**DEPARTAMENTO DE**

**SISTEMAS Y COMPUTACIÓN**

Desarrollo de un sistema de tickets de soporte técnico para el Centro de Cómputo del ITL.

I N F O R M E T É C N I C O D E

R E S I D E N C I A P R O F E S I O N A L

INGENIERÍA EN SISTEMAS COMPUTACIONALES

**P R E S E N T A:**

**OSVALDO EMMANUEL RUIZ VELOZ**

**ASESOR:**

**ING. JOSÉ ELÍAS MARTÍNEZ ARIAS**

LEÓN, GUANAJUATO. 2024

**AGRADECIMIENTOS:**

# **RESUMEN:**

El proyecto plantea una solución robusta que servirá como sistema gestor de tickets de soporte técnico del centro de cómputo del ITL, se usa el modelo CRUD para la aplicación de escritorio, un formulario web para la entrada de datos y se usa una base de datos sostenida en el gestor de base de datos “SQL Server”.

Este tipo de soluciones son complejas, pero al conocer que el centro de computo de un Instituto Tecnológico no cuenta con un sistema de este tipo, es imperativo contar con la solución, ya que suena absurdo que una institución con esta trayectoria tenga soluciones primitivas para problemas situaciones que se presentan a diario, como la entrada de tickets de soporte técnico.

**ÍNDICE**

Capítulo Pág.

[**RESUMEN:** 4](#_Toc165018792)

[**INTRODUCCIÓN** 7](#_Toc165018793)

[**1.- CAPÍTULO I. ANTECEDENTES** 7](#_Toc165018794)

[**1.1.- ANTECEDENTES** 7](#_Toc165018795)

[**1.1.1.- INFORMACION GENERAL DE LA EMPRESA** 7](#_Toc165018796)

[**1.1.2.- DESCRIPCIÓN DEL SISTEMA ACTUAL** 11](#_Toc165018797)

[**1.2.- PLANTEAMIENTO DEL PROBLEMA** 11](#_Toc165018798)

[**1.3.- PROPUESTA DE SOLUCIÓN** 12](#_Toc165018799)

[**1.4.- OBJETIVOS DEL PROYECTO** 12](#_Toc165018800)

[**1.4.1.- OBJETIVO GENERAL** 12](#_Toc165018801)

[**1.4.2.- OBJETIVOS ESPECIFICOS** 12](#_Toc165018802)

[**1.5.- JUSTIFICACIÓN DE LA ESTRATEGIA DE SOLUCIÓN** 13](#_Toc165018803)

[**1.6.- ALCANCES Y LIMITACIONES** 14](#_Toc165018804)

[**1.7.- ENTREGABLES FINALES** 14](#_Toc165018805)

[**1.8.- MODELO, MÉTODO Y/O METODOLOGÍA** 14](#_Toc165018806)

[**1.9.- CRONOGRAMA** 15](#_Toc165018807)

[**2.- CAPÍTUO II. MARCO TEÓRICO.** 15](#_Toc165018808)

[**2.1.- DESCRIPCIÓN DE MODELO DE DESARROLLO** 15](#_Toc165018809)

[**2.2.- DESCRIPCION DE PATRONES DE DISEÑO Y/O ARQUITECTURAS A APLICAR** 15](#_Toc165018810)

[**2.3.- DESCRIPCION DE AMBIENTE DE DESARROLLO** 16](#_Toc165018811)

[**2.3.1.- PLATAFORMAS Y/O FRAMEWORKS DE DESARROLLO** 16](#_Toc165018812)

[**2.3.2.- GESTORES DE BASES DE DATOS** 16](#_Toc165018813)

[**2.3.3.- OTRAS HERRAMIENTAS A UTILIZAR** 16](#_Toc165018814)

[**2.4.- SELECCIÓN Y JUSTIFICACION DE AMBIENTE DE DESARROLLO** 16](#_Toc165018815)

[**3.- CAPÍTULO III. ANALISIS DE SISTEMA** 17](#_Toc165018816)

[**3.1.- REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES** 17](#_Toc165018817)

[**3.2.- DIAGRAMA DE MODELADO** 19](#_Toc165018818)

[**3.3.- JERARQUIZACION DE MODULOS E INTEROPERABILIDAD CON OTROS SISTEMAS** 19](#_Toc165018819)

[**4.- CAPÍTULO IV. DISEÑO** 20](#_Toc165018820)

[**4.1.- DIAGRAMA DE CLASES O RELACIONAL** 20](#_Toc165018821)

[**4.2.- DIAGRAMAS DE SECUENCIA Y ACTIVIDAD** 20](#_Toc165018822)

[**4.3.- DISEÑO DE MAQUETADO** 20](#_Toc165018823)

[**5.- CAPÍTULO V. ARQUITECTURA DE LA APLICACIÓN.** 21](#_Toc165018824)

[**5.1- CÓDIGO DE APLICACIÓN DE PATRONES DE DISELO Y PROGRAMACIÓN** 21](#_Toc165018825)

[**5.2.-DESCRIPCIÓN DE LA CODIFICACIÓN DE UN MÓDULO** 21](#_Toc165018826)

[**5.2.1.- INSERCIÓN** 21](#_Toc165018827)

[**5.2.2.-CONSULTA** 21](#_Toc165018828)

[**5.2.3.-EDICIÓN** 21](#_Toc165018829)

[**5.2.4.-ELIMINACIÓN** 21](#_Toc165018830)

[**5.3.- DESCRIPCIÓN DE LA UTILIZACION DE FRAMWORK.** 21](#_Toc165018831)

[**6.-CAPÍTULO VI. DISEÑO Y APLICACIÓN DE PRUEBAS.** 21](#_Toc165018832)

[**6.1.- PLAN DE PRUEBAS** 21](#_Toc165018833)

[**6.2.- DISEÑO DE PRUEBAS** 21](#_Toc165018834)

[**6.2.1.-PRUEBAS UNITARIAS** 21](#_Toc165018835)

[**6.2.2.-PRUEBAS DE INTEGRACIÓN** 21](#_Toc165018836)

[**6.2.3.-PRUEBAS DE SISTEMA** 21](#_Toc165018837)

[**6.2.4.-PRUEBAS DE COMPATIBILIDAD CON NAVEGADORES** 21](#_Toc165018838)

[**CONCLUSIONES** 22](#_Toc165018839)

[**BIBLIOGRAFÍA** 22](#_Toc165018840)

**INDICE DE FIGURAS**

Figura Pág.

[Figura 1: Ubicación del ITL 10](#_Toc165022832)

[Figura 2: Organigrama de la Institución 11](#_Toc165022833)

# **INTRODUCCIÓN**

El propósito de este proyecto es desarrollar un sistema híbrido que sirva como gestor de tickets de soporte técnico del centro de cómputo del ITL. Dicho sistema consiste en un formulario web, que servirá como entrada de datos, y una aplicación de escritorio que será para administrar todas las entradas. El sistema se desarrollará en distintos lenguajes de programación que, a continuación, mencionaré:

* Aplicación de escritorio: Uso de Python
* Formulario web: Uso de JavaScript, HTML y CSS.
* Base de Datos: Uso del Sistema Gestor de Bases de Datos “SQL Server”
* API: Uso de C#

Se uso el modelo CRUD para toda la aplicación y para el cifrado de contraseñas Se uso el cifrado SHA-256, para brindar seguridad a la aplicación.

# **1.- CAPÍTULO I. ANTECEDENTES**

## **1.1.- ANTECEDENTES**

### **1.1.1.- INFORMACION GENERAL DE LA EMPRESA**

El Instituto Tecnológico de León es una institución de educación superior. La institución tiene como objetivo lograr la vinculación real y efectiva del Instituto en los aspectos social, político, económico y productivo de la región.

La misión del ITL: Formar profesionales íntegros que promuevan la cultura, los valores humanos y el conocimiento científico, que se orienten a un crecimiento constante y trascendente manteniendo su arraigo con la sociedad mediante una conducta de servicio que permita proporcionar calidad de vida a su comunidad.

La visión del ITL: En el año 2025 el Instituto Tecnológico de León, comprometido con un proyecto de alta calidad e innovación, congruente con su naturaleza académica y pertinente con relación a las necesidades del país, se consolida en sus procesos educativos, con el reconocimiento público en la búsqueda de la equidad, el humanismo y el compromiso con la construcción de una sociedad mejor.

Los valores del ITL:

* Compromiso
* Calidad
* Equidad
* Humanismo
* Congruencia

Cuenta con 8 licenciaturas, 3 maestrías y un doctorado.

Las licenciaturas que se ofertan son:

* Ingeniería Electrónica.
* Ingeniería Electromecánica.
* Ingeniería en Gestión Empresarial
* Ingeniería Industrial
* Ingeniería en Logística
* Ingeniería Mecatrónica
* Ingeniería en Sistemas Computacionales
* Ingeniería en Tecnologías de la Información y Comunicación

Las maestrías que ofertan son:

* Maestría en Ciencias de la ingeniería
* Maestría en Ciencias de la computación

El doctorado que oferta es en colaboración con otro Instituto Tecnológico:

* Doctorado Interinstitucional en Ciencias en Computación.

El Instituto Tecnológico de León está ubicado en Privada Tecnológico, Industrial Julian de Obregon, León de los Aldama, Gto.

Mapa

Descripción generada automáticamente

Figura 1: Ubicación del ITL

Teléfono: 4777105200

Se estaría laborando dentro del Centro de cómputo a cargo del Dr. Carlos Lino Ramírez siendo el jefe del centro de cómputo.

A continuación, se muestra el Organigrama de la empresa:

**Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja**

Figura 2: Organigrama de la Institución

### **1.1.2.- DESCRIPCIÓN DEL SISTEMA ACTUAL**

Actualmente, el Instituto Tecnológico de León cuenta con un sistema de tickets para el soporte técnico, el cual se basa en formularios de Google. Los datos recabados a través de estos formularios se envían a un documento de hoja de cálculo de Google.

El primer paso al recibir un reporte de parte de: alumno, docente, personal administrativo, jefe de departamento, subdirector(a) y director(a) (ya sea por correo, mensaje, de manera personal o llamada) que requiera atención, la solicitud se anota en una libreta para llevar un seguimiento completo desde el inicio hasta la conclusión. Una vez que los reportes están registrados, el equipo se encarga de completar los formularios y actualizar la hoja de cálculo correspondiente para presentar los informes al jefe del centro de cómputo. Todo esto si es que no se hizo el registro desde el formulario como primer recurso.

## **1.2.- PLANTEAMIENTO DEL PROBLEMA**

En la actualidad, pensar en trabajar información importante en físico, implicar el uso de hojas, folders, archiveros, etc… es algo usual, pero que definitivamente, con el paso del tiempo, se volverá algo obsoleto y difícil de manipular. Y pensar que en el centro de cómputo del Instituto Tecnológico de León manejan información de soporte técnico en formato físico, es algo inverosímil. Esto puede traer muchos problemas, los cuales enlistaré:

* Traspapeleo: Si se trabaja en formato físico es fácil que algún ticket de soporte técnico pueda quedar al fondo de un cajón, en la basura, en medio de una libreta o un sinfín de escenarios más.
* Perdida de información: Hay varias maneras en las que se puede perder la caja de registros de soporte técnico, como si ocurre un incendio, si roban la caja, si ocurre un desastre natural.
* Falta de información: Al levantar el ticket se puede olvidar pedir algún dato que después puede ser de utilidad.
* Ausencia de estadísticas: Las estadísticas son algo importante en el manejo de datos para ver tendencias y el comportamiento de datos, si bien, no es imposible realizar estadísticas con información contenida en físico, si es tardado ya que se deben migrar los datos a una base de datos o archivo digital para poder computarlos.
* Pérdida de tiempo: El estar haciendo tickets a mano, puede significar perder productividad dentro del centro de cómputo, ya que esta actividad se puede delegar a una aplicación.

## **1.3.- PROPUESTA DE SOLUCIÓN**

El proyecto se sustenta en la urgencia de tener un sistema de tickets digital y de fácil manejo, por lo que se plantea un sistema de tickets híbrido, contando con una aplicación de escritorio, basada en el modelo CRUD, un formulario web, que servirá para obtener las entradas de tickets, y una base de datos sostenida en el SGBD “SQL Server”, toda la información se administrará mediante una API, esto facilitará todo el manejo de datos.

## **1.4.- OBJETIVOS DEL PROYECTO**

### **1.4.1.- OBJETIVO GENERAL**

Desarrollar sistema de tickets anidado a un portal web para ingresar y administrar solicitudes de soporte técnico en el Instituto Tecnológico de León, con el fin de automatizar y gestionar los tickets.

### **1.4.2.- OBJETIVOS ESPECIFICOS**

* Realizar la ingeniería de requerimientos.
* Desarrollar tablas relacionales de la base de datos.
* Modelar y crear una base de datos.
* Modelar y desarrollar las páginas web para el formulario.
* Modelar y crear la aplicación de escritorio para los encargados del centro de cómputo.
* Realizar la conexión de la base de datos con el formulario y con la aplicación de escritorio.
* Realizar las pruebas generales del portal web, aplicación y base de datos.
* Implementar un servidor para el sistema asignado por el centro de cómputo.

## **1.5.- JUSTIFICACIÓN DE LA ESTRATEGIA DE SOLUCIÓN**

El Instituto Tecnológico de León, al ser un centro de estudios tan extenso y con tanto personal, se enfrenta a una creciente demanda de servicios de soporte técnico. La implementación de un sistema de tickets se presenta como una solución esencial que ha tardado en concretarse. El proceso actual, que implica el registro manual de reportes seguido por el llenado de formularios para la presentación de resultados al jefe de departamento, se revela como un procedimiento oneroso y propenso a demoras.

El soporte técnico, siendo un servicio con una demanda constante, requiere con urgencia la automatización de estas operaciones para agilizar la gestión de reportes. La carga manual no solo consume tiempo valioso, sino que también introduce la posibilidad de pérdida de información y errores humanos. La implementación de un sistema automatizado no solo optimizará estos procesos, sino que también facilitará una evaluación más eficiente de los resultados por parte del departamento.

En resumen, la implementación de un sistema automatizado no solo responde a la demanda actual de servicios de soporte técnico, sino que también sienta las bases para una mejora continua y una mayor alineación del Instituto con las mejores prácticas tecnológicas.

## **1.6.- ALCANCES Y LIMITACIONES**

**Alcances:**El sistema de tickets será utilizado por varios usuarios. Los alumnos, docentes, personal administrativo, jefe de departamento, subdirector(a) y director(a) tendrán la posibilidad de ingresar un ticket de atención a soporte técnico, mientras que el personal del centro de cómputo podrá administrar los tickets ingresados.

**Limitaciones**:

* El sistema no mandará correos electrónicos de aviso al usuario.
* No estará disponible en otro idioma.
* Solo el rol de administrador podrá ver las estadísticas que proporciona el sistema.
* No se requerirá una cuenta para realizar la petición de un ticket.
* Uso exclusivo del centro de cómputo del ITL.

## **1.7.- ENTREGABLES FINALES**

* Documentación de requerimientos.
* Diagramas de casos de uso, clases, entidad relación.
* Manual del usuario.
* Propuesta para futuras mejoras al sistema.
* Informe técnico de residencias.

## **1.8.- MODELO, MÉTODO Y/O METODOLOGÍA**

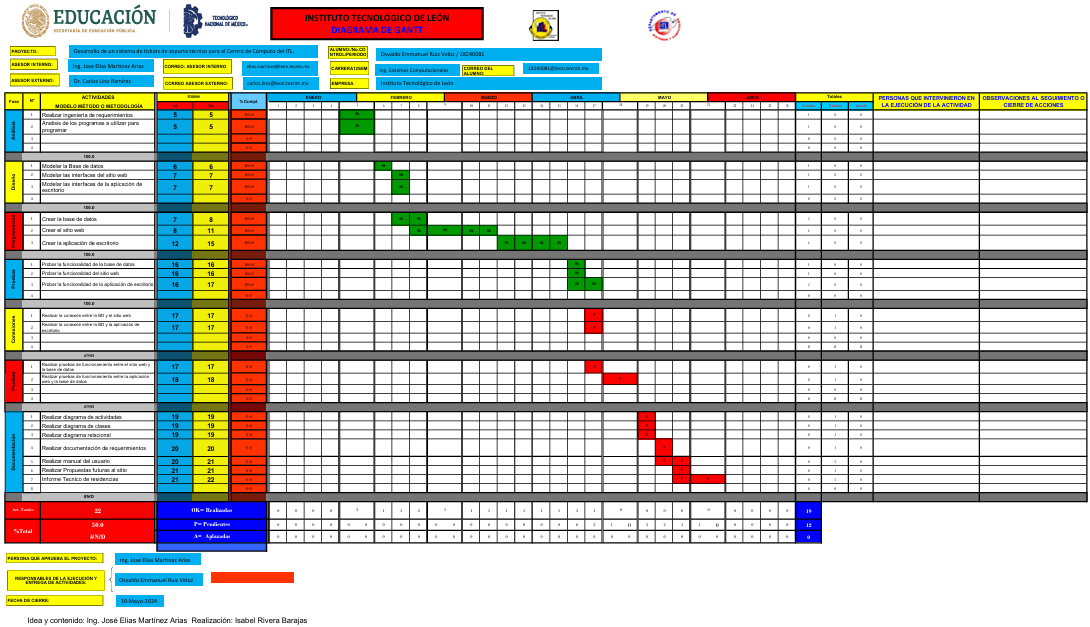
Para el desarrollo del proyecto se empleará la metodología de cascada, teniendo en cuenta la estructura y los requisitos específicos del proyecto, así como la necesidad de una planificación detallada y secuencial.

La metodología de cascada es un enfoque lineal y secuencial para el desarrollo de software, donde cada fase debe completarse antes de pasar a la siguiente. En este contexto el proyecto se beneficiará de la rigurosidad de la metodología de cascada, que asegura una planificación detallada desde el inicio y una clara comprensión de los requisitos antes de avanzar por las etapas de Análisis, diseño, programación, conexiones, pruebas y entregables (documentación).

Como se estipuló con anterioridad las fases que tendrá el proyecto son las siguientes, acompañado de lo que se hará en cada etapa:

* **Análisis**: Para la ingeniería de requerimientos y determinar que lenguajes de programación se usarán
* **Diseño**: Para modelar las interfaces web, de escritorio y modelar las bases de datos.
* **Programación**: Para codificar el portal web, la aplicación de escritorio y la base de datos.
* **Conexiones**: Para conectar el portal web con la base de datos, y la aplicación de escritorio con la base de datos.
* **Pruebas**: Para verificar que todo funcione correctamente, después de la programación y de la conexión.
* **Entregables** (documentación): Para redactar toda la documentación necesaria.

## **1.9.- CRONOGRAMA**



# **2.- CAPÍTUO II. MARCO TEÓRICO.**

## **2.1.- DESCRIPCIÓN DE MODELO DE DESARROLLO**

## **2.2.- DESCRIPCION DE PATRONES DE DISEÑO Y/O ARQUITECTURAS A APLICAR**

## **2.3.- DESCRIPCION DE AMBIENTE DE DESARROLLO**

### **2.3.1.- PLATAFORMAS Y/O FRAMEWORKS DE DESARROLLO**

### **2.3.2.- GESTORES DE BASES DE DATOS**

### **2.3.3.- OTRAS HERRAMIENTAS A UTILIZAR**

## **2.4.- SELECCIÓN Y JUSTIFICACION DE AMBIENTE DE DESARROLLO**

Se ha analizado y concluido que las plataformas y lenguajes de programación serán:

* **Sistema gestor de base de datos**: Se usará SQL Server. Se cuenta con la versión gratuita, y es un SGBD muy amplio y tiene una alta gama de herramientas. Pensando en el futuro, este SGBD podrá alojar N cantidad de datos, se pueden hacer auditorias, triggers para tener un control amplio en los datos. También es compatible con muchos lenguajes de programación, siendo esto una ventaja.
* **Ambiente de desarrollo para el formulario web**: Se usará HTML, para codificar la página web/formulario, CSS para los estilos que tendrá la página, y JavaScript para poder hacer operaciones en la página, esta tercia es confiable y altamente usada en el mercado.
* **Ambiente de desarrollo de la aplicación de escritorio**: Se usará Python, ya que es un lenguaje muy sencillo de usar y capaz de crear Interfaces Graficas de Usuario.
* **Ambiente de desarrollo de API**: Se usará C#, ya que es un lenguaje robusto pero sólido, aparte de que, al desarrollarlo con el modelo DAO ya que ayudan a mantener un código más modular, flexible y fácil de mantener al separar las preocupaciones de acceso a datos de otras partes de la aplicación.

# **3.- CAPÍTULO III. ANALISIS DE SISTEMA**

## **3.1.- REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES**

**Requerimientos Funcionales:**

En la siguiente tabla se describirán los requerimientos funcionales del proyecto.

***De la página web:***

|  |  |  |
| --- | --- | --- |
| ID | Requerimiento | Descripción |
| RF001 | Formulario de soporte técnico | Deberá contener todos los campos para obtener los datos y levantar el ticket. |
| RF002 | Validar Datos | Tendrá que checar que los campos del formulario se hayan llenado para dar por terminada la operación. |

***De la aplicación de escritorio:***

|  |  |  |
| --- | --- | --- |
| ID | Requerimiento | Descripción |
| RF003 | Inicio de sesión | Se debe de asegurar que el ingreso a la aplicación sea seguro mediante un login cifrado. |
| RF004 | Acceso a datos | Permitirá a los usuarios la lectura y manipulación de datos, recabados con anterioridad en el formulario. |
| RF005 | Cambiar estados del ticket | Gestionar si el ticket ya se atendió o está siendo atendido. |
| RF006 | Obtener estadísticas. | Mostrará gráficos con la incidencia de tickets y el departamento al que pertenecen. |

**Requerimientos No Funcionales:**En esta parte de la documentación se mostrarán los requerimientos no funcionales que debe de contener el sistema.

***Para la Pagina web:***

|  |  |  |
| --- | --- | --- |
| ID | Requerimiento | Descripción |
| RNF001 | Seguridad | Debe de ser una página fuera del alcance de hackers  Nota: ¿Eso debe debería ir más orientado al servidor? |
| RNF002 | Rendimiento | Deberá de cargar rápido el formulario |
| RNF003 | Navegabilidad | Debe ser una página fácil e intuitiva de usar para todos los usuarios |
| RNF004 | Mantenimiento | Deberá ser fácil de manipular al dar soporte a la página. |

***Para aplicación de escritorio:***

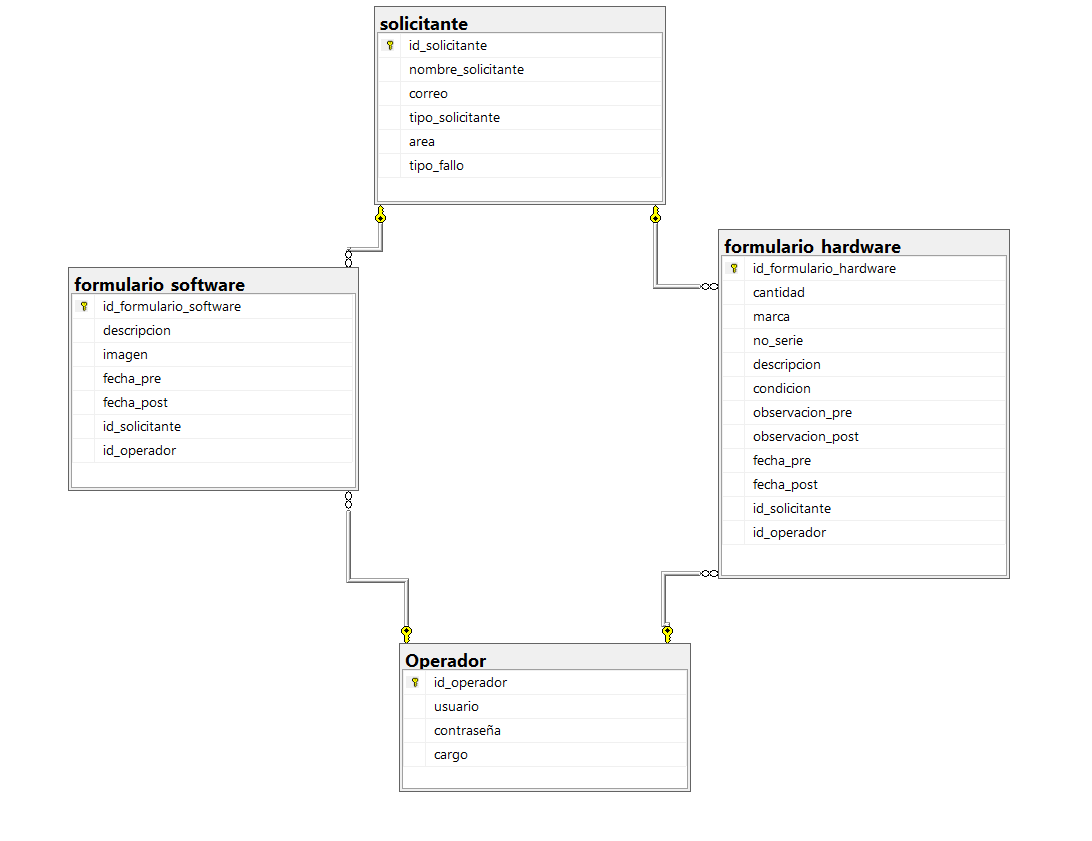
|  |  |  |
| --- | --- | --- |
| ID | Requerimiento | Descripción |
| RNF005 | Seguridad | Se deben de encriptar de los datos de inicio de sesión |
| RNF006 | Rendimiento | Deberá ser eficaz y eficiente en cargar y mostrar los datos |
| RNF007 | Navegabilidad | Debe ser una aplicación fácil e intuitiva de usar para todos los usuarios |
| RNF008 | Mantenimiento | Deberá ser fácil de manipular al dar soporte. |

## **3.2.- DIAGRAMA DE MODELADO**

## **3.3.- JERARQUIZACION DE MODULOS E INTEROPERABILIDAD CON OTROS SISTEMAS**

# **4.- CAPÍTULO IV. DISEÑO**

## **4.1.- DIAGRAMA DE CLASES O RELACIONAL**



## **4.2.- DIAGRAMAS DE SECUENCIA Y ACTIVIDAD**

## **4.3.- DISEÑO DE MAQUETADO**

# **5.- CAPÍTULO V. ARQUITECTURA DE LA APLICACIÓN.**

## **5.1- CÓDIGO DE APLICACIÓN DE PATRONES DE DISEÑO Y PROGRAMACIÓN**

## **5.2.-DESCRIPCIÓN DE LA CODIFICACIÓN DE UN MÓDULO**

### **5.2.1.- INSERCIÓN**

### **5.2.2.-CONSULTA**

### **5.2.3.-EDICIÓN**

### **5.2.4.-ELIMINACIÓN**

## **5.3.- DESCRIPCIÓN DE LA UTILIZACION DE FRAMWORK.**

# **6.-CAPÍTULO VI. DISEÑO Y APLICACIÓN DE PRUEBAS.**

## **6.1.- PLAN DE PRUEBAS**

## **6.2.- DISEÑO DE PRUEBAS**

### **6.2.1.-PRUEBAS UNITARIAS**

### **6.2.2.-PRUEBAS DE INTEGRACIÓN**

### **6.2.3.-PRUEBAS DE SISTEMA**

### **6.2.4.-PRUEBAS DE COMPATIBILIDAD CON NAVEGADORES**

# **CONCLUSIONES**

# **BIBLIOGRAFÍA**

**No hay ninguna fuente en el documento actual.**