | Tracy |
| Fecha de creación | 05-10-2013 07:17:51 PM |
| Fecha de modificación | 05-10-2013 10:43:40 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| prh\_id | int(10) | PK | No | Identificador único de la tabla |
| prh\_usuario | varchar(255) | N/A | Si | Usuario creador |
| prh\_fecha | datetime | N/A | Si | Fecha de creación |
| tsg\_proyectopro\_id | int(10) | FK ([tsg\_proyecto.pro\_id](#PPNOioKGAqA8cQM5)) | No | - |
| sqi\_tipo\_proyectotip\_id | int(10) | FK ([sqi\_tipo\_proyecto.tip\_id](#KdHuioKGAqA8cQPB)) | No | - |
| tsg\_estado\_proyectoest\_id | int(10) | FK ([tsg\_estado\_proyecto.est\_id](#.mEuioKGAqA8cQNk)) | No | - |

##### tsg\_usuario

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | Tabla de Usuarios |
| Autor | Tracy |
| Fecha de creación | 24-09-2013 03:26:09 PM |
| Fecha de modificación | 29-10-2013 02:27:35 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| usu\_id | int(10) | PK | No | Identificador único de la tabla |
| usu\_nombre | varchar(255) | N/A | No | Nombre |
| usu\_apellido | varchar(255) | N/A | No | Apellido del usuario |
| usu\_telefono | tinyint(10) | N/A | No | teléfono del usuario |
| usu\_direccion | varchar(255) | N/A | No | Dirección del Usuario |
| usu\_fecha\_crea | timestamp | N/A | No | Fecha de creación, se usara la secuencia de timestamp para guardar la hora y la fecha. |
| usu\_fecha\_mod | timestamp | N/A | No | fecha de modificación, se guardará con la secuencia timestamp para guardar la hora y la fecha de la modificación. |
| usu\_rut | varchar(50) | N/A | No | se define el rut del usuario, se considera como varchar ya que contienen número y caracteres. |
| usu\_correo | varchar(255) | N/A | No | correo electrónico del usuario |
| usu\_activo | int(2) | N/A | No | se deja la tabla de usuario con la opción para saber si el usuario esta activo o inactivo, en la BD de mostrara oculto si se elimino por sistema. (1 o 0) |

##### tsg\_rol

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | Tabla de roles |
| Autor | Tracy |
| Fecha de creación | 24-09-2013 03:35:46 PM |
| Fecha de modificación | 05-10-2013 10:33:09 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| rol\_id | int(10) | PK | No | Identificador único de la tabla |
| rol\_nombre | varchar(100) | N/A | Si | Nombre del rol |
| rol\_descrip | varchar(1000) | N/A | No | Descripción del rol |
| rol\_activo | tinyint(1) | N/A | No | Indicador de registro activo |
| rol\_usu\_creador | varchar(255) | N/A | No | Usuario creador |
| rol\_fecha\_creacion | datetime | N/A | No | Fecha de creación |
| rol\_fecha\_modificacion | datetime | N/A | Si | Fecha modificación |

##### tsg\_modulo\_tsg\_rol

|  |  |
| --- | --- |
| Nombre | Valor |
| Autor | Tracy |
| Fecha de creación | 05-10-2013 06:09:56 PM |
| Fecha de modificación | 05-10-2013 06:13:31 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| tsg\_modulomod\_id | int(10) | PK/FK ([tsg\_modulo.mod\_id](#PuSeioKGAqA8cQPi)) | No | - |
| tsg\_rolrol\_id | int(10) | PK/FK ([tsg\_rol.rol\_id](#hkYeioKGAqA8cQPP)) | No | - |

##### tsg\_comentario\_proyecto

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | Tabla de observaciones del proyecto |
| Autor | Tracy |
| Fecha de creación | 05-10-2013 09:33:05 PM |
| Fecha de modificación | 05-10-2013 10:43:40 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| cop\_id | int(10) | PK | No | Identificador único de la tabla |
| cop\_descrip | varchar(1000) |  | Si | Descripción |
| tsg\_proyectopro\_id | int(10) | FK ([tsg\_proyecto.pro\_id](#PPNOioKGAqA8cQM5)) | No | - |
| tsg\_usuariousu\_id | int(10) | FK ([tsg\_usuario.usu\_id](#1mn2ioKGAqA8cQMj)) | No | - |
| tsg\_archivoarc\_id | int(10) | FK ([tsg\_archivo.arc\_id](#SXkASoKGAqA8cQWp)) | No | - |
| cop\_fecha\_creacion | datetime | N/A | Si | Fecha de creación |

##### tsg\_cliente

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | Tabla sistema de gestión, está tabla contiene los datos del cliente que interactuara con el sistema. |
| Autor | Tracy |
| Fecha de creación | 24-09-2013 03:30:39 PM |
| Fecha de modificación | 05-10-2013 10:54:11 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| cli\_id | int(10) | PK | No | Identificador único de la cliente. |
| cli\_nombre | varchar(255) | N/A | No | corresponde al nombre del cliente |
| cli\_apellido | varchar(255) | N/A | No | apellido del cliente |
| cli\_correo | varchar(255) | N/A | Si | correo electrónico del cliente |
| cli\_empresa | varchar(255) | N/A | Si | nombre de la empresa en la cual se encuentra posicionado el cliente |
| cli\_rut | varchar(255) | N/A | No | rut del cliente |
| cli\_direccion | varchar(255) | N/A | Si | dirección del cliente |
| cli\_fecha\_ini | timestamp | N/A | No | fecha de inicio del cliente |
| cli\_fecha\_mod | timestamp | N/A | Si | fecha de modificación del cliente |
| cli\_activo | tinyint(1) | N/A | No | se deja la tabla de cliente con la opción para saber si el usuario esta activo o inactivo, en la BD se mostrara oculto si se elimino por sistema. (1 o 0 ) Eliminación logica. |
| cli\_usu\_creador | varchar(255) | N/A | No | Usuario creador |

##### tsg\_historico\_ticket

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | Tabla de registro histórico del ticket |
| Autor | Tracy |
| Fecha de creación | 24-09-2013 03:36:06 PM |
| Fecha de modificación | 05-10-2013 11:04:40 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| his\_id | int(10) | PK | No | identificador único |
| his\_nombre | varchar(100) | N/A | No | nombre del histórico |
| his\_descrip | varchar(1000) | N/A | Si | descripción histórico |
| tsg\_estado\_ticketest\_id | int(10) | N/A | No | - |
| tsg\_tickettic\_id | int(10) | N/A | No | - |
| tsg\_usuariousu\_id | int(10) | N/A | No | - |
| tsg\_proyectopro\_id | int(10) | N/A | No | - |
| tsg\_prioridadpri\_id | int(10) | N/A | No | - |
| tsg\_categoriacat\_id | int(10) | N/A | No | - |

##### tsg\_categoria

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | tabla de las categorías del proyecto |
| Autor | Tracy |
| Fecha de creación | 24-09-2013 03:35:05 PM |
| Fecha de modificación | 05-10-2013 11:04:40 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| cat\_id | int(10) | PK | No | identificador único |
| cat\_nombre | varchar(100) | N/A | No | nombre de la categoría. |
| cat\_descrip | varchar(1000) | N/A | Si | descripción de la categoría |
| cat\_activo | tinyint(1) | N/A | No | eliminación lógica del sistema. |
| cat\_usu\_creador | varchar(255) | N/A | No | Usuario creador |
| cat\_fecha\_creacion | datetime | N/A | No | Fecha creación |
| cat\_fecha\_modificacion | datetime | N/A | Si | Fecha modificación |

##### tsg\_ticket

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | tabla ticket corresponde a los tickets que tiene asociado el proyecto |
| Autor | Tracy |
| Fecha de creación | 24-09-2013 03:29:54 PM |
| Fecha de modificación | 05-10-2013 10:54:11 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| tic\_id | int(10) | PK | No | identificador único del ticket |
| tic\_nombre | varchar(255) | N/A | No | nombre del ticket |
| tic\_fecha\_crea | timestamp | N/A | No | fecha de creación del ticket |
| tsg\_estado\_ticketest\_id | int(10) | FK ([tsg\_estado\_ticket.est\_id](#aVyuioKGAqA8cQN6)) | No | - |
| tsg\_usuariousu\_id | int(10) | FK ([tsg\_usuario.usu\_id](#1mn2ioKGAqA8cQMj)) | No | - |
| tsg\_proyectopro\_id | int(10) | FK ([tsg\_proyecto.pro\_id](#PPNOioKGAqA8cQM5)) | No | - |
| tsg\_prioridadpri\_id | int(10) | FK ([tsg\_prioridad.pri\_id](#aBduioKGAqA8cQO0)) | No | - |
| tsg\_categoriacat\_id | int(10) | FK ([tsg\_categoria.cat\_id](#SMPuioKGAqA8cQPG)) | No | - |
| tic\_destacado | tinyint(1) | N/A | Si | Desatacado (Estrellita) |
| tsg\_tic\_correo\_en\_copia | varchar(1000) | N/A | Si | Correos en copia del cambio de estado |

##### tsg\_estado\_ticket

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | tabla del estado del ticket |
| Autor | Tracy |
| Fecha de creación | 24-09-2013 03:32:19 PM |
| Fecha de modificación | 05-10-2013 11:04:40 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| est\_id | int(10) | PK | No | identificador único |
| est\_nombre | varchar(100) | N/A | No | nombre del estado del ticket |
| est\_descrip | varchar(1000) | N/A | Si | descripción del estado del ticket |
| est\_activo | tinyint(1) | N/A | No | eliminación lógica del sistema |

##### tsg\_estadistica\_diaria

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | Tabla de registro estadístico diario |
| Autor | Tracy |
| Fecha de creación | 24-09-2013 03:37:20 PM |
| Fecha de modificación | 05-10-2013 11:04:40 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| dis\_id | int(10) | PK | No | Identificador único de la tabla |
| dis\_fecha | int(10) | N/A | No | Fecha |
| dis\_total | int(10) | N/A | No | Total de tickets |
| dis\_procesadas | int(10) | N/A | No | Total procesados |
| dis\_pendientes | int(10) | N/A | No | Total pendientes |
| dis\_cerradas | int(10) | N/A | No | Total cerradas |
| tsg\_proyectopro\_id | int(10) | N/A | No | - |

##### tsg\_prioridad

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | prioridad del ticket |
| Autor | Tracy |
| Fecha de creación | 24-09-2013 03:34:09 PM |
| Fecha de modificación | 05-10-2013 09:40:04 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| pri\_id | int(10) | PK | No | identificador único de la prioridad del ticket |
| pri\_nombre | varchar(100) | N/A | No | nombre de la prioridad |
| pri\_descrip | varchar(1000) | N/A | Si | descripción de la prioridad |
| pri\_activo | tinyint(1) | N/A | No | eliminación lógica del sistema. |

##### tsg\_comentario\_ticket

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | comentario asociado al ticket |
| Autor | Tracy |
| Fecha de creación | 24-09-2013 03:33:20 PM |
| Fecha de modificación | 05-10-2013 11:04:40 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| com\_id | int(10) | PK | No | identificador único del comentario |
| com\_descrip | varchar(1000) | N/A | Si | descripción del comentario asociado al ticket |
| tsg\_tickettic\_id | int(10) | FK ([tsg\_ticket.tic\_id](#72DOioKGAqA8cQNR)) | No | - |
| tsg\_usuariousu\_id | int(10) | FK ([tsg\_usuario.usu\_id](#1mn2ioKGAqA8cQMj)) | No | - |
| tsg\_archivoarc\_id | int(10) | FK ([tsg\_archivo.arc\_id](#SXkASoKGAqA8cQWp)) | No | - |
| com\_fecha\_creacion | datetime | N/A | Si | Fecha de creación |

##### tsg\_archivo

|  |  |
| --- | --- |
| Nombre | Valor |
| Documentación | Tabla de archivos adjuntos |
| Autor | Tracy |
| Fecha de creación | 24-09-2013 06:04:34 PM |
| Fecha de modificación | 05-10-2013 11:04:40 PM |

Resumen Columnas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de datos | Limitaciones | Anulable | Documentación |
| arc\_id | int(10) | PK | No | identificador único |
| arc\_nombre | varchar(255) | N/A | No | nombre del archivo adjunto |
| arc\_peso | float(100) | N/A | No | peso del archivo adjunto |
| arc\_url | varchar(255) | N/A | No | url del archivo adjunto |

### 4.5.3 Diseño de las Interfaces

#### 4.5.3.1 Pantalla: Acceso al sistema

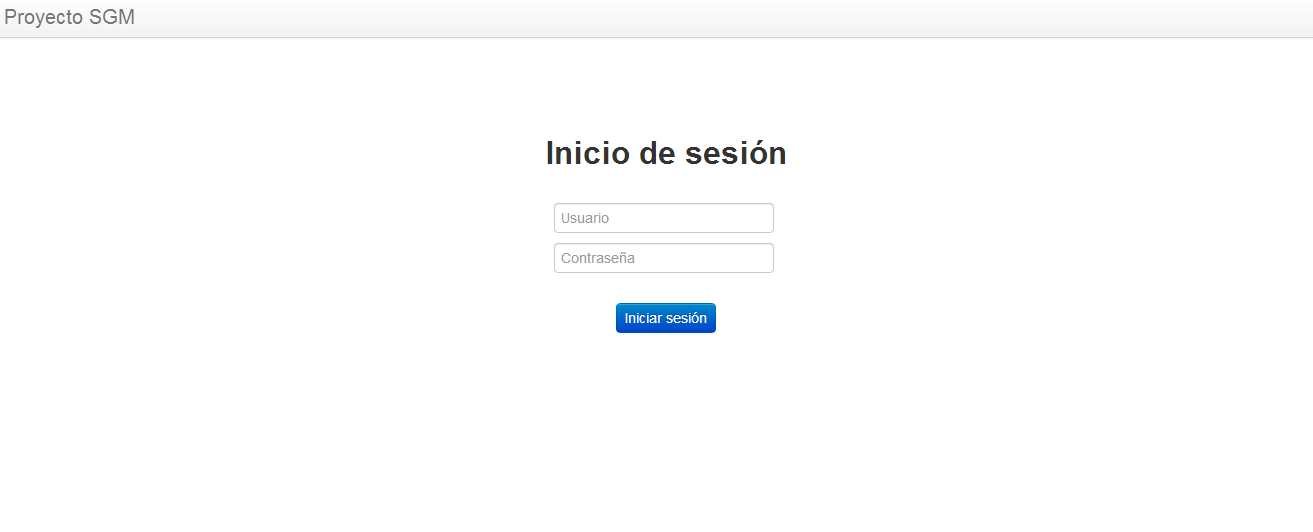


Ilustración 30 - Acceso al sistema

#### 4.5.3.2 Pantalla: Ingreso al sistema

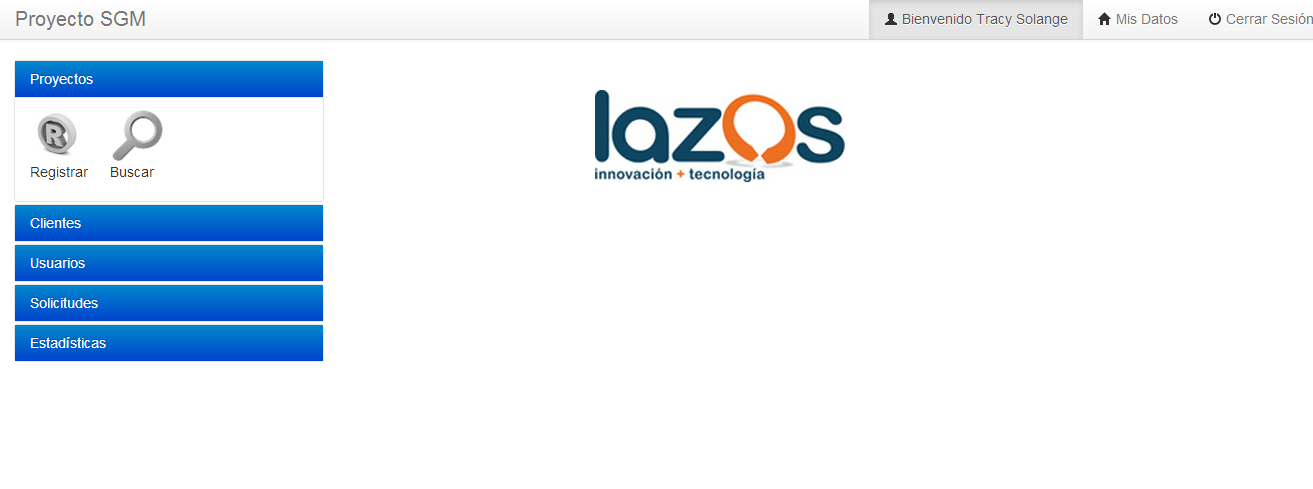


Ilustración 31 - Ingreso al sistema

#### 4.5.3.3 Pantalla: Datos Personales

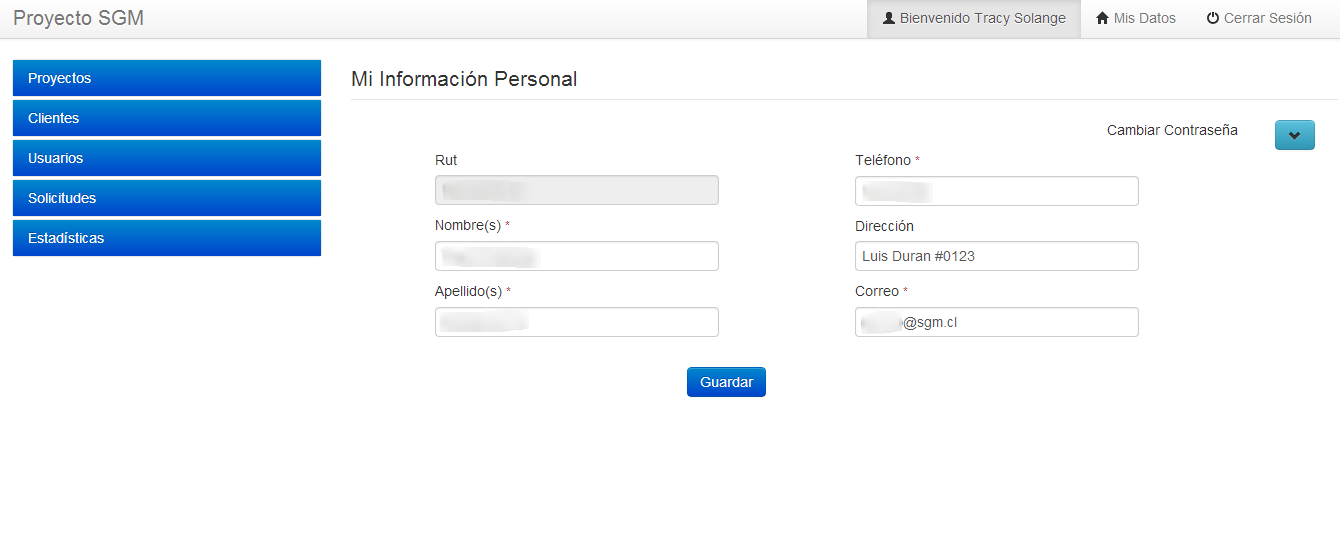


Ilustración 32 - Mis datos

#### 4.5.3.4 Pantalla: Cambio de contraseña

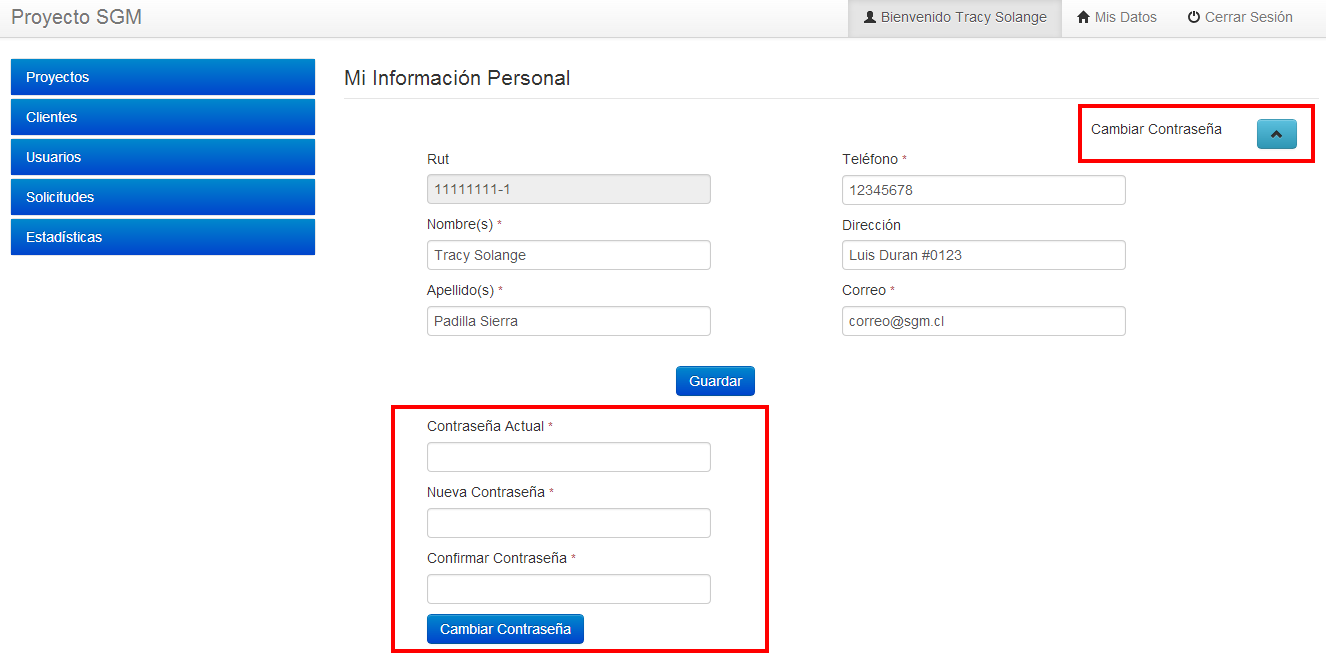


Ilustración 33 - Cambio de contraseña

#### 4.5.3.5 Pantalla: Proyectos - Agregar Proyecto

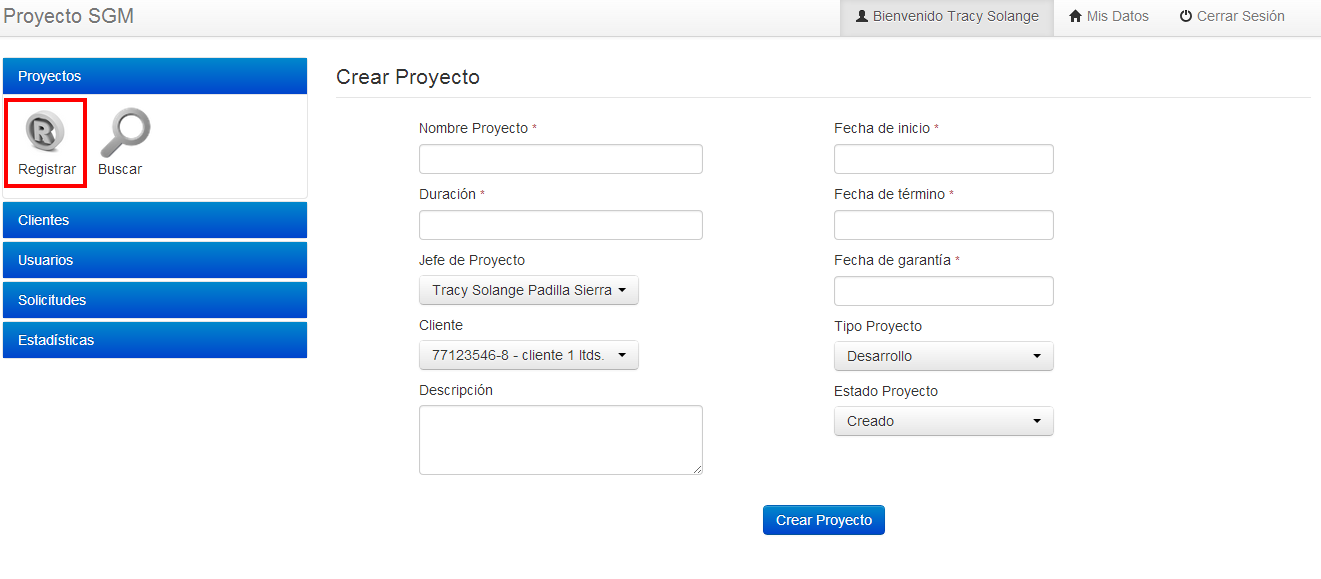


Ilustración 34 - Registrar proyecto

#### 4.5.3.6 Pantalla: Proyectos - Búsqueda Básica

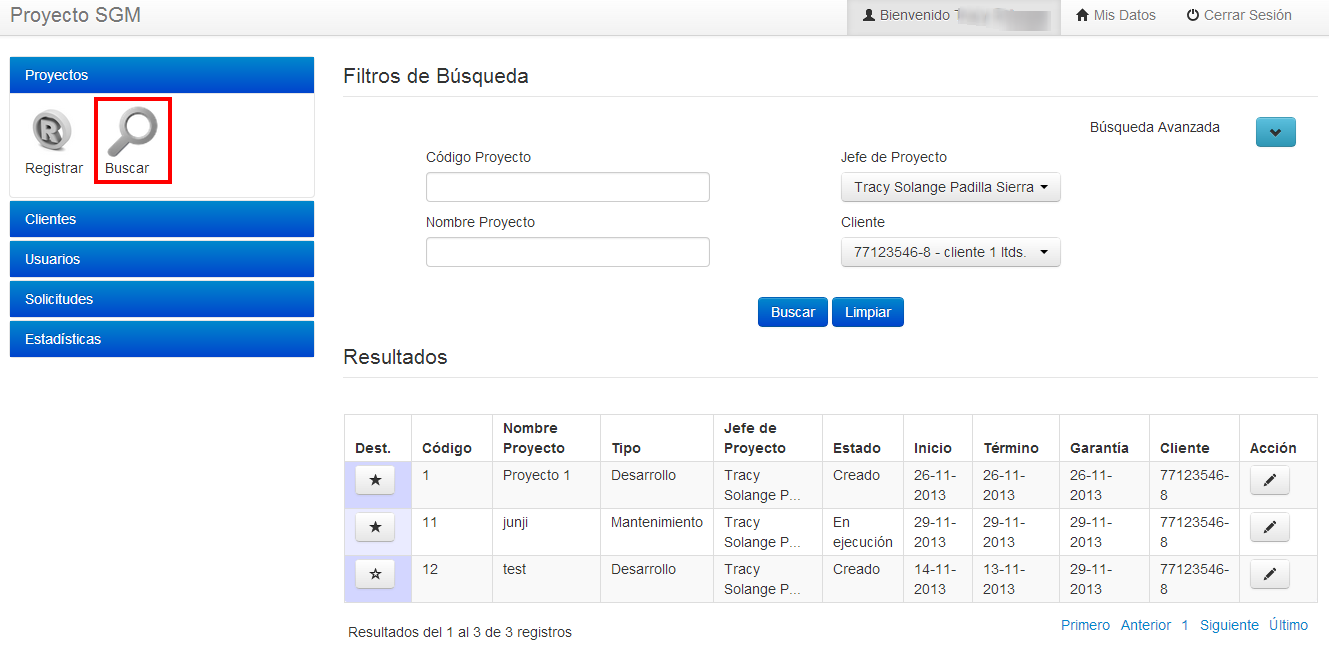


Ilustración 35 - Búsqueda básica

#### 4.5.3.7 Pantalla: Proyectos - Búsqueda Avanzada

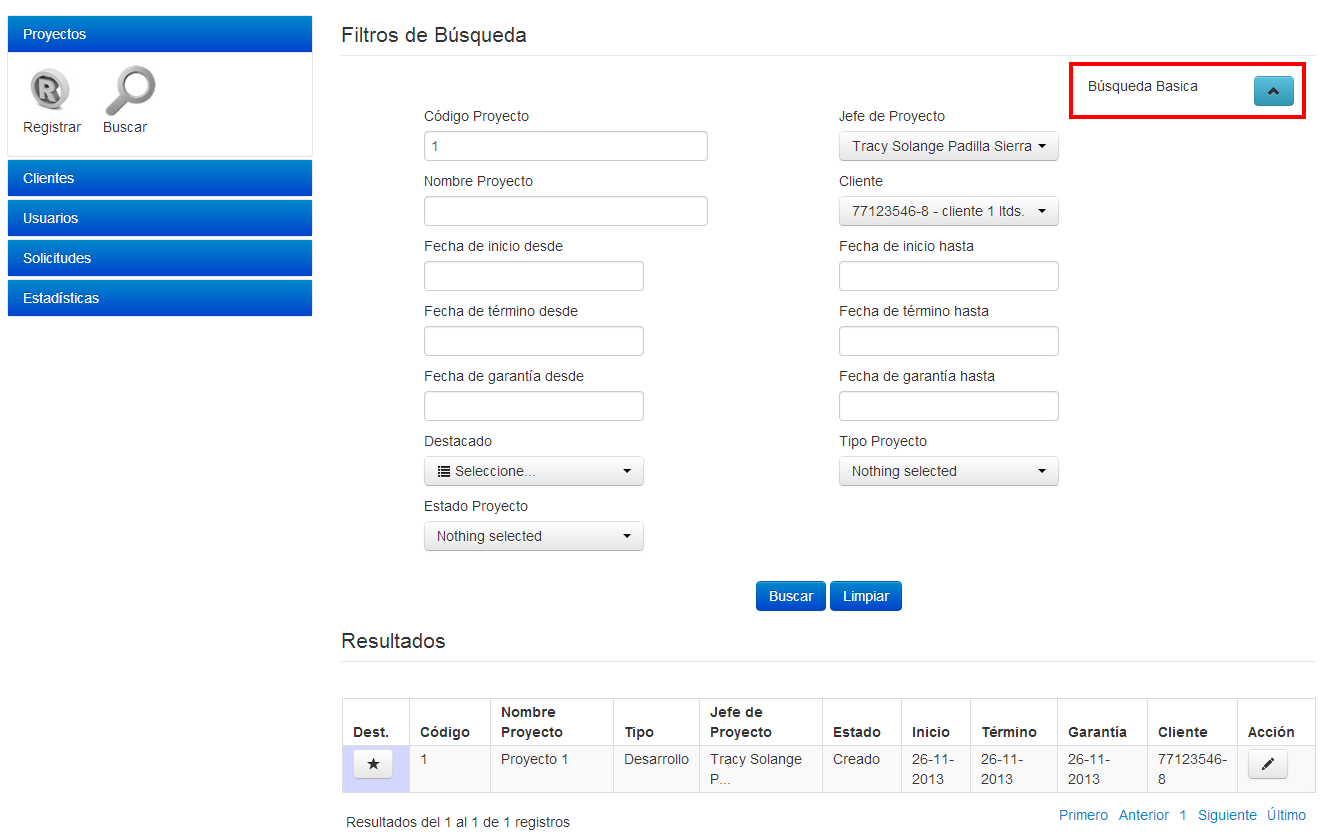


Ilustración 36 - Búsqueda avanzada

#### 4.5.3.8 Pantalla: Proyectos - Modificar Proyecto

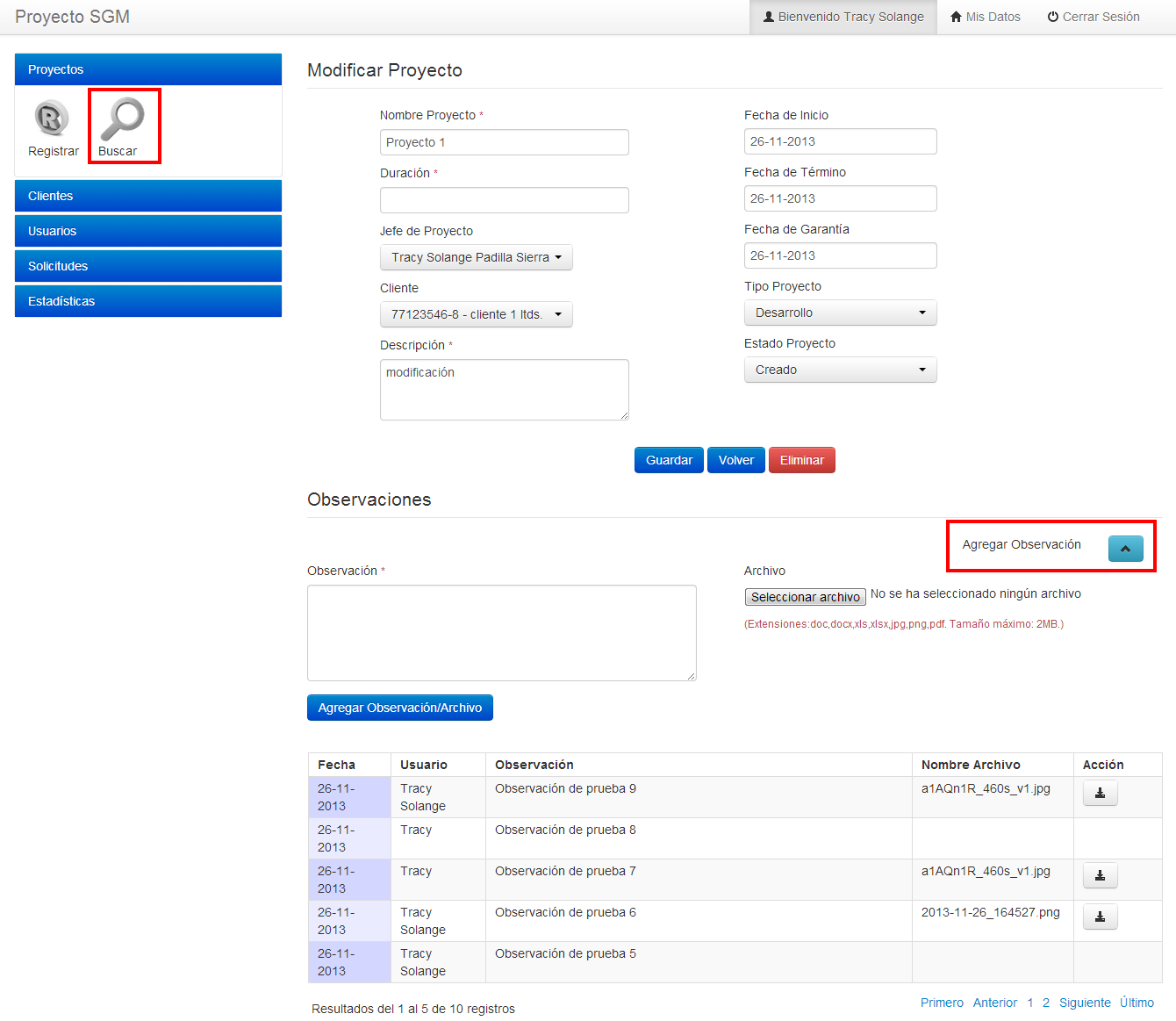


Ilustración 37 - Modificar proyecto

#### 4.5.3.9 Pantalla: Clientes - Agregar Cliente

/\*PENDIENTE\*/

#### 4.5.3.10 Pantalla: Clientes - Búsqueda Básica

/\*PENDIENTE\*/

#### 4.5.3.11 Pantalla: Clientes - Búsqueda Avanzada

/\*PENDIENTE\*/

#### 4.5.3.12 Pantalla: Clientes - Modificar Cliente

/\*PENDIENTE\*/

#### 4.5.3.13 Pantalla: Usuarios - Agregar Usuario

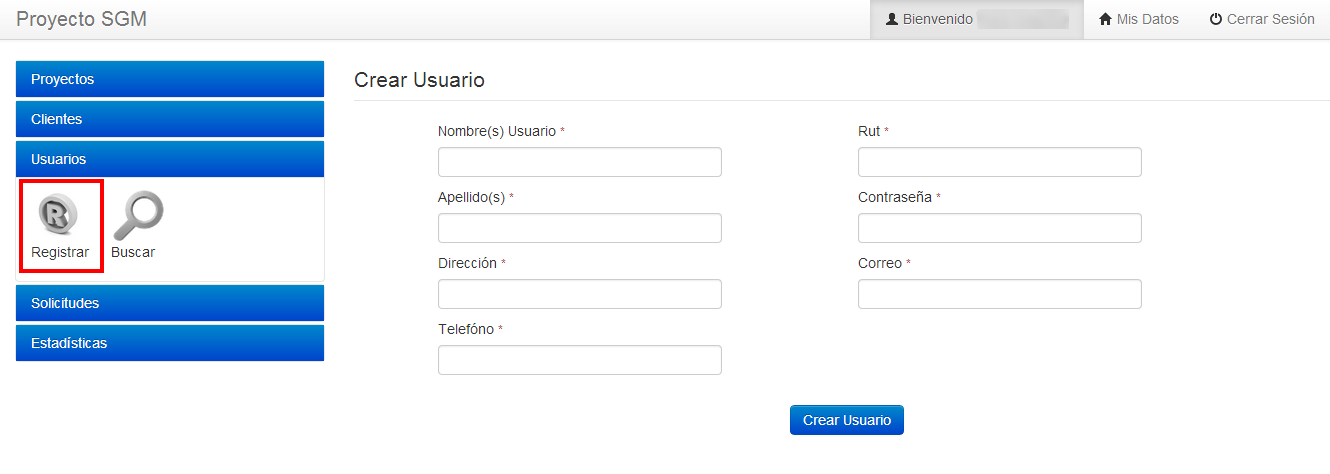


Ilustración 38 - Agregar usuario

#### 4.5.3.14 Pantalla: Usuarios - Buscar Básica

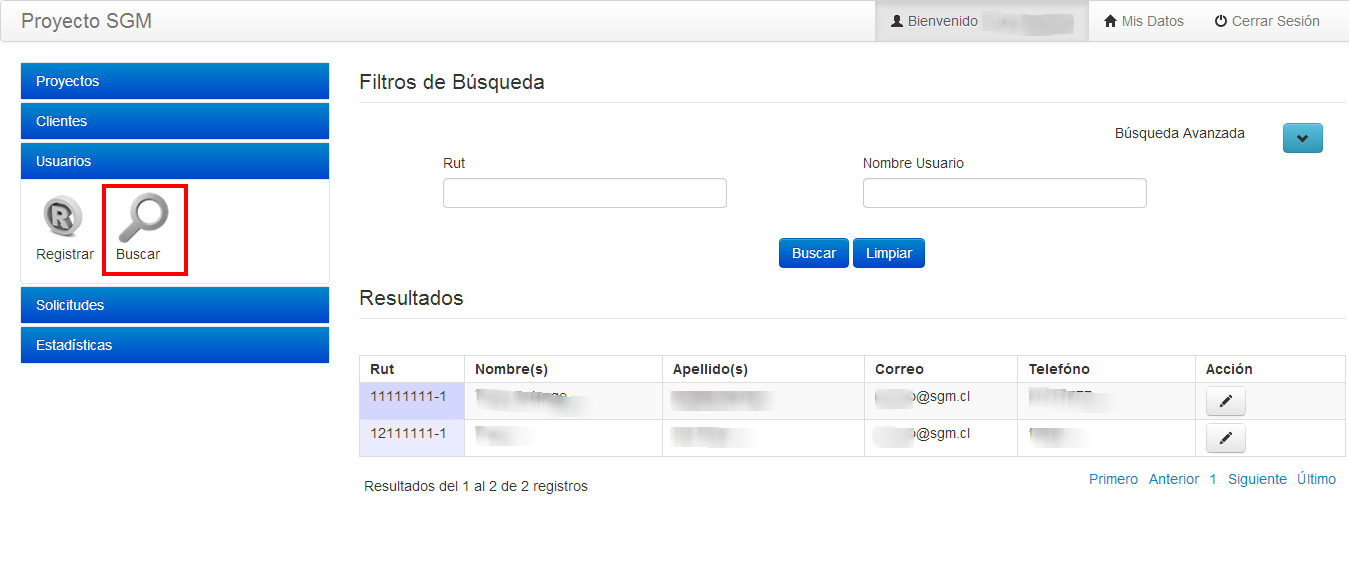


Ilustración 39 - Búsqueda básica

#### 4.5.3.15 Pantalla: Usuarios - Buscar Avanzada

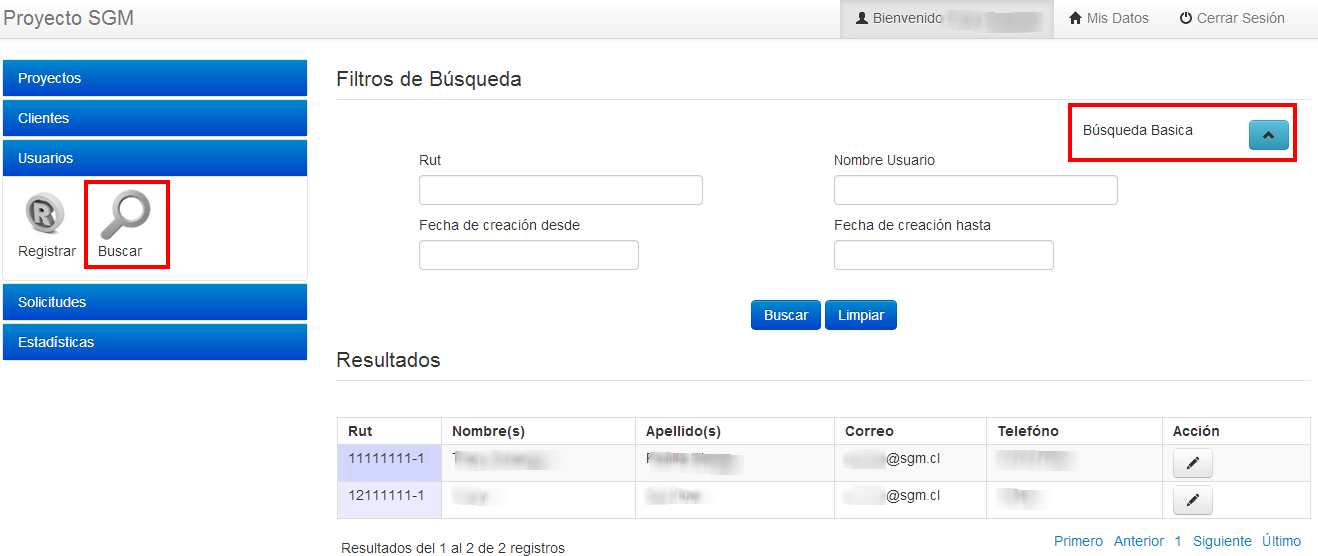


Ilustración 40 - Búsqueda avanzada

#### 4.5.3.16 Pantalla: Usuarios - Modificar Usuario

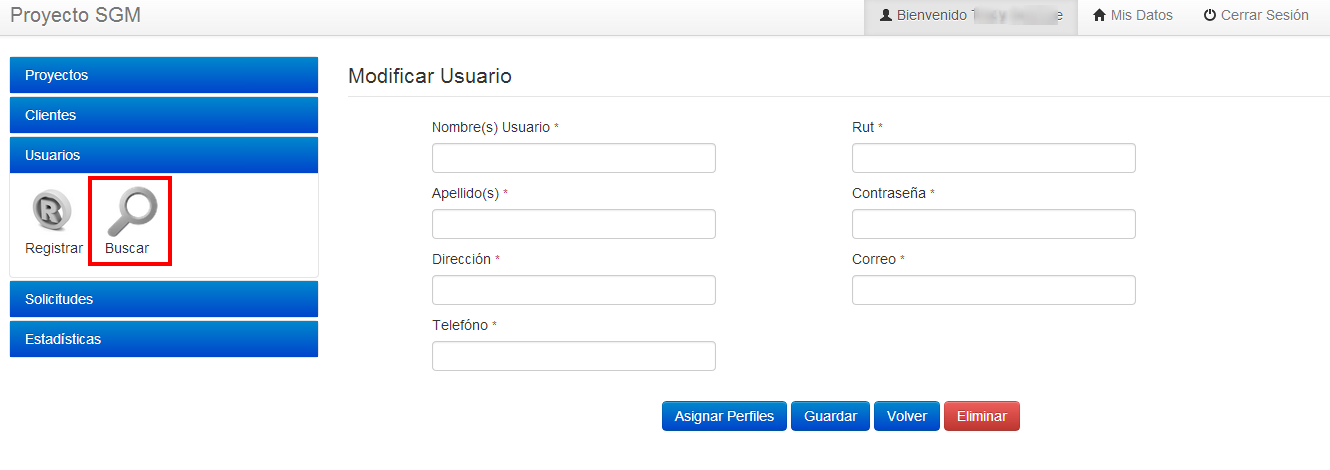


Ilustración 41 - Modificar usuario

#### 4.5.3.17 Pantalla: Usuarios - Asignar Perfiles

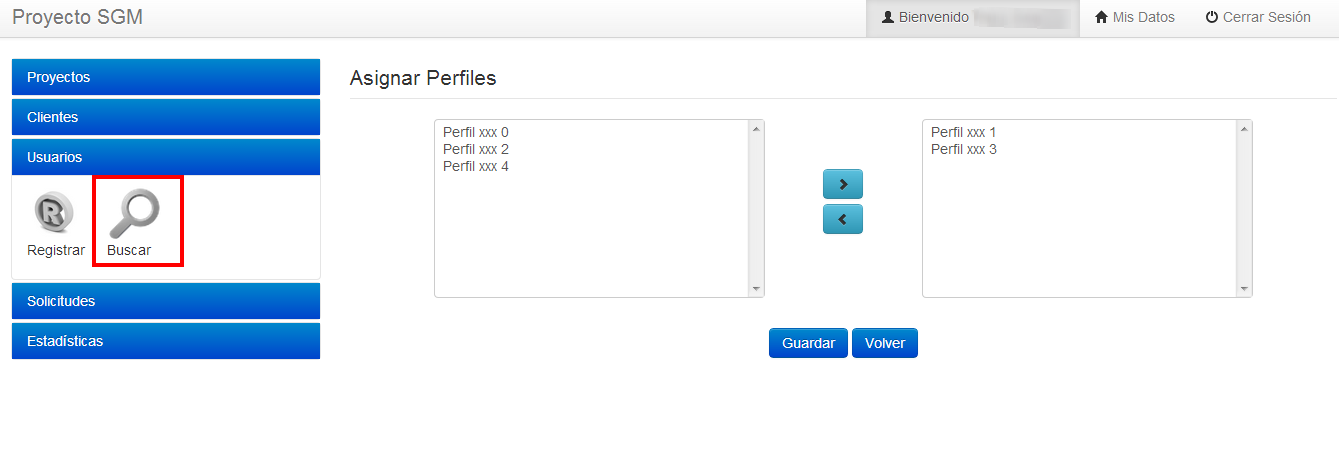


Ilustración 42 - Asignar perfiles

#### 4.5.3.18 Pantalla: Solicitudes - Registrar Solicitud

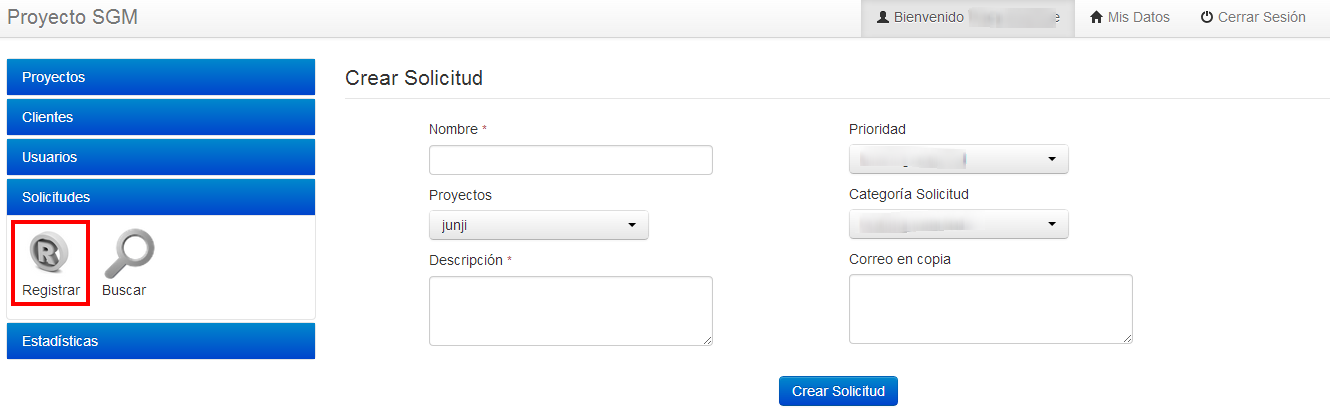


Ilustración 43 - Registrar solicitud

#### 4.5.3.19 Pantalla: Solicitudes - Búsqueda Básica

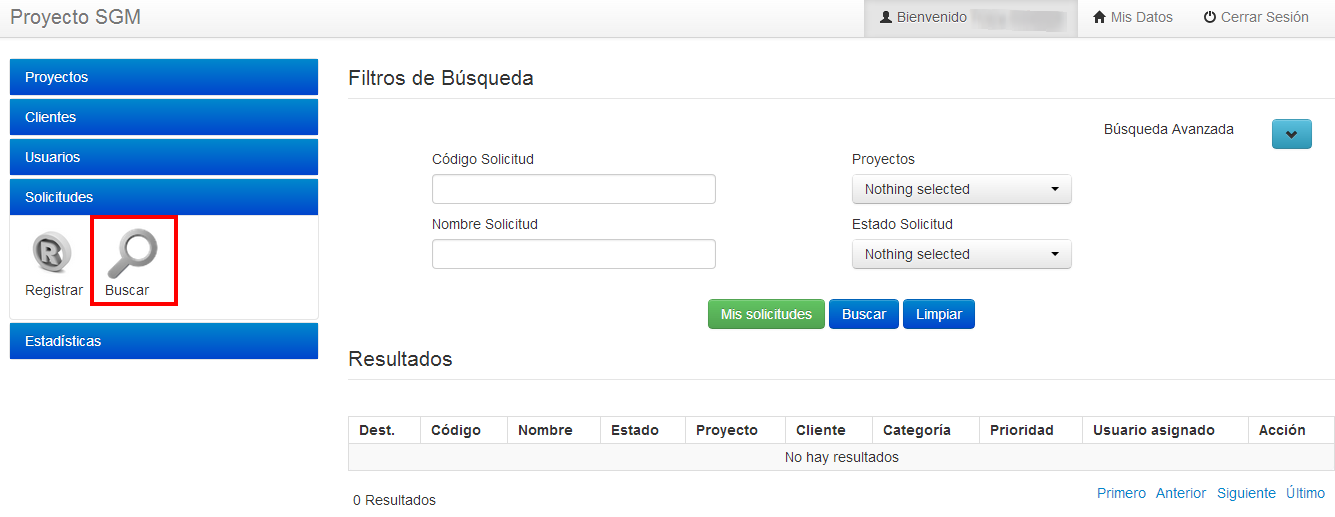


Ilustración 44 - Búsqueda básica

#### 4.5.3.20 Pantalla: Solicitudes - Búsqueda Avanzada

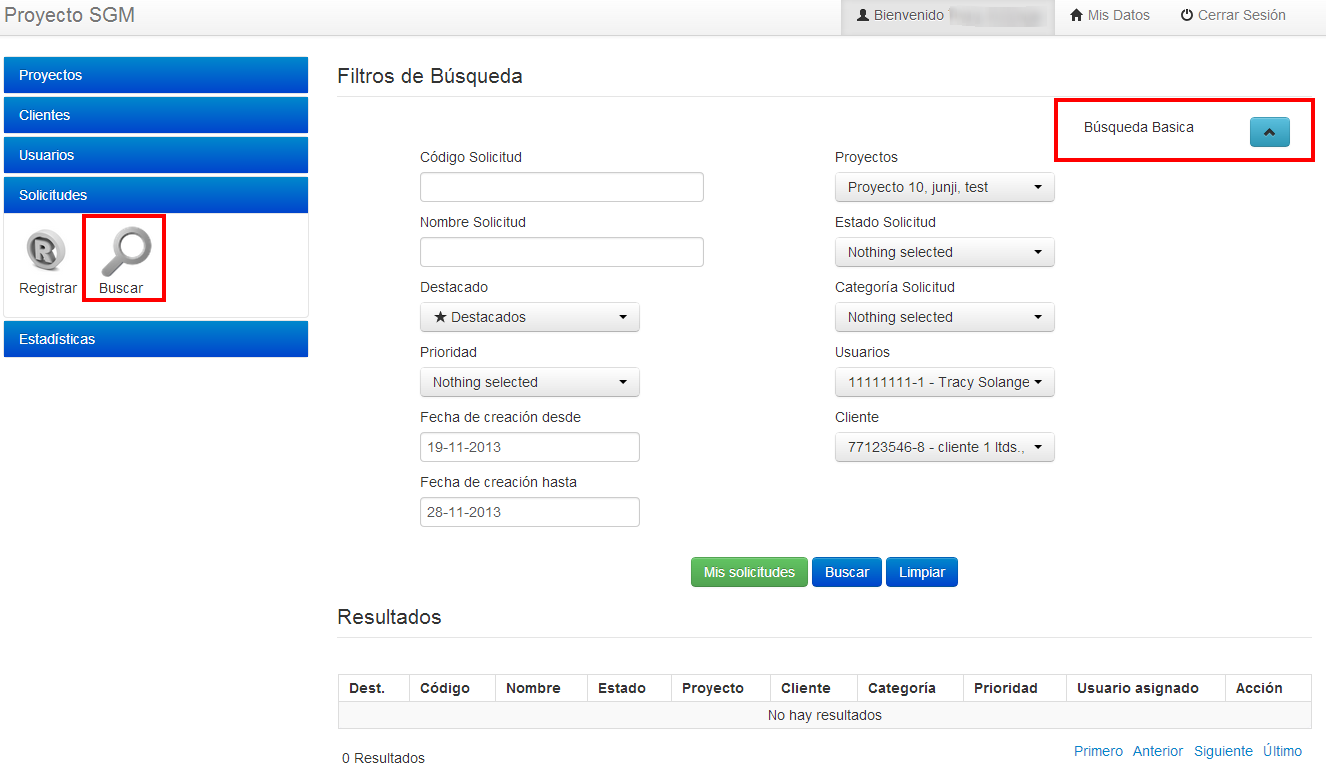


Ilustración 45 - Búsqueda avanzada

#### 4.5.3.21 Pantalla: Solicitudes - Modificar Solicitud

/\*PENDIENTE\*/

#### 4.5.3.22 Pantalla: Estadística - Búsqueda Básica

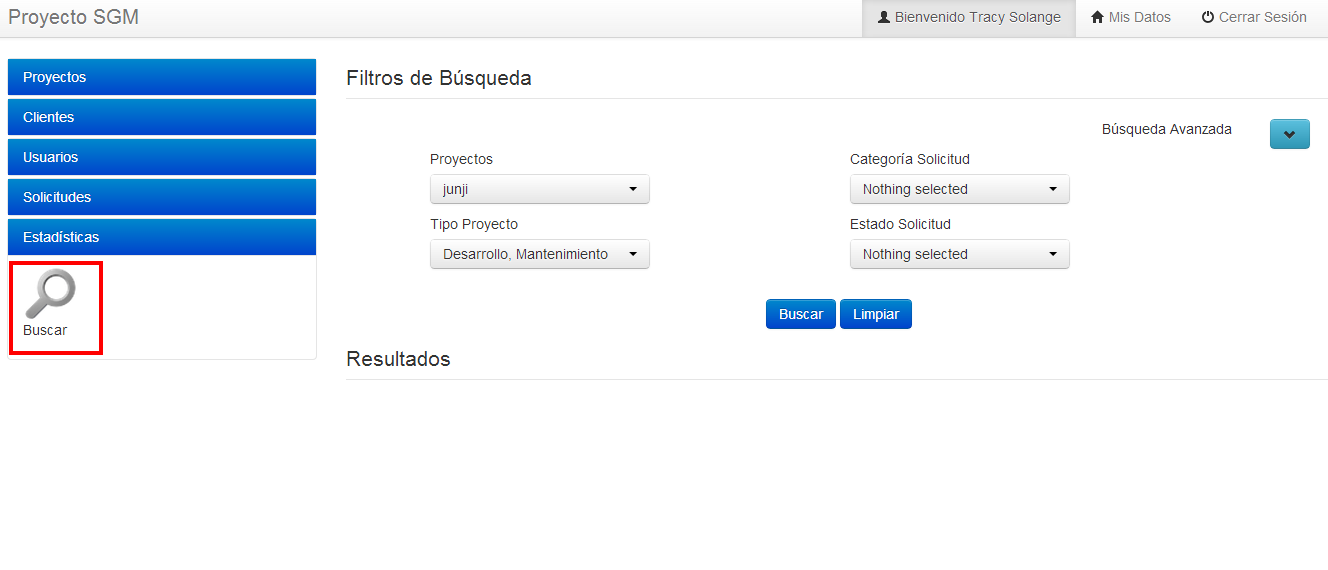


Ilustración 46 - Búsqueda básica

#### 4.5.3.23 Pantalla: Estadística - Búsqueda Avanzada

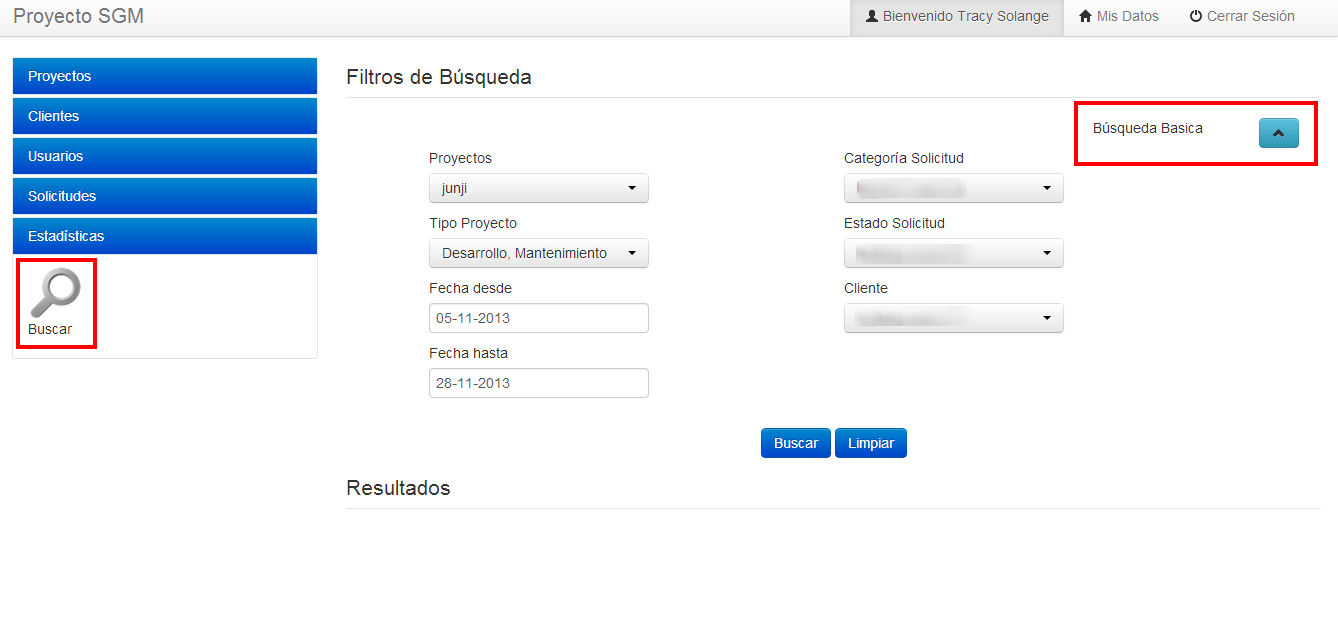


Ilustración 47 - Búsqueda avanzada

## Implementación

# CAPÍTULO V CONCLUSIONES

## Bibliografía

1. Recuperado en Octubre del 2013, de [www.lazos.cl](http://www.lazos.cl) [↑](#footnote-ref-1)
2. Recuperado en Octubre del 2013, de http://www.barloventotech.com.ar/detalle.php?a=sistema-de-gestion-de-tareas---workflow-saas&t=19&d=44 [↑](#footnote-ref-2)
3. Recuperado en Octubre del 2013, de http://osticket.com/ [↑](#footnote-ref-3)
4. Recuperado en Octubre del 2013, de http://www.mantisbt.org/ [↑](#footnote-ref-4)
5. Recuperado en Octtubre del 2013, de http://www.itmplatform.com/es/projects-and-teamwork/pt-presentacion [↑](#footnote-ref-5)
6. Recuperado en Octubre del 2013, de http://php.net/ [↑](#footnote-ref-6)
7. Recuperado en Octubre del 2013, de http://www.asp.net/ [↑](#footnote-ref-7)
8. Recuperado en Octubre del 2013, de http://www.oracle.com/es/products/index.html [↑](#footnote-ref-8)