



Facultad de Ciencias y Tecnologías Computacionales
Equipo de Servicios de Traductores e Intérpretes

Aplicación web para la gestión de incidencias tecnológicas en el Equipo de Servicios para Traductores e Intérpretes

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor: Yasmin R Vegueria Portuondo

Tutor(es): Dra. C. Natalia Martínez Sánchez
Ing. Lidice Veguería López

La Habana, noviembre de 2022

Año 63 de la Revolución

PENSAMIENTO

“Nuestra recompensa se encuentra en el esfuerzo y no en el resultado, un esfuerzo total es una victoria completa.”

Mahatma Gandhi.

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título “**Aplicación web para la gestión de incidencias tecnológicas en el Equipo de Servicios para Traductores e intérpretes**”, concede a la Universidad de las Ciencias Informáticas

los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como únicos autores de su contenido. Para que así conste firma la presente a los <día> días del mes de noviembre del año 2022.

Yasmin R Vegueria Portuondo

Firma del Autor

Natalia Martínez Sánchez

Firma del Tutor

Lidice Veguería Portuondo

Firma del Tutor

DATOS DE CONTACTO

Lidice Veguería López, Ingeniera Informática, se desempeña como Jefa de Departamento de Informática y Electrónica en el Equipo de Servicios de Traductores e intérpretes. Tiene más de 20 años de experiencia laboral, se ha especializado en el diseño de bases de datos.

Natalia Martínez Sánchez: graduada de Lic. Cibernética Matemática, Máster en Computación aplicada y Doctora en Ciencias Técnicas. Es Vicerrectora de Formación en la Universidad de las Ciencias Informáticas, investigadora en el área de Inteligencia Artificial; natalia@uci.cu.

AGRADECIMIENTOS

A mis padres por ser las personas más comprensivas del mundo, por su amor inagotable, su confianza y su apoyo incondicional.

A toda mi familia, a mi abuela, a todos mis tíos por su gran apoyo.

A las personas que me han brindado su amistad.

A todos mis compañeros del edificio 106 que tanto pasamos juntos.

A las personas que me ayudaron en mi preparación durante la carrera, a los profesores y a los trabajadores del Departamento de Informática y Electrónica del Equipo de Servicios de Traductores e intérpretes.

A mis tutoras Natalia y Lidice por su generosidad al brindarnos la oportunidad de recurrir a su capacidad y experiencia en un marco de confianza, afecto y amistad.

En general a todas y cada una de las personas que han vivido la realización de este trabajo por habernos brindado su apoyo, colaboración, ánimo y sobre todo cariño y amistad.

DEDICATORIA

A mis padres, que siempre me han apoyado y que nunca han dejado de confiar en mí. A ellos que siempre están ahí, en las malas y en las buenas; por la educación y consejos sabios ofrecidos en los momentos oportunos.

A mi novio que me ha soportado siempre, por todos los momentos vividos que ha formado parte de las páginas de mi vida.

A mis amigos que siempre han estado cuando les he necesitado y nunca me han dado la espalda.

A mis familiares y a todos aquellos que de una forma u otra han estado presentes durante el desarrollo de mi vida.

RESUMEN

El Equipo de Servicios de Traductores e intérpretes es una entidad, que posee una infraestructura con un alto número de recursos tecnológicos para el funcionamiento del centro. En la entidad se decidió crear un equipo asignando a varios informáticos para atender problemas puntuales y así; ofrecer un mejor servicio a todos los departamentos de la entidad.

El presente trabajo tiene como objetivo, desarrollar una aplicación web que permita la gestión de las incidencias reportadas de tal forma que se garantice brindar un mejor servicio y la distribución de la fuerza de trabajo asociada a esta. En el contexto empresarial cubano, gestionar incidencias y destinar el personal encargado de solucionarlas, constituyen operaciones que se vuelven cada vez más complicadas. Esto implica gran cantidad de tiempo en realizar dichas acciones. Escasea su gestión y de existir, la información se registra en formato duro, o digital como Word u hoja de cálculo. Teniendo como problemática la situación anteriormente planteada se obtiene como resultado una aplicación que facilita esta gestión tecnológica y el procesamiento de datos que esta conlleva para que se convierta en un producto sostenible.

Para el diseño e implementación del sistema web, se optó por la utilización de la metodología OpenUp, se utilizó la herramienta Visual-Paradigm para el modelado. Como lenguaje de programación Python, Java Script y Django; Visual Studio Code como entorno de desarrollo y SQLite como sistema gestor de base de datos. Como resultado, se obtuvo un sistema capaz de gestionar las incidencias tecnológicas, generados dentro de la entidad.

PALABRAS CLAVE

Recursos tecnológicos, aplicación web, reporte de incidencias.

ABSTRACT

The Translators and Interpreters Services Team is an entity that has an infrastructure with a high number of technological resources for the operation of the center. In the entity, it was decided to create a team assigning several IT specialists to deal with specific problems and thus offer a better service to all the entity's departments.

The objective of this work is to develop a web application that allows the management of reported incidents in such a way as to guarantee a better service and the equitable distribution of the workforce associated with it. In the Cuban business context, managing incidents and allocating the personnel in charge of solving them, constitute operations that become increasingly complicated. This implies a large amount of time in carrying out said actions. Its management is scarce and if it exists, the information is recorded in a hard format, or digital such as Word or spreadsheet. Considering the aforementioned situation as a problem, the result is an application that facilitates this technological management and the data processing that this entails so that it becomes a sustainable product.

For the design and implementation of the web system, the use of the OpenUp methodology was chosen, the Visual-Paradigm tool was used for modeling. As a programming language Python, Java Script and Django; Visual Studio Code as development environment and SQLite as database management system. As a result, a system capable of managing technological incidents generated within the entity was obtained.

KEYWORDS

Technological resources, web application, incident report.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	6
I.1 Conceptos fundamentales.....	6
I.2 Metodología de desarrollo de software.....	8
I.2.1 Proceso Unificado Abierto (OpenUp).....	8

I.3 Sistemas gestión de incidencias tecnológicas.....	10
I.3.1 Herramientas para la gestión de incidencias tecnológicas.....	10
I.3.4 Análisis de los sistemas de los sistemas homólogos.....	13
I.4 Lenguajes de programación y herramientas utilizadas en el desarrollo de la solución	14
I.4.1 Lenguaje de Modelado.....	14
I.4.2 Lenguaje de Programación.....	16
I.4.3 Framework.....	18
I.4.4 IDE de desarrollo.....	19
I.4.5 Sistema de gestión de base de datos.....	19
Conclusiones del capítulo.....	20
CAPÍTULO II: ANALISIS Y DISEÑO DE LA PROPUESTA SOLUCION.....	21
II.1 Propuesta Solución.....	21
II.1.1 Modelo conceptual propuesto.....	21
II.2 Propuesta del sistema.....	23
II.2.1 Personas relacionadas con el sistema.....	24
II.2.2 Especificación de los requisitos de software.....	24
II.2.3 Diagrama de caso de uso del sistema.....	29
II.2.4 Descripción textual de los Caso de uso.....	30
II.3 Diseño.....	35
II.3.1 Arquitectura.....	36
II.3.2 Patrones Arquitectónico.....	37
II.3.3 Patrones de Diseño.....	39
II.3.4 Diagrama de clases de diseño.....	40
II.3.5 Diagrama de interacción.....	41
II.4 Diseño de Base de Datos.....	43
Conclusiones del capítulo.....	44
CAPÍTULO III: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	45
III.1 Modelo de Despliegue de la solución.....	45
III.2 Pruebas de software.....	46
III.2.1 Diseño de casos de pruebas.....	47
III.2.2 Pruebas para requisitos no funcionales.....	49
III.3 Métodos y Técnicas.....	50
III.4 Prueba de rendimiento.....	52
III.5 Prueba de seguridad.....	52
.....	53
Conclusiones del capítulo.....	54
CONCLUSIONES FINALES.....	55

RECOMENDACIONES.....	56
REFERENCIAS BIBLIOGRÁFICAS.....	57
ANEXOS.....	60

ÍNDICE DE TABLAS

Tabla 1: Personas relacionadas con el sistema (elaboración propia).....	24
Tabla 2: Requisitos funcionales del sistema (elaboración propia).....	25
Tabla 3: Requisitos no funcionales del sistema (elaboración propia).....	28
Tabla 4: Especificación Caso de uso Autenticar Usuario (elaboración propia).....	30
Tabla 5: Especificación Caso de uso Gestionar Incidencia (elaboración propia).....	32
Tabla 6: Especificación Caso de uso Actualizar estado de la incidencia (elaboración propia).	34
Tabla 7: Descripción de entidades de la base de datos (elaboración propia).....	44
Tabla 8: Variables del caso de prueba Autenticar usuario (elaboración propia).....	47
Tabla 9: Matriz de caso de prueba Autenticar usuario (elaboración propia).....	47
Tabla 10: Variables del caso de prueba Gestionar incidencias (elaboración propia).....	48
Tabla 11: Matriz de caso de prueba Gestionar incidencia (elaboración propia).....	48
Tabla 12: No conformidades por Caso de uso (elaboración propia).....	51

ÍNDICE DE FIGURAS

Ilustración 1 Fases del OpenUP (14).....	10
Ilustración 2 Interfaz de GATServer.....	13
Ilustración 3 Modelo conceptual de la propuesta solución (elaboración propia).....	22
Ilustración 4 Diagrama de caso de uso del sistema (elaboración propia).....	30
Ilustración 5 Arquitectura Cliente-Servidor(34).....	36
Ilustración 6 Patrón Arquitectónico MTV (elaboración propia).....	38
Ilustración 7 Diagrama de paquetes para la solución propuesta (elaboración propia).....	39
Ilustración 8 Empleo de los patrones de diseño de nuestra aplicación.....	40
Ilustración 9 Diagrama de clases del diseño Caso de uso Gestionar Incidencia. Escenario Adicionar incidencia (elaboración propia).....	41
Ilustración 10 Diagrama de secuencia Caso de uso Gestionar Incidencia. Escenario Adicionar incidencia (elaboración propia).....	42
Ilustración 11 Diagrama de secuencia Caso de uso Gestionar Incidencia. Escenario Eliminar incidencia (elaboración propia).....	42
Ilustración 12 Diagrama de secuencia Caso de uso Actualizar estado de incidencia (elaboración propia).....	43
Ilustración 13 Diagrama entidad-relación (elaboración propia).....	43
Ilustración 14 Diagrama de despliegue (elaboración propia).....	46
Ilustración 15 Captura de pantalla del sistema (autenticar usuario) (elaboración propia).....	53
Ilustración 16 Captura de pantalla del sistema (pantalla principal) (elaboración propia).....	53
Ilustración 17 Captura de pantalla del sistema (pantalla para reportar incidencia) (elaboración propia).....	54

OPINIÓN DEL(OS) TUTOR(ES)

<Contenido de la opinión de los tutores>

AVAL DEL CLIENTE

<Contenido del aval del cliente sobre la solución desarrollada>

INTRODUCCIÓN

La evolución de las Tecnologías de la Información (TI) han traído consigo un progreso tanto económico como social a nivel mundial. El uso de estas tecnologías, es cada vez más creciente imponiéndose en todas las ramas de la sociedad. Su indetenible avance es evidente, así como su notable impacto, al punto que se puede considerar la base fundamental del desarrollo de la humanidad. Estas tecnologías contribuyen al mejoramiento de las empresas; elevar la calidad, control y productividad son solo algunos de los pasos para favorecer la obtención de procesos eficientes.

En el contexto actual, existen empresas que han establecido sistemas y herramientas de gestión que brindan asistencias de conocimiento de la información. Las mejores prácticas logradas a nivel mundial para este tipo de modelos de gestión de incidencias, están basadas en los estudios ITIL (Information Technology Infrastructure Library). **ITIL** es un conjunto de conceptos y mejores prácticas referentes a la gestión de servicios de TI que describe detalladamente un extenso conjunto de funciones y procesos ideados para ayudar a las organizaciones a lograr calidad y eficiencia en las operaciones de TI (1).

El uso de TI es parte de la cotidianidad de las personas. Las TI juegan un papel fundamental para las organizaciones hoy en día, para que estas puedan competir en el mercado global (2). Las empresas de estos tiempos necesitan agilidad en los negocios e interactividad con los clientes para así solucionar las necesidades o incidencias que surgen en el día a día, es por esta razón que buscan dinamismo productivo, por lo que realizan sus operaciones y funciones de la mano de la tecnología, empleando sistemas, internet y otras actividades en función de las TI. Es por ello, que la TI y en particular el soporte técnico tiene como función resolver y atender solicitudes de los usuarios finales, las cuales pueden ser incidencias de tipo hardware o software de manera rápida y oportuna para no retrasar las labores cotidianas de los trabajadores. En ese sentido la utilización de un plan de dirección de las incidencias hará que se sistematice los procesos, esto permitirá registrar todas las solicitudes de incidencias tecnológicas y cumplir en atender la totalidad de estas incidencias.

Cuba no está ajena a estos procesos que brindan asistencia a la TI, tanto así que algunas de las empresas destinadas al desarrollo del software tienen su propio sistema para la gestión de estos procesos. La Empresa Nacional de Software (DESOFT) creó el sistema AvilaQuid que actualmente se encuentra funcionando en varias entidades y empresas de nuestro país brindando servicios de soporte. La Universidad de las Ciencias Informáticas (UCI) con varios sistemas, entre los que se destaca el GATServer.

La UCI, centro de estudios superiores, que abrió, sus puertas al primer curso escolar el 23 de septiembre de 2002, como fruto de la visión futurista del Comandante en Jefe Fidel Castro Ruz (3). Se ha convertido en un centro de altos estudios que tiene como misión formar profesionales comprometidos con su Patria y altamente calificados en la rama de la informática. Se encarga de realizar aplicaciones y sistemas informáticos, a partir de la vinculación estudio-trabajo como modelo de formación y sirve de soporte a la industria cubana de la informática.

No solo las instituciones dedicadas a la industria del software brindan las herramientas necesarias para la gestión de incidencias, sino que las propias entidades se pueden encargar de desarrollar lo que necesitan para asegurar el correcto funcionamiento de sus herramientas de trabajo. Tal es el caso del Equipo de Servicios de traductores e intérpretes (ESTI) el cual posee una estructura que le permite concebir una aplicación web que se encargue de la gestión de las incidencias en la entidad.

El ESTI creado a partir del Acuerdo No. 306, adoptado por el Consejo de Ministros de la República de Cuba, ante la necesidad de disponer de un colectivo de traductores e intérpretes calificados que pudieran prestar estos servicios en cualquier tipo de eventos nacionales o internacionales. Los antecedentes del ESTI se remontan a apenas tres años después del Triunfo de la Revolución, cuando se crea el sistema de becas y se inicia la formación de traductores e intérpretes en el Instituto "Pablo Lafargue", fundado con un grupo de estudiantes del naciente Instituto "Máximo Gorki", concebido para formar docentes. Se prepararon entonces, durante un corto período de tiempo, los primeros especialistas en ruso, inglés, alemán, francés y checo. Años más tarde se comenzarían a formar traductores e intérpretes en 13 idiomas. El ESTI concentra su atención en las actividades de mayor rigor profesional, como reuniones internacionales que se efectuaban en Cuba y en otros países; las misiones del nivel central y la atención que en este perfil de trabajo requería la alta dirección del país (4).

Esta entidad presupuestada administrativa bajo la subordinación del Ministerio de Relaciones Exteriores controla varios procesos administrativos, entre estos se encuentran la gestión de incidencias tecnológicas. El control de incidencias tecnológicas consiste en todos aquellos fenómenos que puedan ocurrirle a la tecnología de la institución; por lo que se hace necesario el mantenimiento y disponibilidad de sistemas críticos es fundamental para obtener el éxito en cualquier organización moderna y para la productividad del personal que los utilizan. Para garantizar que esto suceda, su equipo de soporte técnico de TI debe responder de modo eficaz e inmediato a los problemas que puedan surgir.

En el estudio realizado en la entidad se detectó que el departamento Informático y Técnico electrónico se divide en dos áreas, una de desarrollo de software y otra de soporte técnico. El área de soporte técnico se encarga de gestionar los ciber incidentes de los usuarios finales. El problema en el departamento de servicios técnicos es que los incidentes informáticos se reportan de forma manual; a través de llamadas telefónicas, correos electrónicos y en otras ocasiones de forma verbal con el servicio de atención a los usuarios; estas incidencias reportadas se registran de forma manual en hojas de papel o en el mejor de los casos en Microsoft Word debido a esto no se lleva un buen control, por cuanto no se logran registrar la totalidad de las incidencias, por lo tanto no puede brindar atención a todos los requerimientos formulados provocando una baja productividad del personal de soporte y una mala opinión de los trabajadores que solicitan gestionar sus incidencias.

El procedimiento actual trae como consecuencia que:

- ✓ No se conocen con claridad la cantidad de detalles de roturas de equipamiento.

- ✓ La búsqueda, análisis y actualización de la información se realiza de forma engorrosa, lenta y propensa a errores.
- ✓ Dificultad para controlar el cumplimiento del plan de trabajo individual de los técnicos.
- ✓ Los tiempos de respuesta a los reportes de incidencias recibidos no cumplen con las expectativas de los usuarios.

Teniendo cuenta lo expuesto anteriormente, se plantea como **problema** la siguiente interrogante: ¿Cómo contribuir a la gestión de las incidencias tecnológicas de equipos de cómputo y sistemas informáticos en el Equipo de Servicios para Traductores e intérpretes?

El Objeto de estudio de la presente investigación va dirigido al proceso de gestión de reportes de incidencias tecnológicas basado en aplicaciones web, enmarcando como **campo de acción**:

Aplicación web para la automatización de la gestión de incidencias tecnológicas en el ESTI.

Para dar solución al problema planteado se define como **objetivo general**: Desarrollar una aplicación web que contribuya al proceso de gestión de incidencias tecnológicas de mantenimiento en el ESTI.

Para dar cumplimiento a lo planteado en el objetivo general, es necesario dar cumplimiento a las siguientes **tareas de la investigación**:

- ✓ Sistematización de los referentes teóricos que sustentan la investigación, relacionados con el uso de las herramientas informáticas en función de la gestión de reportes de incidencias.
- ✓ Estudio del estado del arte sobre las herramientas y tecnologías actuales de desarrollo de software para seleccionar las más factibles a usar en este trabajo.
- ✓ Estudio y análisis de los procesos que se llevan a cabo en la gestión de incidencias.
- ✓ Confección del diseño de la solución para los procesos de gestión de incidencias, así como implementación de la solución diseñada.
- ✓ Implementación de la solución propuesta.
- ✓ Estudio y análisis de la factibilidad de la solución obtenida.
- ✓ Diseño de los casos de pruebas de las soluciones implementadas.
- ✓ Validación, a través de una estrategia de pruebas, de la Aplicación web para la gestión de incidencias tecnológicas en el ESTI.

Para obtener los conocimientos necesarios que hagan posible el cumplimiento del objetivo trazado, se lleva a cabo una investigación en las que se utilizan algunos de los métodos científicos existentes, tanto teóricos como empíricos.

Teóricos:

Histórico-Lógico: Permite una mayor comprensión de la evolución de las herramientas informáticas para la gestión de reportes de incidencias de mantenimiento.

Analítico-Sintético: Se utiliza con el objetivo de realizar un análisis bibliográfico para establecer las bases teóricas en relación al desarrollo de los sistemas para la gestión de reportes de incidencias de mantenimiento.

Modelación: Es empleada en la representación mediante diagramas de las características, procesos y componentes de la solución propuesta, así como la relación existente entre ellos.

Empíricos:

Entrevista: Se emplea en encuentros con el cliente para conocer la necesidad del desarrollo de la propuesta de solución, definir sus funcionalidades e identificar a la vez particulares de cada usuario y las restricciones que se imponen (ver Anexo 1).

Observación: Posibilita obtener conocimiento acerca del funcionamiento de los sistemas existentes en la actualidad para la gestión de reportes de incidencias de mantenimiento.

La presente investigación está estructurada en tres capítulos, los cuales se explican brevemente a continuación.

Capítulo I: Fundamentación teórica.

Se hace una descripción de los conceptos relacionados con las aplicaciones web para la gestión de incidencias tecnológicas, las tendencias, técnicas, tecnologías, metodologías y el análisis de algunas soluciones existentes a nivel internacional y nacional que son de interés por tener estrechos lazos con el alcance de esta investigación. Se define además el lenguaje de programación, el entorno de desarrollo para la implementación de la solución propuesta.

Capítulo II: Análisis y diseño de la propuesta solución.

Características y diseño del sistema, explicando la propuesta de solución al problema de investigación, se definen los principales requisitos a partir tomando como base la metodología seleccionada y se describen los diagramas de clases del diseño, diagramas de comportamiento (diagramas de secuencia), lo referente al estilo arquitectónico del sistema, así como los patrones de diseño utilizados y el entorno de despliegue propuesto para desplegarlo.

Capítulo III: Validación de la solución propuesta.

En este capítulo se detalla la propuesta de solución al problema planteado. Se define el entorno de despliegue necesario para el sistema y se describe la organización del mismo en un diagrama de componentes. Se especifican los estándares de codificación que se utilizan en base al lenguaje de programación propuesto para el desarrollo del sistema. Además, se realiza la estrategia de pruebas definida para validar la propuesta de solución verificando el cumplimiento de los requisitos funcionales y no funcionales descritos en el capítulo anterior y se muestran imágenes de interfaces del sistema como parte del resultado final.

Con la realización de la presente investigación, se pretende obtener un sistema para la gestión de incidencias tecnológicas que permitirá, al Departamento de Informática y Técnicos electrónicos del ESTI, ejecutar los procesos que se desarrollan durante la gestión de incidencias tecnológicas que se realicen desde cada departamento de la entidad. Además de proveer a los usuarios de una herramienta para registrar y supervisar sus reportes sobre cualquier incidencia tecnológica, con el fin de que estas sean solucionadas eficientemente. El sistema facilitará el manejo de toda la información referente a las incidencias y mostrará el estado real del trabajo que se realiza por parte de Departamento de Informática y Técnicos electrónicos del ESTI.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se hace un estudio de los elementos teóricos que sirven de base a la investigación del problema planteado; del estado del arte relacionado con soluciones informáticas existentes para la gestión de reporte de incidencias para mantenimiento; se caracterizan las tendencias actuales, el uso de la metodología. Además, se seleccionan el lenguaje de modelado, el lenguaje de programación y las herramientas empleadas en el desarrollo de la solución.

I.1 Conceptos fundamentales

A continuación, se relacionan los principales conceptos que ayudan a entender el desarrollo de la investigación.

Las aplicaciones web o también conocidos como “sistemas web” son aquellos sistemas que están creados e instalados no sobre una plataforma o sistema operativos, sino que se aloja en un servidor en internet o sobre una intranet (red local). Su aspecto es muy similar a páginas web que pueden ser vistas normalmente, pero en realidad las aplicaciones web tienen funcionalidades potentes que brindan respuestas a casos particulares, entre los que se encuentran los sistemas de gestión(5).

Gestión:

Procesos, acciones y tareas que se llevan a cabo sobre un conjunto de elementos (personas, procedimientos, estrategias, planes, recursos, productos) para lograr el éxito sostenido de una organización, es decir, disponer de los recursos necesarios para satisfacer las necesidades y las expectativas de sus clientes o beneficiarios, trabajadores y de otras partes interesadas a largo plazo y de un modo equilibrado y sostenible (6). Dentro de la gestión de procedimientos, destacan los sistemas de gestión para incidencias.

Gestión de incidencias:

La gestión de Incidencias es un proceso que tiene como objetivo optimizar y resolver incidentes de una manera óptima y sencilla, también es cualquier suceso que cause una interrupción, demora del servicio que se brinda al usuario. Es así como en Cuba, se busca la satisfacción y calidad hacia el usuario final aplicando un conjunto de buenas prácticas por la cual se emplea ITIL estos están destinados a optimar la gestión de los servicios de TI brindando calidad y satisfacción al usuario (7).

La gestión de incidencias es un conjunto de acciones para anticipar, resolver y documentar eventos no planificados en una organización. Su objetivo es guiar a los profesionales a través de una situación inesperada y ayudarles a volver a la normalidad lo más rápido posible.

En otras palabras, la gestión de incidencias es un plan utilizado para detectar, comunicar y solucionar un incidente. Incluye las herramientas y pasos necesarios para manejar emergencias con la mayor efectividad posible.

Elementos a tener en cuenta para el procesamiento de gestión de incidencias:

- Tipo de incidencia.
- Breve descripción de lo ocurrido.

- Establecer nivel prioridad de la incidencia.
- Establecer responsable de dar solución al problema.
- Monitorear los estados por los que esta transcurre la incidencia.
- Fecha de detección
- Fecha de asignación
- Fecha de resolución.
- En el caso que no se pueda resolver, establecer los motivos que conllevan a este resultado.

Elementos que se deben cumplir para monitorear las incidencias

- **Monitorear las incidencias antes, durante y después que ocurran.**
- **Documentar** las incidencias y los responsables por detectarlas; las acciones correctoras y los responsables por aplicarlas.
- **Analizar** el informe de seguimiento.
- **Actuar** sobre la información recopilada y adoptar medidas concretas para evitar las incidencias.
- **Revisar** los resultados obtenidos y hacer los cambios necesarios.

Reporte:

Documento que a partir de los objetivos de quien lo crea, pretende transmitir información. Estos pueden ser de diversos tipos o estar soportados en diversos formatos (documentos impresos, digitales, audiovisual). En el ámbito de la informática, los reportes son informes que se organizan y muestran información a partir de fuentes de datos digitales diversas (textos, bases de datos). Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios (8). Constituye una de las principales actividades para documentar la información de cualquier empresa, principalmente en el aspecto de las incidencias porque permite registrar a detalles los acontecimientos ocurridos durante cierto periodo de tiempo.

Incidencia:

En terminologías de TI, “una incidencia es cualquier acontecimiento que tenga una influencia negativa sobre

una organización, incluidos su personal, el producto de la organización, los equipos o el entorno en el que opera. También puede ser definida como “los sucesos que provocan la degradación o pérdida del funcionamiento normal de un servicio” (9).

El término “incidencia” es utilizado para hacer referencia a distintas situaciones. Puede referirse a un hecho que acontece mientras está ocurriendo un negocio u otra situación, relacionada con ello. En un contexto informático es asociado usualmente con cualquier mal funcionamiento de los sistemas de hardware y software. Se considera para esta investigación que una incidencia es: “Cualquier evento que no forma parte de la operación estándar de un servicio y que causa, o puede causar, una interrupción o una reducción de calidad del mismo”. Para no llegar a estos eventos se hace necesario brindar servicios de mantenimientos para que el flujo de trabajo sufra las menores afectaciones posibles.

Mantenimiento:

La realización de tareas o actividades que permitan reparar una unidad funcional para que ésta pueda continuar cumpliendo sus objetivos, e incluyen acciones de inspección, comprobaciones, clasificación, reparación, entre otra (10).

I.2 Metodología de desarrollo de software

El término **metodología** se define como el **grupo de mecanismos o procedimientos racionales**, empleados para el logro de un objetivo, o serie de objetivos que dirige una investigación científica (11). Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas, herramientas y soporte documental para el desarrollo de un producto (software). Se basan en una combinación de los modelos de proceso genéricos (cascada, evolutivo e incremental). Son para estructurar, planear y controlar el proceso de desarrollo del software. Constituyen una guía que define las tareas y actividades que se deben realizar para obtener un software de buena calidad(12).

Una metodología de desarrollo de software indica paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben tener. Es un proceso de software detallado y completo. Se encarga de:

1. Proporcionar la tarea de planificación.
2. Facilitar la tarea del control y seguimiento de un proyecto.
3. Mejorar la relación coste / beneficio.
4. Optimizar el uso de recursos disponibles.
5. Posibilitar la evaluación de los resultados y cumplimiento de los objetivos.
6. Posibilitar la comunicación efectiva entre usuarios y desarrolladores.

La metodología a emplear en el desarrollo en el presente trabajo es OpenUp que es una metodología ágil que se basa en Rational Unified Process (RUP).

I.2.1 Proceso Unificado Abierto (OpenUp)

OpenUp es un proceso modelo y extensible, que mantiene las mismas características que RUP, orientado a la gestión y desarrollo de proyectos de software basados en desarrollo iterativo, ágil e incremental adecuados para proyectos pequeños y de bajos recursos, validos a un conjunto muy vasto de plataformas de desarrollo (13). Inspirado en la metodología de RUP, es un subconjunto de la misma que contiene el conjunto mínimo de prácticas que apoyan a un equipo de desarrollo de software a realizar un producto de alta calidad. Con una

filosofía ágil se centra en la naturaleza de colaboración con procesos de desarrollo de software. Entre sus principales características se encuentra su alto nivel de adaptabilidad a las necesidades del proyecto.

Open UP nos ofrece una metodología ágil y flexible, que se puede acoplar a la mayoría de proyectos, además que cubre aspectos como la seguridad y contratación de personal, también incluye a otras personas interesadas en el proyecto o parte de.

Como metodología de desarrollo es conducida por el principio de colaboración para alinear intereses y compartir su comprensión. Es el desarrollo unificado que aplica acercamientos a interactivos e incrementales dentro del ciclo de vida del software. OpenUp es completa en el sentido de que manifiesta por completo el proceso de construir un sistema. Para atender las necesidades que no están cubiertas en su contenido OpenUp es extensible a ser utilizado como base sobre la cual se pueden añadir o adaptarse a contenido de otro proceso que sea necesario.

Características de Open Up

- Desarrollo incremental
- Uso de caso de uso y escenarios
- Manejo de riesgos
- Diseño basado en la arquitectura

Ventajas

- Es una metodología ágil
- Se puede adaptar con otros procesos

Desventajas

- Se espera que cubra un amplio sistema de necesidades para los proyectos de desarrollo en un plazo muy corto.

El ciclo de vida del proyecto provee a los interesados un mecanismo de supervisión y dirección para controlar los fundamentos del proyecto, su ámbito, la exposición a los riesgos, el aumento de valor y otros aspectos. El OpenUp estructura el ciclo de vida de un proyecto en cuatro fases: inicio, elaboración, construcción y transición (13).

Concepción: primera fase en el proyecto del ciclo de vida, acerca del entendimiento del propósito y objetivos obteniendo suficiente información para confirmar que el proyecto debe hacer. El objetivo de esta fase es capturar las necesidades de los stakeholder (clientes) en los objetivos del ciclo de vida para el proyecto.

Elaboración: se trata los riesgos significativos para la arquitectura. El propósito de esta fase es establecer la base la elaboración de la arquitectura del sistema.

Construcción: esta fase está enfocada al diseño, implementación y prueba de las funcionalidades para desarrollar un sistema completo. El propósito de esta fase es completar el desarrollo del sistema basado en la arquitectura definida.

Transición: es la última fase, cuyo propósito es asegurar que el sistema es entregado a los usuarios, evalúa la funcionalidad y rendimiento del último entregable de la fase de construcción.

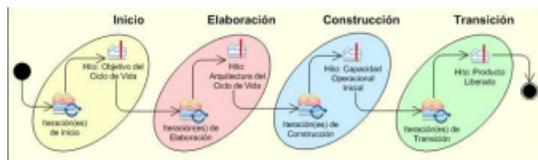


Ilustración 1 Fases del OpenUP (14)

I.3 Sistemas gestión de incidencias tecnológicas

La gestión de incidencias tecnológicas es una de las muchas tareas que se realizaban en formato duro, ya que no existía una herramienta capaz de realizar dicho proceso. En la actualidad con el avance de las tecnologías, cada vez son más las actividades que van ganando terreno debido a las facilidades que nos brindan las aplicaciones que son capaces de gestionarlos automáticamente o que facilitan la generación de los mismos. Estos programas, permiten obtener la información deseada con mayor rapidez ofreciéndoles a los usuarios un mejor nivel de detalle y flexibilidad; además brindan la posibilidad de interactuar con los registros realizados y saber qué estado tiene su reporte en cada momento.

I.3.1 Herramientas para la gestión de incidencias tecnológicas

La investigación sobre los Sistemas de gestión de incidencias mostró que existe una tendencia creciente de su utilización en la actualidad. En la industria del software ya son cada vez más las herramientas que se encargan de estos procesos. Al realizar un análisis sobre estos sistemas a nivel internacional y nacional, se encontraron diversas soluciones, a continuación, se ofrece una breve descripción de algunas de ellas.

Sistemas a nivel internacional.

Protecnus

Es un software de mantenimiento dirigido a todas aquellas empresas con servicio técnico de campo, donde podrán gestionar y planificar de manera fácil y *online* toda la actividad de la empresa, desde sus técnicos hasta el análisis de datos y gestión empresarial. Posee un módulo de escritorio y un módulo móvil. Permite conocer el estado de los trabajos e incidencias en tiempo real. Localiza a sus técnicos en el momento. Al final del día la información es procesada automáticamente. A pesar de estas ventajas, no permite la obtención de reportes que necesitan los directivos y, además, es un software privativo (15).

OTRS

Es un sistema online de gestión de tickets y procesos, moderno y flexible, que permite a los profesionales de la gestión de servicios, de cualquier industria, mantenerse al día con el entorno empresarial actual orientado a resultados. OTRS está implantado en empresas de todos los tamaños. Desde departamentos de IT de empresas de tamaño medio a grandísimas compañías y organizaciones como el Instituto Nacional de Estadística, Boeing, Porsche, Lu-

fthansa, Nokia o la NASA. Entre sus principales características, y lo que la hace una herramienta muy útil, se encuentran la gestión y monitorización automática de incidencias, la existencia de una aplicación de OTRS para iPhone, y la generación de reportes para la toma de decisiones. Sin embargo, es un sistema privativo que no permite el acceso a su código fuente (16).

KMKey Help Desk:

Knowledge Management Key (KMKey) Help Desk es un software de gestión de incidencias indicado para servicios de mantenimiento, ayuda al usuario y resolución de problemas en cualquier sector. Permite definir flujos de trabajo para abordar problemáticas derivadas de anomalías en servicios y maquinaria. La incidencia puede recibirse de forma automática (e-mail, entrada a través de una web, desde un dispositivo móvil) o bien ser abierta por el servicio de atención. Una vez en marcha seguirá el flujo diseñado por el cliente para su resolución (17).

Este sistema no se ajusta a las necesidades del cliente pues abarca otras áreas, además de la gestión de incidencias, que no son de interés. Está enfocado en minimizar los tiempos de respuesta y brindar servicios de mantenimiento, lo cual sacrifica el correcto seguimiento de la incidencia procesada, uno de los requerimientos principales exigidos por el cliente.

La incidencia puede recibirse de forma automática (e-mail, entrada a través de una web, desde un dispositivo móvil) o bien ser abierta por el servicio de atención. Una vez en marcha seguirá el flujo diseñado por el cliente para su resolución. Permite: resolución inmediata, escalado, consulta de información anterior, reparto de recursos. No permite la generación de reportes estadísticos que necesitan los directivos para la toma de decisiones. Otra desventaja es su licencia privativa, la cual no permite tener acceso al código fuente de esta herramienta ni su uso libre (17).

Suite CRM (Módulo de gestión de incidencias)

Es una solución con la que se puede trabajar eficientemente la recogida y tratamiento de quejas. Normalmente, la entrada de las quejas se realiza por vía telemática o por centralita. En este sentido, cada operador accede a la plataforma con su cuenta de usuario. En caso de recibir una incidencia a través de un correo electrónico o formulario de la web, toda esta información se traspasa de forma automática.

Una vez registrada la incidencia, éste deberá pasar por todos los estados y validadores antes de considerarse como cerrada. El siguiente paso a la validación del informe es la emisión del informe de respuesta al cliente. Esta comunicación, puede realizarse por diferentes canales (correo electrónico, teléfono, o por el canal que utilice la empresa). Se engloba dentro de los softwares Open Source, lo cual significa que el código fuente del programa es editable. A pesar de estas ventajas, no permite la obtención de reportes estadísticos que necesitan los directivos para la toma de decisiones (18).

Sistemas a nivel nacional.

AvilaQuid:

AvilaQuid es una aplicación web (desarrollada en la provincia de Ciego de Ávila por la empresa DESOFT), destinada a la automatización de la gestión de incidencias. Este objetivo lo satisface atractiva y rápidamente, facilitando el seguimiento de todas las incidencias generadas por los clientes tanto externos como internos, a través de toda su entidad o grupo de entidades interconectadas en una Intranet, basándose principalmente en la agrupación de las incidencias en función de las Unidades Organizativas que intervienen en su gestión. Para ello, basa su funcionamiento en el tránsito de toda incidencia por los siguientes pasos: su registro; su clasificación por diversos aspectos y asignación de responsabilidad de atención; y su procesamiento que permite indicar el estado de la gestión, la conclusión y la respuesta, llegando hasta la posible aplicación de medidas. Precisamente en esta característica el sistema apoya también la protección de la información, diferenciando los accesos de los usuarios de acuerdo a sus funciones, las cuales se corresponden con estos pasos. Con ello y con la distribución de información por Unidades Organizativas se logra que no todo usuario

pueda modificar toda información, y a la vez, que todo usuario sí pueda consultar toda la información que le concierne. Esenciales ventajas de AvilaQuid resultan el acceso total y pleno que proporciona en cualquier momento a la información que fluye, acorde a las Unidades Organizativas; al igual que la inmediatez y alcance que se logra en cuanto a la disponibilidad de la información actualizada, que se tiene de forma única y congruente. Además, se puede resaltar su flexibilidad al permitir configurar todos los tipos de datos que se manejan respecto a las incidencias, incluso, Ud. puede manejar diferentes tipos de incidencia con la misma aplicación; pudiéndose establecer así el funcionamiento del sistema acorde a los intereses y forma de trabajo de la entidad (19). En estos momentos se encuentra en uso en varias empresas e instituciones de nuestro país, por ejemplo, ETECSA, Correos de Cuba, Inmunología Molecular, DESOFT y Asambleas del Poder Popular.

CuCERT:

La gestión de incidentes de ciberseguridad en el país no es una actividad nueva, se realiza a partir de un sistema de trabajo que involucra a un grupo de entidades especializadas en su detección y respuesta, con alcance a las personas jurídicas, y para lo cual se constituyó el Centro a respuesta de incidentes (Cucert), insertado en la estructura de la Oficina para la Seguridad de las Redes Informáticas (OSRI) (20) Entidad nacional adscrita al Ministerio de la Informática y las Comunicaciones que lleva a cabo la prevención, evaluación, aviso, investigación y respuesta a las acciones, tanto internas como externas, que afecten el normal funcionamiento de las TI de Cuba. Su misión es la de prevenir y responder a los incidentes computacionales en el país. Dentro de sus funciones se encuentran recibir los reportes y responder los incidentes computacionales que se presenten en el país. También proporciona informes acerca de vulnerabilidades y amenazas sobre sistemas cubanos. Educar a la comunidad en general sobre temas de seguridad informática, así como la detección con rapidez de las incidencias computacionales son otras de sus metas.

GATServer

Es una aplicación web desarrollada por el Grupo de Asistencia Técnica (GAT) de la UCI para la gestión de las afectaciones tecnológicas, que permite la gestión de reportes para las afectaciones de medios computacionales, redes y telefonía. La aplicación permite insertar reportes de roturas y generar las soluciones de los mismos, pero al ser una aplicación web propia, adaptada a su función en el centro no es una solución óptima para la DSTI UCI pues en la misma se necesita una aplicación que se ajuste al flujo de sus procesos internos y que responda a las necesidades existentes, además debe ser una aplicación escalable que permita la posible integración futura con el resto de los sistemas contables usados en la misma.

The screenshot shows the GATServer web application interface. At the top, there is a navigation bar with the title 'GATServer' and a search bar. Below the navigation bar, there is a section titled 'Mis Solicitudes' with a 'Filtrar solicitud' button. The main content is a table with the following columns: #, Creado, Venice, Local, Cliente, Equipo, Prioridad, Telefono, Problematika, Descripción, and Hora. The table contains several rows of data representing service requests.

#	Creado	Venice	Local	Cliente	Equipo	Prioridad	Telefono	Problematika	Descripción	Hora
12194	3/8/2014 2:33:44 PM	-1188	Rectorado - Asistente de Vicerectoría de Programas Especiales	138911 / maia	Fotocopiadora	Crítica	8219	Defectuosa	no	De 8:00 am -5:00 pm
12093	3/7/2014 8:34:42 AM	-1241	Rectorado - Grupo de Economía de Inversiones	x108584 / joia salvaopaz	Unidad Central	Crítica	8204	Configuración de	es tiene r...	De 8:00 am -5:00 pm
12052	8/14/2014 11:43:41 AM	-447	Docente 2 (SAGE) - Oficina de Trámites de cooperación Internacional y Relaciones Públicas	148236 / maia	Impresora	Normal	8273	Defectuosa	es impime	De 8:00 am -5:00 pm
12474	3/18/2014 9:26:36 AM	-501	Rectorado - Sala de Monitoreo de Cámaras	104843 / pider	Televisor	Baja	3573	Defectuosa	fallado	De 8:00 am -5:00 pm
12105	3/8/2014 8:52:10 AM	-1218	Rectorado - Atención de Protección	521198 / maia	Monitor	Baja	8220	Cable de alimentación	problema e...	De 8:00 am -5:00 pm
11375	6/13/2014 9:17:30 AM	-1721	Docente 2 (SAGE) - Asistente de Vicerectoría Económica	101859 / adios	Unidad Central	Baja	8231	No tiene red		De 8:00 am -5:00 pm
11283	6/12/2014 11:19:25 AM	-1743	Rectorado - Dirección de prebendo	138994 / maia maia	Unidad Central	Baja	8205	Defectuosa	señal de	De 8:00 am -5:00 pm
11057	6/26/2014 1:34:13 PM	-1573	Rectorado - Asistente de Vicerectoría de Producción	131957 / hector garcia	Teclado	A	848854	Defectuosa		De 8:00 am -5:00 pm

Ilustración 2 Interfaz de GATServer

I.3.2 Análisis de los sistemas de los sistemas homólogos

El análisis realizado sobre los sistemas de gestión de incidencias, emitió realizar el siguiente resumen:

- El sistema Protecenus es adaptable a dispositivos móviles, pero es privativo y no genera reportes.
- El sistema OTRS es adaptable a dispositivos móviles y genera reportes, pero no se tiene acceso a su código fuente.
- El sistema Suite CRM es público y genera reportes, sin embargo, no es adaptable a dispositivos móviles.
- El sistema CuCERT es privativo y no tiene acceso a su código fuente.
- El sistema AvilaQuid es privado y no genera reportes.

Después de realizar un análisis de los sistemas homólogos internacionales y nacionales que gestionan incidencias, se decide implementar un nuevo sistema, pues ninguno de estos sistemas cumple con todos los requerimientos deseados por el cliente. Algunos sistemas por poseer un carácter privativo no resultan soluciones factibles dado que no se puede modificar su código fuente para adecuarlo a las necesidades crecientes del Centro de Soporte. Otros están enfocados para satisfacer necesidades de las organizaciones para las cuales fueron desarrollados.

I.4 Lenguajes de programación y herramientas utilizadas en el desarrollo de la solución

Para el desarrollo de una aplicación web es sumamente importante una buena selección del lenguaje de programación y de las herramientas a utilizar ya que estas en muchos casos determinan el tiempo de desarrollo de nuestro proyecto, los costos y el resultado a general de los procesos.

I.4.1 Lenguaje de Modelado

El lenguaje de modelado es cualquier lenguaje informático gráfico o textual que provee el diseño y construcción de estructuras y modelos siguiendo un conjunto sistemático de reglas y marcos (21). Un lenguaje de modelado visual se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Con ellos es posible diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas.

I.4.1.1 Lenguaje de Modelado Unificado (UML)

El Lenguaje de Modelado Unificado (en inglés *Unified Model Language*), incluye conceptos semánticos, notación y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. UML es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño. Existen diferencias importantes entre un método y un lenguaje de modelado. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código, así como, generadores de informes. La especificación de UML no define un proceso estándar, pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos (22). Es una consolidación de muchas de las notaciones y conceptos más usados en la modelación orientada a objetos. Es importante recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, no es un proceso, sino un lenguaje que permite la modelación de sistemas, con tecnología orientada a objetos. Desde el año 1995, UML es un estándar aprobado por la ISO como ISO/IEC 19501:2005 *Information Technology*; está respaldado por el *Object Management Group* (OMG).

Posibilita la modelación del ciclo completo de desarrollo de software y contiene: Los Diagramas de casos de uso, clases, objetos, secuencia, colaboración, estado, actividades, despliegue y componentes.

Los principales beneficios de UML son:

- I. Mejorar tiempos total de desarrollo (en 50% o más).
- II. Modelar sistemas utilizando conceptos orientados a objetos.
- III. Establecer conceptos y artefactos ejecutables.
- IV. Orientar el desarrollo del ascenso en sistemas complejos de misión crítica.
- V. Establecer un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- VI. Mejor soporte a la planeación y al control de proyectos
- VII. Alta reutilización y minimización de costos.

I.4.1.2 Herramienta CASE

Las herramientas CASE, Ingeniería de Software Asistida por Computadoras (por sus siglas en inglés *Computer Aided Software Engineering*) son un conjunto de aplicaciones y ayudas que propician el desarrollo de programas informáticos a los desarrolladores e ingenieros de software. Las mismas automatizan diversas tareas que se realizan durante el ciclo de vida de desarrollo del software, tales como: la documentación, la generación de código y diseños, las pruebas de errores y la gestión del proyecto. Facilita la estandarización de la documentación y el perfeccionamiento del software. Las herramientas CASE son diferentes aplicaciones informáticas destinadas a elevar la productividad en el desarrollo de software minimizar el costo de las mismas en términos de tiempo y de dinero. Estas herramientas son de mucha ayuda en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores (22). Dentro de las múltiples herramientas Case en el mercado, la Universidad de las Ciencias Informáticas utiliza Visual Paradigm.

I.4.1.2.1 Visual Paradigm 8.0

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Además, la herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto y permite control de versiones. Cabe destacar igualmente su robustez, usabilidad y portabilidad (23).

¿Por qué Visual Paradigm?

Esta herramienta es estable en cuanto a su ejecución en diferentes sistemas operativos y la facilidad de abrir y trabajar con un modelo. Es una herramienta que guarda todo el modelo en un solo fichero por lo que basta con copiarse solo ese fichero para estar seguro de que tiene todo el trabajo encapsulado en él. Además, el equipo de desarrollo presenta conocimientos básicos de la herramienta lo que posibilita un mejor desempeño con la misma. Finalmente, Visual Paradigm es una herramienta bastante ligera permitiendo su uso en ordenadores poco potentes.

1.4.2 Lenguaje de Programación

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; permitiendo al desarrollador comunicarse con los dispositivos de hardware y software existentes.

1.4.2.1 JavaScript 1.5

JavaScript es un lenguaje de programación interpretado, ligero y orientado a objetos, se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. A pesar de su nombre, no guarda ninguna relación directa con el lenguaje de programación Java, tienen semánticas y propósitos diferentes. Los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Java es un lenguaje de programación con el que se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del *Document Object Model* (DOM) (24).

JavaScript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, con funciones y estructuras de datos complejas. Además, pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente. Este lenguaje por sus características y posibilidades permite un desarrollo completo de cualquier aplicación web por compleja que sea, facilita el manejo de información en el navegador sin necesidad de realizar llamadas redundantes a una base de datos.

1.4.2.2 Python 3.9

Python es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma. Este lenguaje forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional (25).

¿Por qué Python?

Es un lenguaje de alto nivel que contiene implícitas algunas estructuras de datos como listas, diccionarios, conjuntos y tuplas, que permiten realizar algunas tareas complejas en pocas líneas de código y de manera legible. Python ha ido ganando en terreno en comunidades como la de software libre, científica y educacional, por su sencillez y posibilidad de concentrarse en los problemas actuales.

1.4.2.3 CSS 3

Las Hojas de Estilos en Cascada (CSS, por sus siglas en inglés) es un lenguaje creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas Web complejas. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien formateados y con significado completo (también llamados documentos semánticos). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferente. Al crear una página Web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, entre otros. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, además de otras funcionalidades.

1.4.2.4 HTML 5

De las siglas de (*Hyper Text Markup Language*) Lenguaje de Marcas de Hipertexto, es el lenguaje de marcado más utilizado para la construcción de páginas web. Es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la síntesis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica como desplegar el contenido del documento, incluyendo texto imágenes y otros medios soportados. Su versión 5 brinda más facilidades que sus antecesores, lo que permite el desarrollo de la aplicación con una mayor vistosidad, calidad e integración, este es soportado por las versiones actuales de los navegadores Mozilla Firefox, Chrome, Chromiun, Safari e Internet Explorer (26).

1.4.3 Framework

Bootstrap

Bootstrap es un framework basado en HTML y CSS, creado por Twitter y liberado en 2012. Desde entonces ha ganado muchos adeptos, hasta el nivel de contar con la mayor comunidad de Github del mundo. Este framework ayuda a agilizar la creación de la interfaz de nuestra página web. Su uso permite que el sitio web sea adaptable a la pantalla del dispositivo con el que se accede, ya sea un ordenador, tablet, smartphone, televisión. Esto significa que tendremos una web sensible o adaptativa (27).

Django

Django es un marco de trabajo (framework) para el desarrollo de aplicaciones web usando Python. Django considera algunas funcionalidades listas para usar para facilitar el desarrollo de aplicaciones web. Como resultado, no es necesario escribir todo el código ni usar tiempo para buscar errores de código en el framework. Es decir, mediante Django, el desarrollo de sistemas de información web puede ser rápido, seguro, escalable y también fácil de mantener.

Django es un framework web de alto nivel de código abierto, que respeta el patrón de diseño conocido como modelo–vista–controlador (MVC). Permite el desarrollo rápido de sitios web seguros y mantenibles. La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes; proporciona una serie de características que facilitan el desarrollo rápido de páginas orientadas a contenidos.

Este framework proporciona una aplicación incorporada para administrar los contenidos, que puede incluirse

como parte de cualquier página hecha con Django y que puede administrar varias páginas a partir de una misma instalación; la aplicación administrativa permite la creación, actualización y eliminación de objetos de contenido, llevando un registro de todas las acciones realizadas sobre cada uno, y proporciona una interfaz para administrar los usuarios y los grupos de usuarios.

¿Por qué Django?

Django es un framework muy productivo y con características de muy alto nivel, como un ORM (Object to Relational Mapper), y un sistema de plantilla muy útil a la hora de elaborar formularios.

I.4.4 IDE de desarrollo.

Una de las decisiones más importantes a la hora de desarrollar en Django que es el lenguaje de programación a utilizar, es el IDE a usar, ya que el entorno de desarrollo puede suponer una verdadera diferencia de tiempo de trabajo.

I.4.4.1 Visual Studio Code.

Es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para depuración, control de Git integrado, resaltado de sintaxis, finalización de código inteligente, fragmentos de código y refactorización de código. También es personalizable, de modo que los usuarios pueden cambiar el tema del editor, los métodos abreviados de teclado y las preferencias. Es gratuito y de código abierto.

¿Por qué Visual Studio Code?

Porque es una herramienta que tiene soporte nativo para gran variedad de lenguajes, entre ellos podemos destacar los principales del desarrollo Web: HTML, CSS, y JavaScript, entre otros.

- Brinda la posibilidad de configurar la interfaz a nuestro gusto. De esta forma, podremos tener más de un código visible al mismo tiempo, las carpetas de nuestro proyecto y también acceso a la terminal o un detalle de problemas, entre otras posibilidades.
- Por la existencia de una amplísima gama de temas o estilos visuales para Visual Studio Code, que hacen el trabajo con el software más agradable a la vista.

I.4.5 Sistema de gestión de base de datos.

Un Sistema Gestor de Bases de Datos (SGBD) se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos (28). Este permite la extracción, almacenamiento y modificación de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Proporcionan funciones con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos. Permiten que la información de la base de datos sea almacenada en varios formatos. La mayoría incluyen un generador de informes. También pueden incluir un módulo gráfico que permita presentar la información con gráficos y tablas.

PostgreSQL 10.5

PostgreSQL es un Sistema Gestor de Bases de Datos Relacionales de código abierto y orientado a objetos. Brinda un control de concurrencia multi-versión, que permite trabajar con grandes volúmenes de datos. Funciona en todos los sistemas operativos Linux, UNIX y Windows.

El código fuente se encuentra disponible para todos sin costo alguno. Debido a la liberación de la licencia, PostgreSQL se puede usar, modificar y distribuir con fines privativo, comercial, así como académico. Por estas características es considerado como uno de los gestores de bases de datos más avanzado del mundo. Se trabajará con PostgreSQL Server por ser muy usado en entornos de software libre y puede funcionar en múltiples plataformas.

Herramienta para prueba de software I.4.6

Las pruebas de software son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada (29). En el caso de la propuesta de solución de la investigación, este será validado a través de varios tipos y técnicas de pruebas, algunas de ellas de forma manual y otras mediante el uso de herramientas que permiten la realización de dicha tarea. Entre las herramientas existentes, se pretende utilizar las herramientas Acunetix Web Vulnerability Scanner y Apache JMeter, para las pruebas de seguridad y de carga y estrés, respectivamente.

Acunetix realiza automáticamente auditorías a aplicaciones web comprobando vulnerabilidades de Inyección SQL, Cross site scripting y otras vulnerabilidades que puedan ser explotadas por atacantes.

Apache JMeter es una herramienta diseñada para pruebas de estrés en aplicaciones web simulando las funcionalidades de un navegador o de cualquier otro cliente. Muestra los resultados de las pruebas en una amplia variedad de informes y gráficas, con gran cantidad de variables que permiten interpretar los resultados desde diferentes puntos de vista.

Conclusiones del capítulo

Los conceptos asociados al dominio del problema enmarcado en el capítulo, enriquecieron el conocimiento sobre la gestión de incidencias y las actividades que se realizan en este sentido. De igual forma, aunque las plataformas similares estudiadas no satisfacen en su totalidad las necesidades de la entidad, el estudio de las mismas permitió entender cómo funcionan los sistemas de gestión de incidencias tecnológicas y aportaron a la selección de algunas características importantes para el desarrollo de la aplicación web. Además, la metodología OpenUp permitió guiar el proceso de desarrollo de software basándonos en las fases por la que transita la misma.

CAPÍTULO II: ANÁLISIS Y DISEÑO DE LA PROPUESTA SOLUCIÓN

Luego de elaborar el marco teórico y seleccionar las herramientas y la metodología de desarrollo de software a utilizar, se han creado las condiciones necesarias para realizar la propuesta solución que satisfaga las necesidades del problema de investigación. En este capítulo se describe aspectos importantes del análisis y diseño para la implementación de la aplicación web y tiene como objetivo describir la propuesta que se planea para establecer la solución. Para describir la solución se realiza el modelo conceptual definiendo los diversos elementos que serán manejados, así como las relaciones que entre estos se establecen. Se define lo que debe hacer la aplicación a través de los requerimientos funcionales y no funcionales, se realiza el diagrama de caso de uso del sistema, así como las descripciones de los casos de uso del mismo.

II.1 Propuesta Solución.

Para complementar el entendimiento del problema se hace uso del modelo conceptual, aunque también la metodología OpenUp nos propone artefactos en este sentido, es flexible y puede contrastarse con algunos que simplifiquen y apoyen el proceso, que esclarezcan los conceptos referentes al problema antes mencionado.

Un modelo conceptual tiene como objetivo identificar y explicar los conceptos significativos en un dominio de problema, identificando los atributos y las asociaciones existentes entre ellos. Puede ser visto, también como una representación de las cosas, entidades, ideas, conceptos u objetos del "mundo real" o dominio de interés (30).

Se representan en UML con un diagrama de clases en el que se muestra conceptos u objetos del dominio del problema (clases conceptuales), asociaciones entre las clases conceptuales y atributos de estas. No muestra comportamiento y las clases conceptuales no pueden ser métodos. Una clase conceptual puede ser una idea o un objeto físico.

II.1.1 Modelo conceptual propuesto.

Para lograr un mejor entendimiento de los procesos que requieren informatización, se realizó un modelo conceptual (ver ilustración 3), el cual recoge y describe los conceptos más importantes dentro del contexto del sistema, así como las relaciones entre ellos.

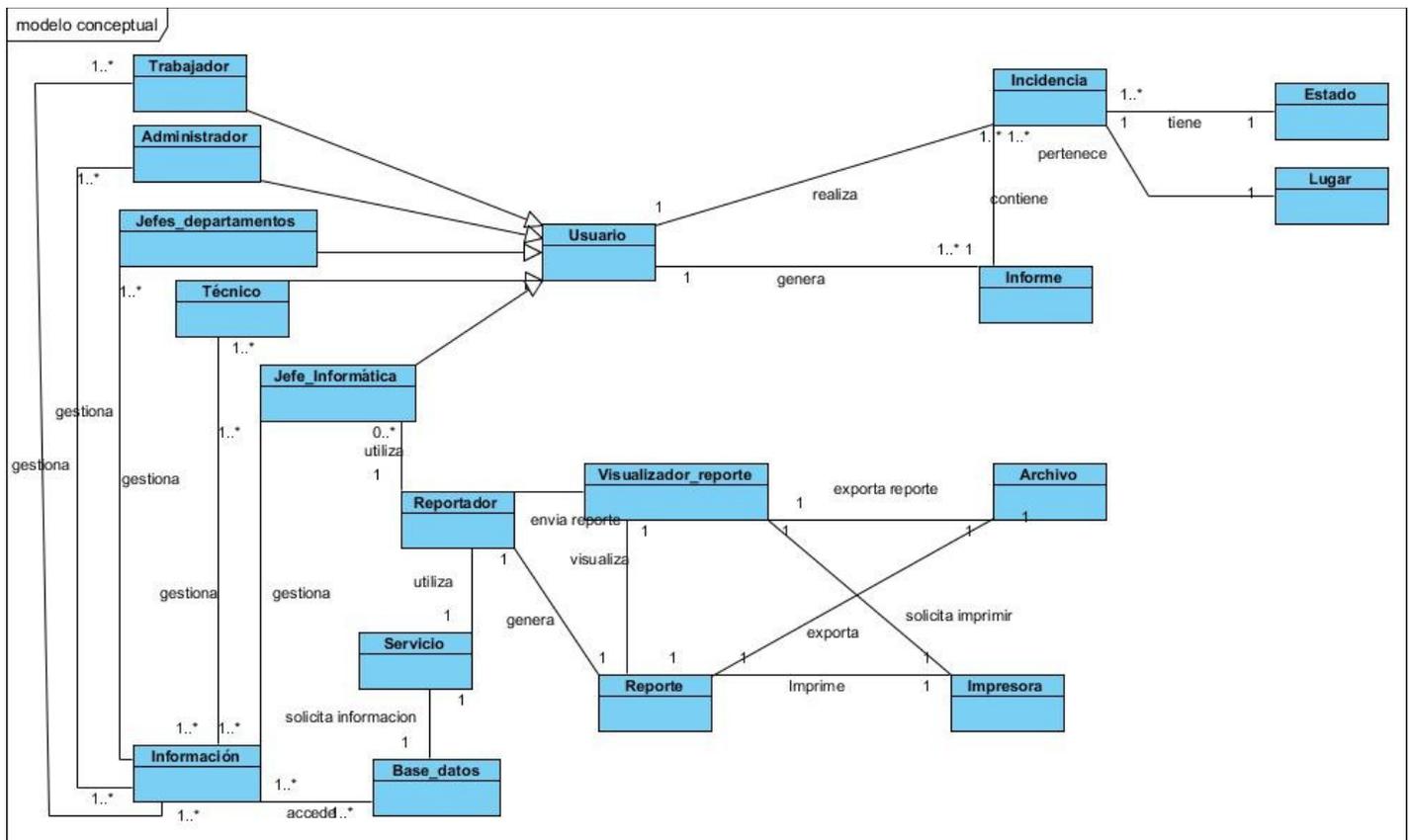


Ilustración 3 Modelo conceptual de la propuesta solución (elaboración propia)

Definición de los conceptos del Diagrama de Modelo de Dominio

Usuario: Hace referencia a los distintos roles que pueden acceder al SRIT. Cualquier persona que necesite obtener o diseñar incidencias a la base de datos.

Trabajador: Persona que necesite obtener o diseñar incidencias a la base de datos.

Técnico: Persona que recibe los reportes que le son asignados por el jefe de informática y le da solución a los mismos.

Administrador: Persona encargada de administrar toda la información del SRIT.

Jefe_informática: jefe del departamento informático de la entidad, encargado de asignar a los técnicos las incidencias recibidas y realizar reportes.

Jefes_departamento: jefes de los diferentes departamentos de la entidad, diseña reportes a la base de datos y recibe información de las incidencias reportadas en su departamento.

Incidencia: Es una incidencia realizado por un usuario sobre cualquier situación fuera de lo normal en un equipo determinado. Permite informar el estado técnico de un equipo determinado.

Informe: Resumen de reportes, quejas y equipos por rango de fecha. Es una ficha que se generará en el

sistema con todos los reportes registrados en la fecha determinada.

Estado: Define los estados que puede tener un reporte.

Lugar: Área donde se encuentra la afectación contenida en el reporte.

Reporte: Información de interés para los usuarios.

Reportador: Aplicación visual para crear modelos de reportes y a partir de estos generarlos.

Servicio: Sistema que permite la comunicación entre la clase Reportador y la Base de datos.

Visualizador_reporte: Interfaz donde se muestra una vista previa del reporte generado.

Archivo: Objeto donde se guarda la información solicitada.

Base_datos: contiene toda la información de interés para el usuario.

Impresora: Dispositivo en el cual se puede imprimir un reporte creado.

II.2 Propuesta del sistema

Teniendo en cuenta los requerimientos planteados y como medio de cumplimiento de los objetivos propuestos inicialmente se ha concebido como propuesta de solución a la problemática la implementación de la aplicación web que permita la gestionar las incidencias reportadas por los usuarios de la entidad que se le brinda servicios y distribución de la fuerza de trabajo asociada a este. Con el objetivo de lograr una gestión más completa de incidencias. El sistema brinda la posibilidad de gestionar los usuarios y los permisos de estos, según el rol asignado es el acceso a los datos que posea.

El sistema proporcionará una interfaz web como forma de representación al usuario, la cual tendrá una página principal que ofrecerá información general sobre el sistema y desde la cual los usuarios podrán acceder a la página de autenticación. El usuario autenticado puede ser: de tipo trabajador, técnico, jefe_informática, jefes_departamentos y administrador.

Una vez autenticado el usuario, el sistema le proporcionará de acuerdo a su rol diferentes secciones en el sistema, ciertos privilegios y permisos que le permitirán tener acceso a diferentes funcionalidades del sistema. El sistema incluirá todo el proceso de gestión de incidencias de rotura de equipamiento; cada usuario podrá generar informes donde se recoja información en dependencia del nivel de acceso que tenga. De manera general el sistema permitirá al administrador la gestión de incidencias, roles, estados y usuarios.

En resumen, con la propuesta de página web se facilita la centralización, manejo y control de la información del proceso de gestión de incidencias tecnológicas en el ESTI de una forma organizada.

II.2.1 Personas relacionadas con el sistema.

Como elemento indispensable a tener en cuenta cuando se comienza el desarrollo de un sistema informático es delimitar la audiencia a la cual va dirigido el mismo. Teniendo en cuenta que la misma a su vez puede ser dividida en grupos atendiendo a sus necesidades.

Tabla 1: Personas relacionadas con el sistema (elaboración propia)

Usuarios relacionados con el sistema	
Reportador(trabajador)	Es la persona que tiene la posibilidad de crear un reporte de rotura llenando los datos requeridos, así

	como consultar información y estadísticas de sus reportes realizados.
Administrador	Es la persona facultada para la gestión del sistema en general. Podrá gestionar la administración de usuarios, equipos y demás elementos del sistema.
Jefe de informática	Es la persona que podrá acceder a los reportes realizados y asignar la responsabilidad del mismo a técnico que se encargará de darle solución. Además, podrá consultar información y estadísticas de los reportes en general
Técnico	Es la persona que podrá acceder a los reportes que le son asignados por el Jefe de informática, así como cambiarle el estado a los mismos una vez solucionados. Además, podrá consultar información.
Jefes de los diferentes departamentos	Es la persona que podrá acceder a los reportes realizados en su departamento, además podrá consultar información acerca de los mismos.

II.2.2 Especificación de los requisitos de software.

Los requisitos del software en general son condiciones o capacidades que necesitan los usuarios para resolver un problema o alcanzar un objetivo. Estos requisitos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar información. Los requisitos tratan de establecer lo que el sistema debe de hacer, sus propiedades emergentes deseadas, esenciales y las restricciones en el funcionamiento del sistema y los procesos de desarrollo de software. Los mismos se pueden clasificar en requisitos funcionales y no funcionales. A continuación, se muestra la captura realizada para el desarrollo de la solución propuesta.

Requisitos funcionales

Los requisitos funcionales son declaraciones de las funcionalidades que debe cumplir el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares (29). La aplicación a desarrollar a través de la realización del presente trabajo debe cumplir los siguientes requisitos:

Tabla 2: Requisitos funcionales del sistema (elaboración propia)

No.	Requisitos Funcionales
1	Autenticar usuario. El sistema debe permitir al usuario autenticarse en la intranet introduciendo su usuario y contraseña.
2	Insertar usuario. El sistema debe permitir al administrador crear usuarios en la base de datos teniendo en cuenta los siguientes campos: nombre, apellidos, género, cargo, usuario, contraseña, departamento, teléfono, is_super_usuario, is_tecnico, is_jefeArea y is_jefeInf para así poder autenticarse por primera vez en el sistema.
3	Modificar Usuario. El sistema debe permitir al administrador modificar el perfil de un usuario para asignar o quitar los roles y permisos de los usuarios autenticados.
4	Eliminar usuario.

	El sistema debe permitir al usuario administrador eliminar un usuario existente en la base de datos.
5	Insertar incidencia. El sistema debe permitir a los usuarios autenticados crear una incidencia tecnológica. Debe introducir los siguientes datos: nombre, tipo de incidencia, departamento y descripción.
6	Modificar incidencia. El sistema debe permitir a los usuarios autenticados de trabajador o jefe de departamento modificar una incidencia tecnológica cambiando los siguientes datos: nombre, tipo de incidencia, departamento y descripción.
7	Eliminar incidencia. El sistema debe permitir a los usuarios autenticados con el rol de trabajador eliminar su incidencia.
8	Mostrar listado de incidencias. El sistema debe permitir al usuario autenticado con el rol trabajador mostrar una lista con todos sus reportes realizados de incidencias tecnológicas.
9	Realizar notificaciones de incidencias. El sistema debe notificar mediante correo electrónico, al trabajador acerca del estado de su incidencia.
10	Crear un reporte de incidencias tecnológicas. El sistema debe permitir al usuario autenticado con el rol administrador, crear un reporte de incidencias tecnológicas con los siguientes datos: título, descripción, departamento, equipo o sistema a reportar, departamento.
11	Exportar reporte de incidencias tecnológicas. El sistema debe permitir al usuario autenticado con el rol administrador exportar un listado con los reportes de incidencias tecnológicas.
12	Mostrar listado de incidencias de su departamento por el jefe de departamento. El sistema debe permitir al usuario autenticado con el rol jefe de departamento mostrar un listado de las incidencias tecnológicas creadas en su departamento.
13	Modificar reporte de incidencia por el administrador. El sistema debe permitir al usuario autenticado con el rol administrador editar el estado a los reportes de incidencias tecnológicas.
14	Eliminar reporte de incidencia por el administrador. El sistema debe permitir al usuario autenticado con el rol administrador eliminar uno o más reportes de tipo tecnológicos de ser necesario.
15	Mostrar un reporte de incidencia por el administrador. El sistema debe permitir al usuario autenticado con el rol administrador mostrar un reporte de una incidencia tecnológica.
16	Mostrar listado de incidencias por el administrador. El sistema debe permitir al usuario autenticado con el rol administrador mostrar un listado con los reportes de incidencias tecnológicas.
17	Asignar responsabilidad de incidencia por el administrador. El sistema debe permitir al usuario autenticado con el rol administrador asignar responsabilidad de las incidencias tecnológicas generadas.
18	Mostrar listado de todas las incidencias reportadas por los usuarios al administrador. El sistema debe permitir al usuario autenticado con el rol administrador mostrar un listado con todos los reportes de incidencias tecnológicas generados por los trabajadores.
19	Realizar notificaciones de incidencias tecnológicas. El sistema debe notificar mediante correo electrónico, a los trabajadores del centro cuando cambia el estado de sus reportes tecnológicos y al administrador y técnico cuando tiene un nuevo reporte por atender.

20	<p>Modificar incidencia por el técnico.</p> <p>El sistema debe permitir al usuario autenticado con el rol técnico editar el estado a los reportes de incidencias tecnológicas.</p>
21	<p>Eliminar incidencia por el técnico.</p> <p>El sistema debe permitir al usuario autenticado con el rol técnico eliminar uno o más reportes de tipo tecnológicos de ser necesario.</p>
22	<p>Mostrar listado de incidencias por el técnico.</p> <p>El sistema debe permitir al usuario autenticado con el rol técnico mostrar un listado con los reportes de incidencias tecnológicas</p>
23	<p>Auto asignarse responsabilidad de atender una incidencia.</p> <p>El sistema debe permitir al usuario autenticado con el rol técnico asignar responsabilidad de las incidencias tecnológicas generadas.</p>
24	<p>Mostrar listado de todas las incidencias reportadas por los usuarios al técnico.</p> <p>El sistema debe permitir al usuario autenticado con el rol técnico mostrar un listado con todos los reportes de incidencias tecnológicas generados por los trabajadores.</p>
25	<p>Mostrar un listado de las incidencias asignadas.</p> <p>El sistema debe permitir al usuario autenticado con el rol técnico mostrar un listado con las incidencias tecnológicas que le fueron asignadas.</p>
26	<p>Imprimir informe de reporte.</p> <p>El sistema debe permitir al usuario autenticado con el rol de jefe de informática imprimir el informe de incidencias reportadas en el sistema.</p>
27	<p>Exportar informe de reporte.</p> <p>El sistema debe permitir al usuario autenticado con el rol de jefe de informática exportar el informe de incidencias reportadas en el sistema.</p>
28	<p>Mostrar reporte realizado por fecha.</p> <p>El sistema debe permitir al usuario autenticado con el rol de jefe de informática ver el informe de incidencias reportadas en el sistema por fecha.</p>
29	<p>Mostrar reporte realizado por estado.</p> <p>El sistema debe permitir al usuario autenticado con el rol de jefe de informática ver el informe de incidencias reportadas en el sistema por estado.</p>
30	<p>Mostrar reporte realizado por usuario.</p> <p>El sistema debe permitir al usuario autenticado con el rol de jefe de informática ver el informe de incidencias reportadas en el sistema por usuarios.</p>
31	<p>Graficar cantidad de reporte por fecha</p> <p>El sistema debe permitir al usuario autenticado con el rol de jefe de informática graficar las incidencias reportadas por fechas.</p>

Requisitos no funcionales.

Los requisitos no funcionales representan características generales y restricciones de la aplicación o sistema que se esté desarrollando. Son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. Suelen expresarse de una manera general y sin hacer referencia a algún módulo, transacción o característica del sistema (31).

Tabla 3: Requisitos no funcionales del sistema (elaboración propia)

No	Requisitos no funcionales
1	<p>Interfaz</p> <p>El diseño de las interfaces que se propone para el sistema a implementar está basado en la</p>

	premisas de la organización y sencillez de la información a mostrar.
2	<p>Software</p> <p>Para el despliegue del sistema se debe contar en el servidor de aplicaciones web con Python y Django y el servidor de base de datos SQLite.</p> <p>Máquina del Servidor</p> <ul style="list-style-type: none"> • Sistema operativo: El servidor debe correr en sistemas operativos Windows, Unix y Linux • Servidor web: • Lenguaje Python versión 9 o superior. • Gestor de base de datos: PostgreSQL <p>Máquina del trabajador</p> <ul style="list-style-type: none"> • Sistema operativo: Windows 7 o superior. • Navegador web: Firefox versión 25.0 o superior, Google Chrome 27.0 o superior, Internet Explorer 7.0 o superior. <p>La comunicación entre el cliente y el servidor de aplicaciones web se realiza a través del protocolo https.</p>
3	<p>Hardware</p> <p>La máquina servidor deberá contar con un microprocesador multinúcleo i3 de segunda generación o superior, 8gb de RAM, un mínimo de 500gb de disco duro</p> <p>Las estaciones de trabajo cliente debe tener como mínimo 2gb de RAM y un procesador a 1.80Ghz dual core.</p>
4	<p>Seguridad</p> <p>Según las características que debe cumplir un sistema para garantizar su seguridad informática, acceder a la información autorizada de acuerdo al rol de cada usuario en la aplicación web asegura la confidencialidad, evita que la información sea modificada garantizando la integridad y el acceso pleno de cada usuario con facultades para el uso de la aplicación las 24 horas del día establece la disponibilidad.</p> <p>La información manejada por el sistema debe ser objeto de una cuidadosa protección contra la corrupción y estados de inconsistencia.</p>
5	<p>Capacidad</p> <p>Considerando las características técnicas mínimas para la ejecución en trabajadores. Soportar hasta 100 conexiones simultáneas. Considerar el crecimiento esperado en el volumen de datos.</p>
6	<p>Eficiencia</p> <p>Rendimiento en tiempo de respuesta menor a que 5 segundos.</p>
7	<p>Usabilidad</p> <p>El sistema deberá está diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en tiempo reducido.</p> <p>Facilidad de uso por parte de los desarrolladores de software.</p> <p>El sistema debe ser fácil de extender por parte de los desarrolladores.</p>

II.2.3 Diagrama de caso de uso del sistema.

Los diagramas de caso de uso brindan un panorama bastante sencillo de una interacción, de modo que usted tiene que ofrecer más detalle para entender lo que está implicado. Este detalle puede ser una simple descripción textual, o una descripción estructurada en una tabla o un diagrama de secuencia, como se discute a continuación. Es posible elegir el formato más adecuado, dependiendo del caso de uso y del nivel de detalle que usted considere se requiera en el modelo (32).

El caso de uso captura un contrato que describe el comportamiento del sistema en diferentes condiciones mientras este responde a la petición de uno de sus usuarios. A continuación, se muestra el Diagrama de Casos de Uso del Sistema (DCUS) para la propuesta de solución.

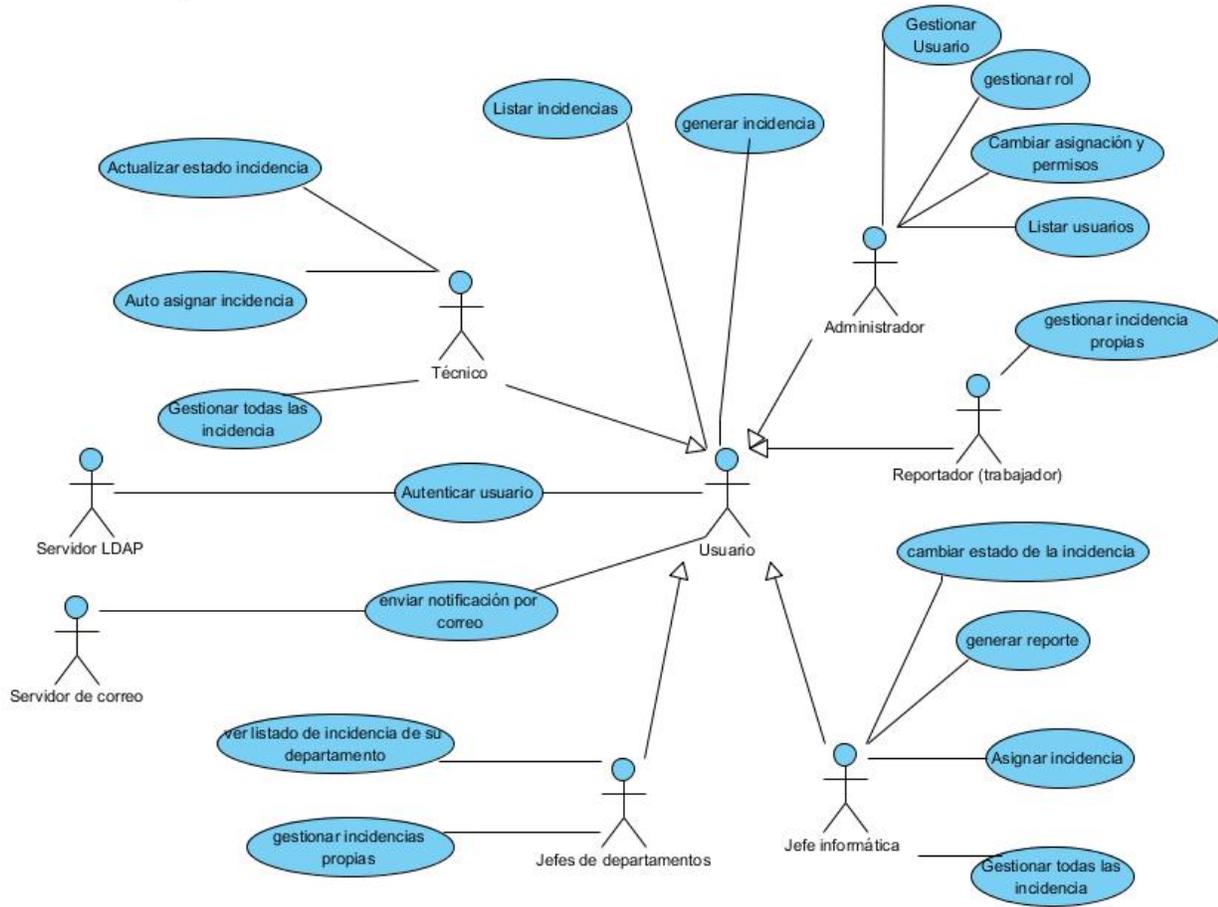


Ilustración 4 Diagrama de caso de uso del sistema (elaboración propia)

II.2.4 Descripción textual de los Caso de uso.

A continuación, se especifican los casos de usos más relevantes del sistema.

Tabla 4: Especificación Caso de uso Autenticar Usuario (elaboración propia)

Caso de Uso del Sis- Autenticar Usuario tema

Objetivo	Permitir entrar al sistema.
Actores	Usuario

Resumen	<i>El caso de uso se inicia cuando el usuario accede al sistema y procede a autenticarse utilizando su usuario y contraseña. El usuario tiene la posibilidad de utilizar una contraseña local y su contraseña del dominio, siempre mediante el usuario de dominio ESTI.</i>
Complejidad	<i>Media</i>
Prioridad	<i>Alta</i>
Precondiciones	<i>La cuenta de usuario debe estar habilitada como cuenta local por uno de los administradores del sistema.</i>
PostCondiciones	<i>El usuario accede mediante su sección a la interfaz de autenticación del sistema.</i>

Acción del actor	Respuesta del Sistema
1. El usuario accede al sistema	2. El sistema muestra el formulario de autenticación solicitándose usuario y contraseña.
3. el usuario introduce su usuario de dominio ESTI, la contraseña y define si el acceso será local o mediante el dominio	4. El sistema verifica el tipo de acceso del usuario. 4.1 El sistema verifica para el usuario local si la contraseña coincide con la almacenada. 4.2 El sistema verifica para el acceso mediante dominio que la contraseña coincida con la contraseña obtenida para el usuario.
	5. El sistema muestra la interfaz de acceso al usuario logueado.

Flujos alternativos El sistema verifica que el usuario este registrado en el sistema y verifica que la contraseña coincida con la almacenada.

Validación de usuario

Acción del actor	Respuesta del Sistema
	El sistema muestra un mensaje indicando “El usuario introducido no está registrado en el sistema” o “la contraseña es incorrecta”

Prototipo de Interfaz



Bienvenido

yazy-semi-Admin

.....

Recuerdame

Autenticarse

Tabla 5: Especificación Caso de uso Gestionar Incidencia (elaboración propia)

Caso de Uso del Sis- Gestionar incidencia tema

Objetivo	Permitir añadir, modificar, listar y eliminar incidencias
Actores	Usuario
Resumen	El caso de uso se inicia cuando el jefe de informática o el técnico precisa gestionar una incidencia generada a partir de la solicitud de un usuario; o cuando el trabajador precisa gestionar sus incidencias. El usuario puede adicionar una incidencia, donde debe especificar la descripción de la incidencia y los requisitos que esta necesita. El usuario puede modificar dichos datos o eliminar el reporte siempre que su estado sea pendiente a asignación o asignado.
Complejidad	Alta
Prioridad	Alta
Precondiciones	El usuario debe estar autenticado y con permisos de gestionar incidencias
PostCondiciones	Se adiciona un nuevo reporte y queda asignado a un técnico, se modifica o elimina

Acción del actor	Respuesta del Sistema
------------------	-----------------------

- | | |
|---------------------------------------|---|
| 1. El usuario accede mediante la fun- | 2. El sistema muestra la lista de incidencias |
|---------------------------------------|---|

cionalidad Incidencias

3. El usuario selecciona la opción de adicionar, modificar o eliminar.

Sección: Adicionar Incidencia

3. El usuario selecciona la opción de adicionar incidencia.

5. El usuario introduce los datos necesarios para reportar una incidencia.

4. El sistema muestra un formulario solicitando los datos de la incidencia a reportar.

6. El sistema verifica si los datos están correctos. En caso de faltar algún dato se pasa al flujo alternativo A, en caso de que el reporte ya exista se pasa al flujo alternativo B.

7. El sistema determina para la incidencia la fecha, la hora, el usuario y el estado.

8. Registra la nueva incidencia.

9. Muestra un mensaje indicando el éxito de la operación.

9.1 El sistema envía correo al jefe de informática y a los técnicos informando la creación de una nueva incidencia.

Flujo alternativo A

El sistema señala los campos vacíos.

Flujo alternativo B

Muestra el mensaje "La incidencia está definida y se encuentra registrada".

Sección Modificar Incidencia

3. El usuario selecciona la opción de adicionar incidencia.

4. El sistema muestra un formulario

5. El sistema muestra un formulario con los datos de la incidencia

6. El sistema verifica si los datos están correctos, en caso de faltar algún dato se pasa al flujo alternativo A.

7. El sistema determina para la incidencia la fecha, la hora, el usuario y el estado, modificando los datos anteriores.

8. Modifica el reporte seleccionado.

9. Muestra un mensaje indicando el éxito de la operación.

Complejidad	<i>Media</i>
Prioridad	<i>Media</i>
Precondiciones	El usuario debe tener permisos habilitados para actualizar los estados de las incidencias.
PostCondiciones	El estado de la incidencia se modifica y se adiciona un registro al historial de cambios de la incidencia.

Acción del actor	Respuesta del Sistema
1. El usuario accede a la interfaz de gestión de incidencias.	2. El sistema muestra una lista con todas incidencias pendientes a asignar a un técnico.
3. El usuario selecciona en la barra superior el botón "Actualizar estado de reporte"	4. El sistema verifica que se haya seleccionado un reporte. 4.2 El sistema muestra un mensaje de confirmación "¿Está seguro que desea actualizar el estado del reporte?" En caso de seleccionar "No" concluye el caso de uso.
5. El usuario selecciona "Si"	6. El sistema muestra un formulario donde permite escoger el nuevo estado de reporte y agregar una nota de no más de 250 caracteres. 6.1 El sistema actualiza el registro seleccionado.
7. El usuario selecciona el estado.	8. El sistema muestra un mensaje de operación exitosa 8.1 El sistema envía correo al usuario informando el cambio de estado de la incidencia que reporto.

Flujos alternativos 6.1

Actualizar registro

Acción del actor	Respuesta del proceso de negocio
	El sistema muestra el mensaje: "Ha ocurrido un error al actualizar los datos del reporte"

II.3 Diseño

El diseño es la etapa del proceso de ingeniería de software en que se desarrolla una aplicación o sistema ejecutable. Para algunos sistemas simples, el diseño y la implementación es ingeniería de software, y todas las demás actividades se fusionan con este proceso. Sin embargo, para sistemas grandes, el diseño y la implementación del programa son sólo uno de una serie de procesos (ingeniería de requerimientos, verificación y validación, etcétera) implicados en la ingeniería de software (30).

II.3.1 Arquitectura

La arquitectura es una representación que permite analizar la efectividad del diseño para cumplir los requerimientos establecidos, considerar alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavía es relativamente fácil y reducir los riesgos asociados con la construcción del software (29).

Arquitectura Cliente Servidor

“En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, produce una demanda de información a cualquiera de las computadoras que proporcionan información, conocidas como servidores” estos últimos responden a la demanda del cliente que la produjo. Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la Internet. Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla como según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura (33).

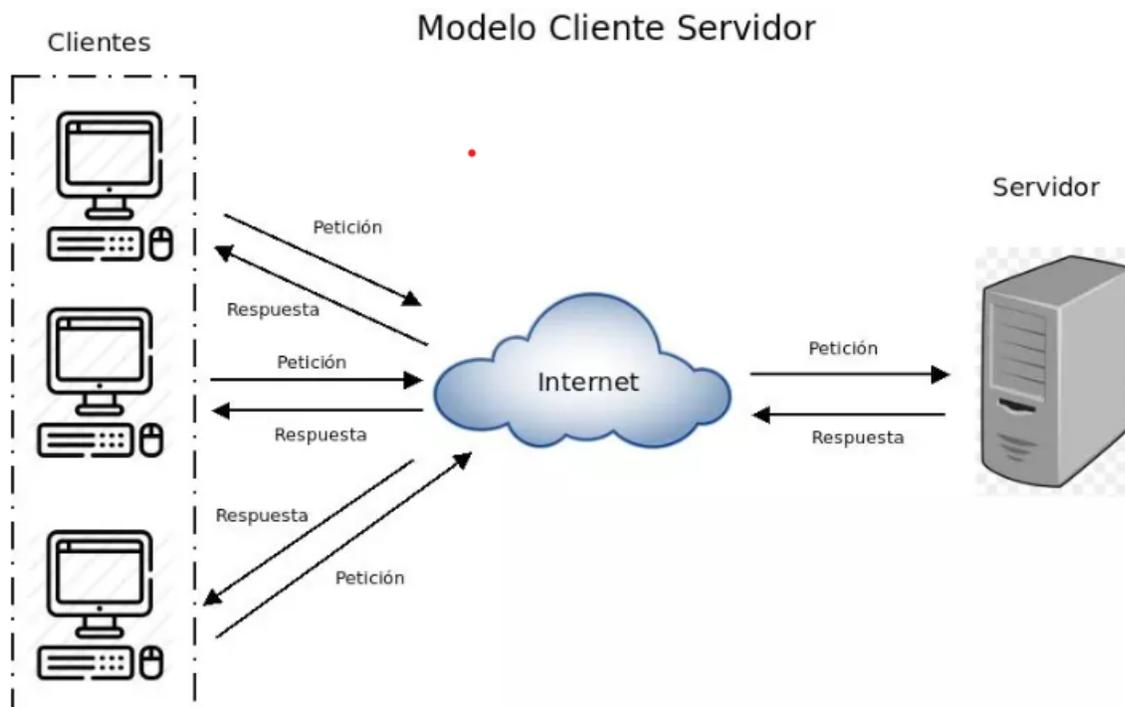


Ilustración 5 Arquitectura Cliente-Servidor(34).

Se decide optar por esta arquitectura debido a que favorece el uso de interfaces gráficas interactivas, teniendo una mayor interacción con el usuario. La estructura modular facilita la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.

II.3.2 Patrones Arquitectónico

Un patrón de diseño es una buena práctica que determina en gran medida la calidad de las aplicaciones de software. Es una solución documentada, que se ha aplicado con éxito en múltiples ambientes para erradicar problemas comunes de diseño de software, con una probada efectividad y con características de reutilización.

Bajo esta perspectiva, los patrones arquitectónicos pueden ser vistos como los bloques constructivos básicos de las arquitecturas de software. Esto es, definen y organizan un vocabulario de elementos basado en un conjunto de componentes y conectores asociados a un grupo de restricciones que indican la forma en la cual tales elementos pueden ser combinados. El objetivo de este vocabulario es la formulación de diseños arquitectónicos que garanticen una adecuada composición de los elementos que lo conforman según el esquema de organización seleccionado. Esta selección no debe realizarse de forma arbitraria, sino que siempre se encuentra relacionada con los requerimientos no funcionales de la aplicación bajo desarrollo (35). De acuerdo con esta perspectiva, los patrones de diseño tienen como objetivo brindar soluciones abstractas a problemas recurrentes en un dominio específico.

El uso del framework Django como escenario de desarrollo, por las características que tiene y su estrecho vínculo con el MVC, está diseñado para promover el acoplamiento débil y una estricta separación entre las piezas de una aplicación. Siguiendo esta filosofía, es fácil hacer cambios en un lugar en particular de la aplicación sin afectar otras piezas.

Debido a que el framework de desarrollo empleado es Django, se opta por el patrón arquitectónico MTV. Se muestra a continuación la estructura del sistema según la arquitectura seleccionada descrita mediante sus diagramas de clases.

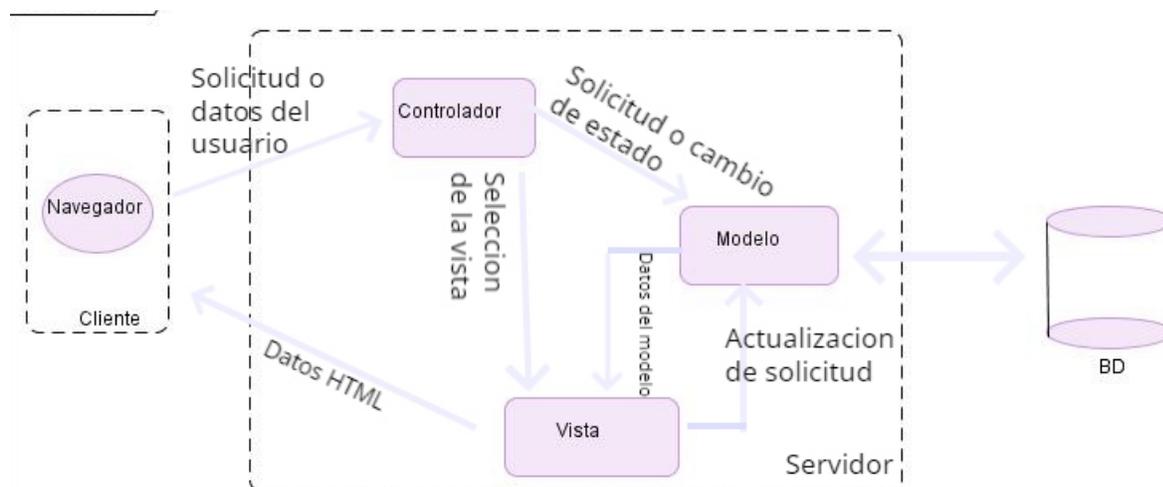


Ilustración 6 Patrón Arquitectónico MTV (elaboración propia)

Django sigue el patrón MVC tan al pie de la letra que puede ser llamado un framework MVC. Someramente, el M, V y C se separan en Django de la siguiente manera:

- M, la porción de acceso a la base de datos, es manejada por la capa de la base de datos de Django, la cual describiremos en este capítulo.
- V, la porción que selecciona qué datos mostrar y cómo mostrarlos, es manejada por la vista y las plantillas.

- C, la porción que delega a la vista dependiendo de la entrada del usuario, es manejada por el framework mismo siguiendo tu URLconf y llamando a la función apropiada de Python para la URL obtenida.

Debido a que la "C" es manejada por el mismo framework y la parte más importante se produce en los modelos, las plantillas y las vistas, Django es conocido como un *Framework MTV*. En el patrón de diseño MTV,

- M significa "Model" (Modelo), la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.
- T significa "Template" (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento.
- V significa "View" (Vista), la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre el modelo y la plantilla.

A continuación, se presentan los diagramas de paquetes para englobar las clases que describen la constitución de las capas de la arquitectura MTV:

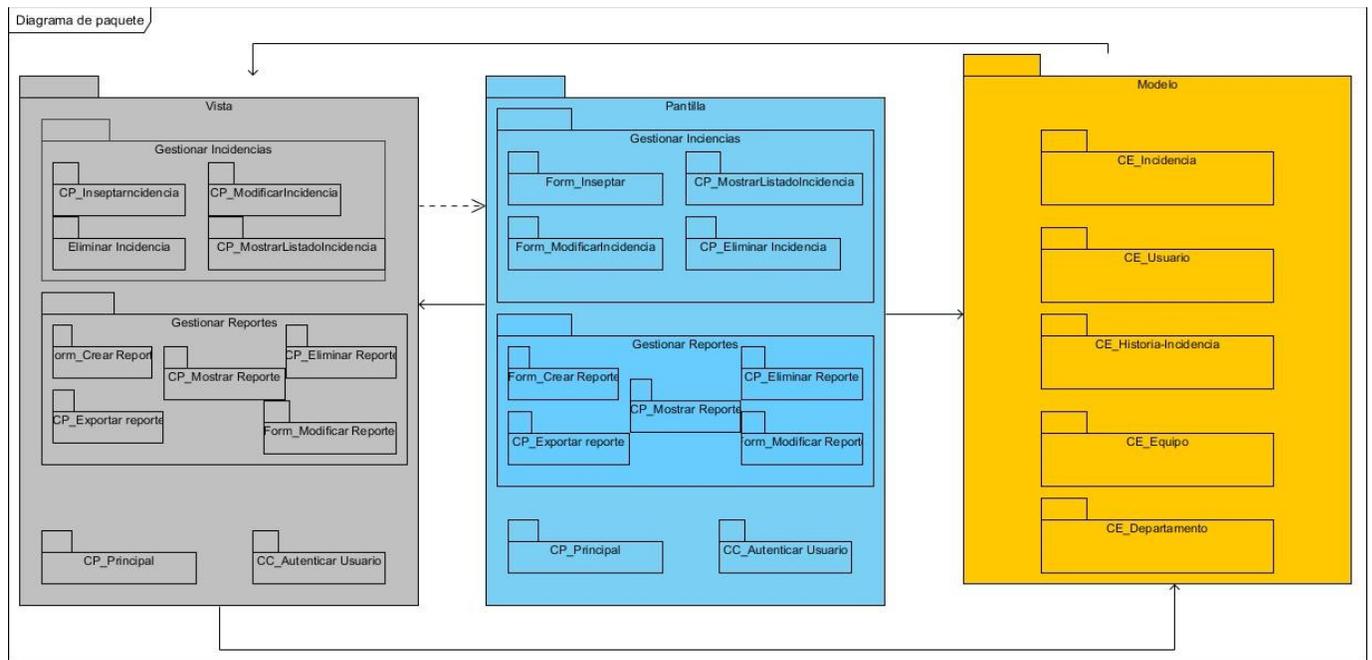


Ilustración 7 Diagrama de paquetes para la solución propuesta (elaboración propia).

II.3.3 Patrones de Diseño

Un patrón de diseño constituye un esquema para refinar subsistemas o componentes. Es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Un patrón de diseño identifica: clases, instancias, roles, colaboraciones y la distribución de responsabilidades, además de que ayuda a construir clases y a estructurar sistemas de clases. Son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces (36).

GRAPS (General Responsibility Assignment Software Patterns, Patrones Generales de Software de Asignación de Responsabilidades).

- **Bajo Acoplamiento:** Consiste en crear pocas dependencias entre las clases con el objetivo que un cambio en alguna de ellas no afecte al resto o dicha afectación tenga una mínima repercusión en el resto de las clases potenciando la reutilización y disminuyendo la dependencia, esto se ve claramente cuando se aplica el estilo MVT, de esta manera se garantiza una dependencia mínima entre las capas Vista y Modelo.
- **Alta Cohesión:** Consiste en que la información que almacena una clase debe de ser coherente y estar en la mayor medida de lo posible relacionada con la clase de modo que cada elemento de nuestro diseño debe realizar una labor única dentro del sistema. Reflejado en todas las clases garantizando que estas posean solo la información pertinente a ellas.
- **Controlador:** está representado por una clase a la cual se le asigna la responsabilidad de las operaciones del sistema. Este patrón se refleja en la clase: Incidencia, encargada de controlar las acciones que realiza el usuario con las interfaces de la aplicación y dar respuesta a las peticiones realizadas.

En el desarrollo de la aplicación web para la gestión de incidencias tecnológicas para el ESTI se aplicó el uso de este patrón para separar la lógica de negocio de la capa de presentación, lo que nos permitió la reutilización del código y así tener un mayor control sobre la aplicación. Se crearon paquetes de servicios que agrupan funcionalidades que son fácilmente reutilizables. A continuación, le mostramos algunas de las funcionalidades que fueron claves en la aplicación de este patrón del diseño:

```

def variante (request):
    user = request.user
    if user is not None:
        if user.is_jefeInf:
            return render (request, "Admin.html")
        if user.is_tecnico:
            return render (request, "semi-Admin.html")
        if user.is_jefeArea:
            return render (request, "jefe_departamentos.html")
        else:
            return render (request, "usuario.html")

def buscar_listado_apropiado(request):
    user = request.user
    if user is not None:
        if user.is_jefeInf:
            return Listado_de_Incidencias(request)
        if user.is_tecnico:
            return Listado_de_Incidencias_Asignadas(request)
        if user.is_jefeArea:
            return Listado_de_Incidencias_usuario(request)
        else:
            return Listado_de_Incidencias_usuario(request)

```

Ilustración 8 Empleo de los patrones de diseño de nuestra aplicación

GoF (Gang of Four)

- Front Controller: patrón usado por el framework Django. Reflejado en el fichero de url.py, recibe las peticiones desde los templates y hace las vistas correspondientes.

II.3.4 Diagrama de clases de diseño

Los diagramas de clase (DC) pueden usarse cuando se desarrolla un modelo de sistema orientado a objetos para mostrar las clases en un sistema y las asociaciones entre dichas clases (30). Estas representaciones contienen información acerca de las clases, asociaciones, atributos, métodos y dependencias. A continuación, se muestra un diagrama de clases del análisis en representación de uno de los principales casos de uso del sistema.

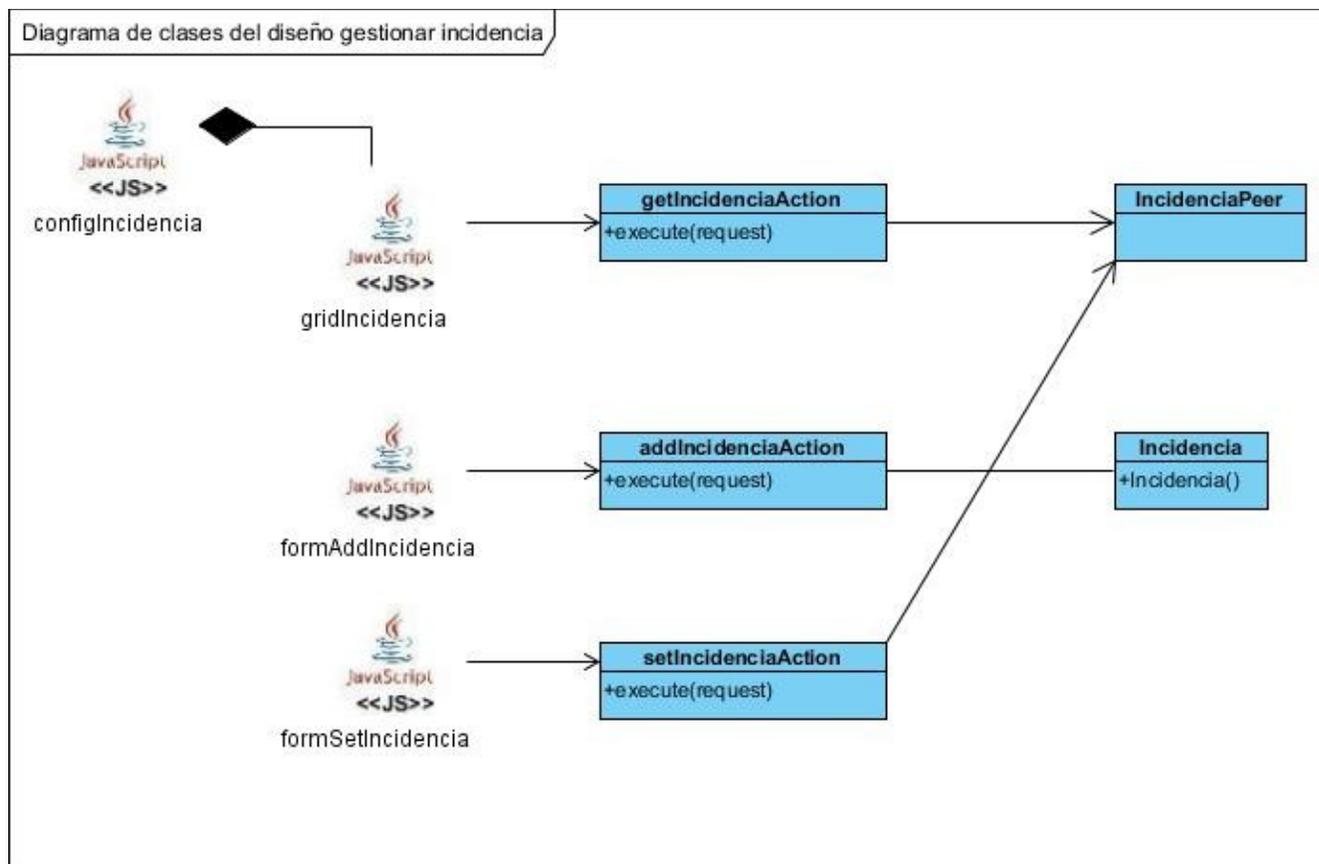


Ilustración 9 Diagrama de clases del diseño Caso de uso Gestionar Incidencia. Escenario Adicionar incidencia (elaboración propia).

II.3.5 Diagrama de interacción

Los diagramas de interacción modelan el comportamiento del sistema y el flujo central en una operación. Describe la interacción entre objetos, los cuales interactúan a través de mensajes para cumplir ciertas tareas. Existen dos tipos de diagramas de interacción: secuencia y colaboración (22).

Los diagramas de secuencia (DS) en el UML se usan principalmente para modelar las interacciones entre los actores y los objetos en un sistema, así como las interacciones entre los objetos en sí (30). Como sugiere el nombre, un diagrama de secuencia muestra la sucesión de interacciones que ocurre durante un caso de uso particular o una instancia de caso de uso (32). En el caso de la solución propuesta se determinaron diagramas de secuencia. Los diagramas de secuencia para los principales casos de uso se observan a continuación:

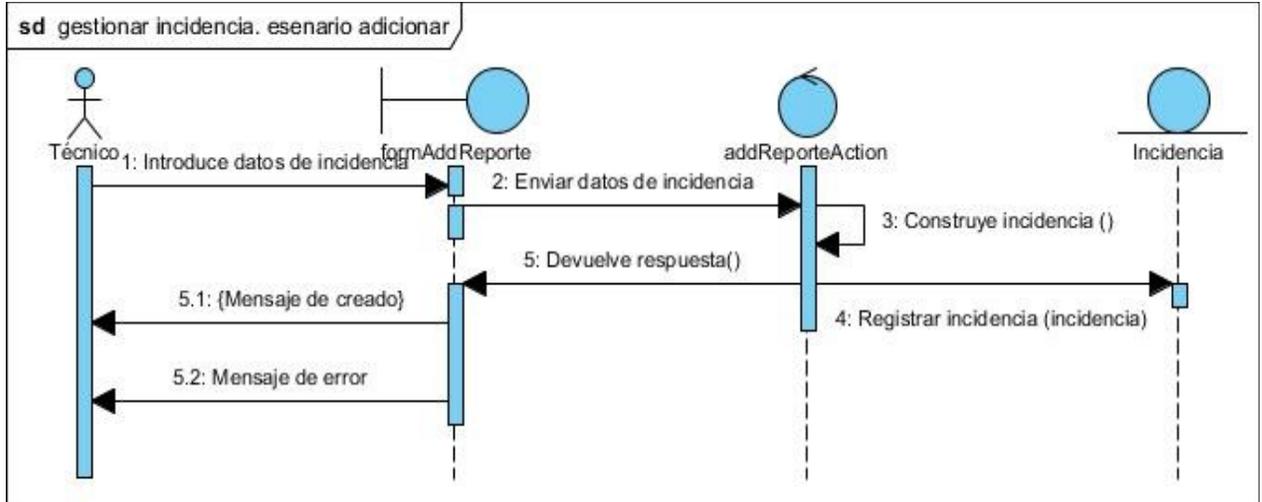


Ilustración 10 Diagrama de secuencia Caso de uso Gestionar Incidencia. Escenario Adicionar incidencia (elaboración propia).

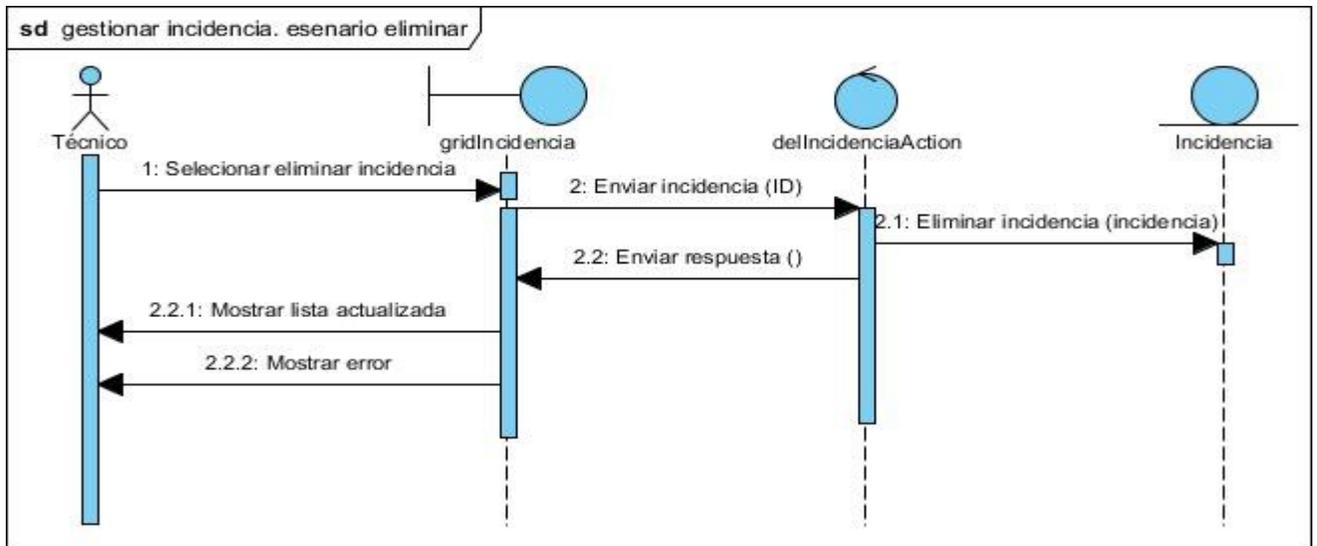


Ilustración 11 Diagrama de secuencia Caso de uso Gestionar Incidencia. Escenario Eliminar incidencia (elaboración propia).

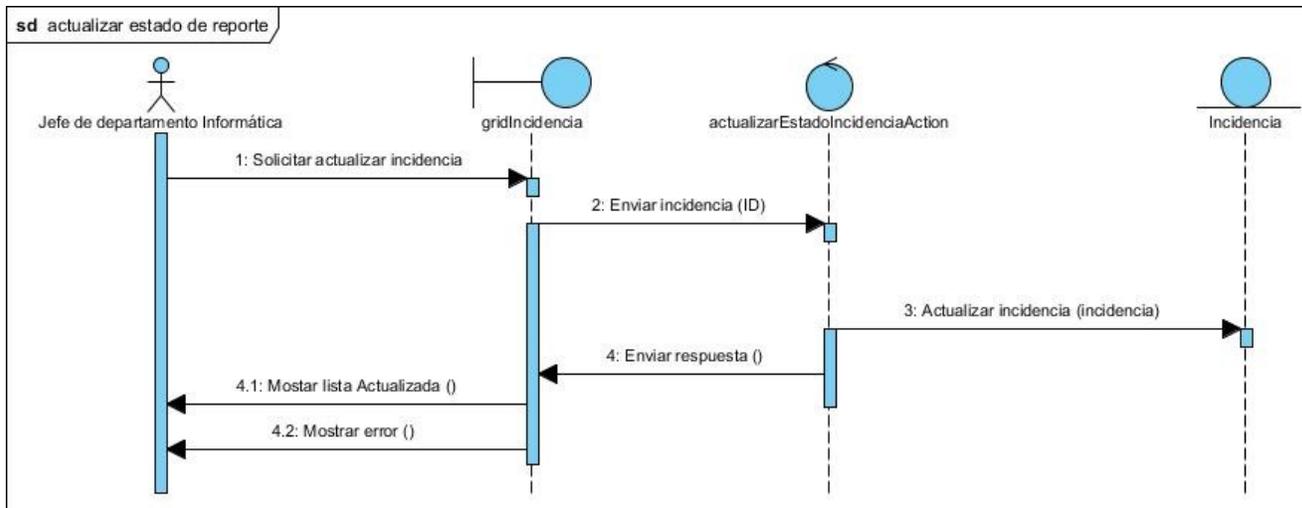


Ilustración 12 Diagrama de secuencia Caso de uso Actualizar estado de incidencia (elaboración propia).

II.4 Diseño de Base de Datos

La construcción de la base de datos es una de las funciones principales en la fase de diseño de un sistema, ya que se exponen los datos necesarios para el correcto desempeño del mismo. El modelado de los datos está descrito por el diagrama entidad-relación, que denota el modo físico en que está dispuesta la información dentro de la base de datos.

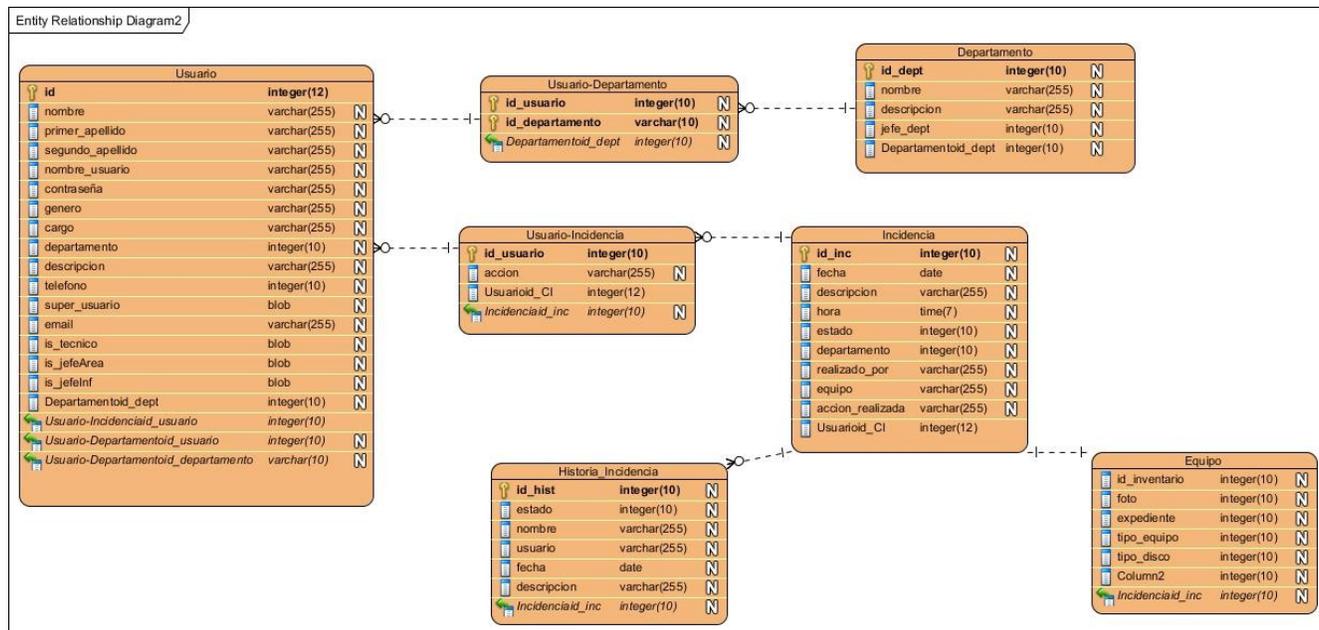


Ilustración 13 Diagrama entidad-relación (elaboración propia).

Tabla 7: Descripción de entidades de la base de datos (elaboración propia).

Entidad	Descripción
Usuario	Entidad que registra los datos generales de cualquier persona, usuario, empleados, terceros, entre otros. Interactúa directamente con demás entidades identificadas para el negocio y permite el registro de todos los usuarios que acceden a la aplicación
Historial de incidencia	Permite definir los diferentes cambios que tiene un reporte, lo que puede ser analizado durante las evaluaciones y representaciones gráficas de los estados dentro de la Dirección de Mantenimiento.
Equipo	Permite definir los distintos tipos de equipos que hay en la entidad.
Incidencia	Entidad principal que recoge los datos de las incidencias.
Departamento	Entidad que recoge el departamento desde cual se realizó el reporte de la incidencia

Conclusiones del capítulo

En el capítulo se describió la propuesta del sistema que se desea implementar para una mejor comprensión del mismo a partir de un modelo de dominio y definiéndose los requisitos funcionales y no funcionales. Quedaron plasmadas las iteraciones por las que transitará el sistema, señalando los casos de uso que serán implementados en cada una de ellos. Se evidenciaron aspectos del diseño de la propuesta de la solución propuesta centrándose en el diseño lógico, quedando definida la arquitectura a utilizar Cliente-Servidor por las ventajas que ofrece y la relación con las tecnologías empleada.

CAPÍTULO III: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

En el siguiente capítulo se abordará la etapa de implementación a partir de la definición del estándar utilizado y detallando el sistema mediante en modelo de implementación. Además, se argumentarán los resultados del despliegue experimental de la solución propuesta.

Se realizará la validación del sistema a partir de las pruebas de software y se argumentaran los resultados del despliegue de la solución propuesta. Se definen los estándares de codificación, que posibilitan la organización, limpieza y claridad en la escritura del código en el proceso de desarrollo del software y se describen los artefactos generados en estas fases.

III.1 Modelo de Despliegue de la solución

Representa de forma visual las relaciones físicas que existen entre los componentes de software y hardware en el sistema. Los nodos son elementos de hardware sobre los cuales pueden ejecutarse los elementos de software. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño.

Descripción de elementos e interfaces de comunicación:

Dispositivo PC_Cliente: La estación de trabajo necesita un navegador web para conectarse al sistema hospedado en el servidor de aplicaciones utilizando el protocolo de comunicación HTTPS.

Servidor de aplicaciones: Es la estación de trabajo que hospeda el código fuente de la aplicación y que le brinda al usuario las interfaces para realizar los procesos del sistema. Es un servidor de aplicaciones web se comunica con el servidor de base de datos donde se almacenan los datos de la aplicación realizando la comunicación mediante el protocolo TCP.

Servidor de BD: Servidor de bases de datos, es el encargado del almacenamiento de los datos del sistema. Se comunica con el servidor de aplicaciones del sistema, posibilitando el acceso mediante el usuario con privilegios para las operaciones determinadas a realizarse en el mismo.

Servidor de correo: Este servidor es el encargado del envío de correos electrónicos comunicándose a través del protocolo SMTP.

A continuación, se presenta el modelo de despliegue correspondiente a la propuesta de solución:

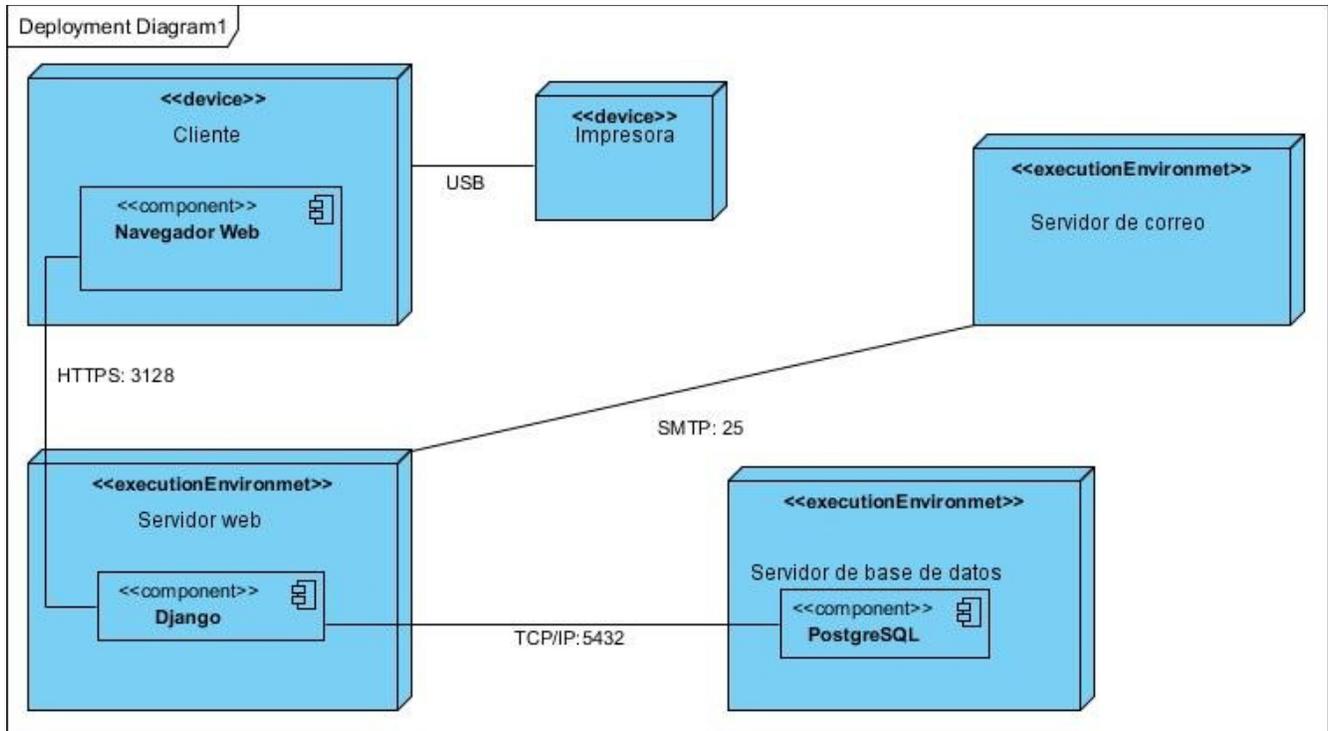


Ilustración 14 Diagrama de despliegue (elaboración propia).

III.2 Pruebas de software

La calidad del software es un proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan (29). Las pruebas representan el último bastión desde donde puede valorarse la calidad y, de manera más pragmática, descubrirse errores. La calidad se incorpora en el software a lo largo de todo el proceso de ingeniería del software. La adecuada aplicación de métodos y herramientas, revisiones técnicas efectivas y gestión y medición sólidas conducen a la calidad que se confirma durante las pruebas (29).

Las pruebas se descomponen en varias actividades interrelacionadas entre sí, cada una con artefactos y trabajadores. Estas contribuyen a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir modificaciones. Una prueba debe comprobar la integración de los elementos del sistema, verificar el funcionamiento completo de las interfaces, así como la comunicación entre los subsistemas. Por otro lado, para garantizar que las pruebas sean exitosas es necesario dividir las por niveles.

Niveles de prueba

Los niveles de prueba permiten definir pruebas de software desde los elementos de cada módulo hasta las cualidades del sistema como un todo. Para el sistema se realizaron las pruebas en diferentes niveles como se detalla a continuación.

Pruebas del sistema: Son pruebas que verifican el funcionamiento del sistema como un todo. Una vez integrados los componentes debe comprobarse el cumplimiento de los requisitos planteados. La comprobación a este nivel permite verificar los atributos de calidad que determinan si el sistema cumple o no los objetivos planteados. Para realizar pruebas al sistema se decidió utilizar el método de la caja negra. Este método permite comprobar el funcionamiento de un sistema a partir del resultado de cada una de las operaciones que debe

cumplir, documentando los resultados sin interesarse en la complejidad de desarrollo de cada operación; a diferencia de los métodos de caja blanca, que si valida que la complejidad y el funcionamiento interno de cada componente cumpla con los requisitos de calidad establecidos para el desarrollo del software.

III.2.1 Diseño de casos de pruebas

Un caso de prueba es una forma específica de probar un sistema testeando las entradas y resultados con las que se debe operar el sistema en las condiciones.

Los casos de prueba son la documentación de una prueba en la mayoría de los casos, de Caja Negra, donde se prueba un componente, módulo, acción o propiedad según el resultado que se espera, sin detenerse en su funcionamiento interno. Para probar la solución se definieron pruebas de funcionalidad, por tanto, los casos de prueba representan pruebas de Caja Negra de Funcionalidad.

Caso de prueba para el caso de uso: Autenticar Usuario.

Tabla 8: Variables del caso de prueba Autenticar usuario (elaboración propia).

No.	Nombre	Descripción	Tipo	Nulo
1	Usuario	Nombre del usuario que se autentica.	texto	No
2	Contraseña	Contraseña del usuario	texto	No

Tabla 9: Matriz de caso de prueba Autenticar usuario (elaboración propia).

Escenario	Combinación	Respuesta esperada	Resultado
Introducir correctamente usuario y contraseña.	1-1	El sistema debe validar los datos y dar acceso al usuario según los permisos de este.	Satisfactorio
Introducir datos de usuario y contraseña incorrectos.	1-1	El sistema valida los datos y notifica al usuario que la combinación usuario contraseña es incorrecta.	Satisfactorio
Dejar en blanco campos o usuario o contraseña.	0-1 o 1-0	El sistema muestra el campo vacío en rojo con un mensaje de: "el campo no puede estar vacío"	Satisfactorio

Caso de prueba para el caso de uso: Gestionar incidencias.

Tabla 10: Variables del caso de prueba Gestionar incidencias (elaboración propia).

No.	Nombre	Descripción	Tipo	Nulo
1	Nombre	Define el nombre de la persona que realiza la incidencia.	texto	No

2	Área	Área de la entidad donde se realiza la incidencia.	Lista desplegable	No
3	Equipo o sistema	Define el tipo de incidencia definiendo el equipo que va realizar el reporte de la incidencia.	Lista desplegable	No
4	Responsable	Define el responsable para dar solución a la incidencia.	Lista desplegable	No
5	Descripción	Describe el hecho que origina la incidencia.	texto	No
6	Usuario	Define el nombre de usuario que realiza la incidencia.	Auto-completable	No

Tabla 11: Matriz de caso de prueba Gestionar incidencia (elaboración propia).

Escenario	Combinación	Respuesta esperada	Resultado
Introducir correctamente datos de la incidencia.	1-1-1-1-1	El sistema asigna al reporte la fecha y la hora, le asigna la brigada y registra el técnico que define el reporte y muestra una notificación de "Reporte guardado satisfactoriamente"	Satisfactorio
Descripción de la incidencia está vacía	1-1-1-1-0-1	El sistema señala el campo de descripción con un color rojo mostrando un mensaje: "La descripción del reporte no puede estar vacía"	Satisfactorio
No se selecciona el tipo de incidencia.	1-1-0-1-1-1	El sistema señala con color rojo en campo vacío y muestra el mensaje: "El campo no puede estar vacío"	Satisfactorio
No se selecciona el área donde se realiza la incidencia.	1-0-1-1-1-1	El sistema señala con color rojo en campo vacío y muestra el mensaje: "El campo no puede estar vacío"	Satisfactorio
Se intenta modificar una incidencia cuyo estado no es el inicial.	-	El sistema muestra un mensaje de: "La incidencia seleccionada no puede ser modificada ni eliminada porque ya ha sido asignado a un técnico"	Satisfactorio
Se intenta eliminar una incidencia cuyo estado no es el inicial.	-	El sistema muestra un mensaje de: "La incidencia seleccionada no puede ser modificada ni eliminada porque ya ha sido asignado a un técnico"	Satisfactorio

III.2.2 Pruebas para requisitos no funcionales

Las pruebas de rendimiento se diseñan para asegurar que el sistema pueda procesar su carga esperada. Éstas se ocupan tanto de demostrar que el sistema satisface sus requerimientos, como de descubrir problemas y defectos en el sistema (30).

Las pruebas de carga consisten en simular una carga de trabajo similar y superior a la que tendrá cuando el sistema esté funcionando, con el fin de detectar si el software instalado (programas y aplicaciones) cumple con los requerimientos de muchos usuarios simultáneos y también si el hardware (servidor y el equipamiento computacional de redes y enlace que lo conecta a Internet) es capaz de soportar la cantidad de visitas esperadas (29).

Las pruebas de estrés evalúan la robustez y la confiabilidad del software sometiéndolo a condiciones de uso extremas. Entre estas condiciones se incluyen el envío excesivo de peticiones y la ejecución en condiciones de hardware limitadas. El objetivo es saturar el programa hasta un punto de quiebre donde aparezcan defectos potencialmente peligrosos (29).

Resultados de las pruebas de rendimiento

Para las pruebas de rendimiento se utiliza el software Apache JMeter v2.10. Para ello se definen las propiedades de las PC implicadas.

Hardware de prueba (PC servidor):

- Procesador i5 de sexta generación.
- Memoria RAM: 8 GB.
- Tarjeta de Red Ethernet 100 Mb/s
- Capacidad: 1 TB.

Hardware de prueba (PC cliente):

- Procesador Intel Celeron 1.6 GHz.
- Memoria RAM: 4 GB
- Tarjeta de Red Ethernet: 100 Mb/s
- Capacidad: 60 Gb.

Luego de definido el hardware se configuran los parámetros del Apache JMeter logrando un ambiente de simulación con un total de 102 usuarios conectados concurrentemente, se realizan peticiones a diferentes páginas de la Aplicación web para la gestión de incidencias tecnológicas en el Equipo de Servicio de Traductores e intérpretes.

La aplicación de las pruebas de rendimiento a nuestra propuesta solución permitió comprobar el cumplimiento de los requisitos no funciones que se establecieron como características que el sistema debía cumplir. Durante las pruebas de carga el servidor soportó 110 conexiones simultáneas a una tasa estable de respuesta de 61 ms, con una carga para el cliente de 3.4 Mb durante el primer almacenamiento del contenido en la caché del navegador cliente. Para las pruebas con la información almacenada previamente en la caché, las tasas estables se comportaron para 58 peticiones de carga, 61 kb de datos, con una cantidad máxima de 100 conexiones simultáneas.

III.3 Métodos y Técnicas

Para el desarrollo de las pruebas se utilizó el método de particiones equivalentes, que permite dividir la etapa de pruebas en iteraciones, lo que garantiza que en cada iteración se pueda solucionar las No Conformidades de la iteración anterior. Para la solución que se presenta se aplicaron dos iteraciones de prueba. Las No Conformidades detectadas durante ambas iteraciones por cada caso de uso pueden observarse a continuación.

Tabla 12: No conformidades por Caso de uso (elaboración propia).

Caso de prueba	Interacción	No conformidades	Cerradas
Autenticar usuario	1ra	2	2
	2da	1	1
Buscar incidencia	1ra	1	1
	2da	0	0
Gestionar incidencia	1ra	4	4
	2da	0	0
Gestionar usuario	1ra	3	3
	2da	0	0
Gestionar reporte	1ra	4	4
	2da	1	1
Actualizar estado de incidencia	1ra	2	2
	2da	0	0
Graficar incidencia	1ra	1	1
	2da	0	0

A partir del cierre de las No Conformidades detectadas en las dos iteraciones de la etapa de prueba se demuestra el cumplimiento de los objetivos planteados para la investigación. Como contrapartida, el proceso de prueba fue validado mediante pruebas funcionales realizadas por los clientes finales, que aprobaron la herramienta y certificaron su realización satisfactoria.

III.4 Prueba de rendimiento

Para dichas cantidades de conexiones, se aseguran todos los requisitos no funcionales, por lo que, dada las condiciones del ambiente de despliegue del sistema, las características de los ordenadores clientes y el servidor, además de las posibles conexiones simultáneas, se considera que las pruebas son satisfactorias.

III.5 Prueba de seguridad

Para la realización de este tipo de prueba, se empleó la herramienta Acunetix Web Vulnerability Scanner, caracterizada en el capítulo 1. En una primera iteración, se obtuvo un total de quince (10) no conformidades, divididas en seis (4) de nivel medio, seis (3) de nivel bajo y tres (3) de carácter informativo. De las de nivel medio, destacó el uso del protocolo no seguro para el envío de datos, así como campos de contraseña con auto completamiento activado. Las de nivel bajo estuvieron relacionadas con problemas para la protección contra ataques de fuerza bruta a la página de autenticación, así como directorios que pueden ser accesibles directamente sin pasar la autenticación y la protección de las cookies y las sesiones en el navegador. De carácter informativo fueron detectadas una dirección de correo y una posible cuenta de usuario en un fichero. Todas estas deficiencias fueron corregidas en la primera iteración, y para una segunda, no se identificó ninguna nueva, por lo cual se obtuvo finalmente una herramienta con seguridad y que cumple con el requisito de confiabilidad definido para la misma.

Interfaces principales de la Aplicación Web para la gestión de incidencias tecnológicas en el Equipo de Servicios para Traductores e Intérpretes

Una vez desarrollado el sistema de gestión de reporte de incidencia, es posible visualizar las pantallas principales del mismo, donde se observa el resultado obtenido durante la implementación de los requisitos funcionales descritos en el capítulo 2.

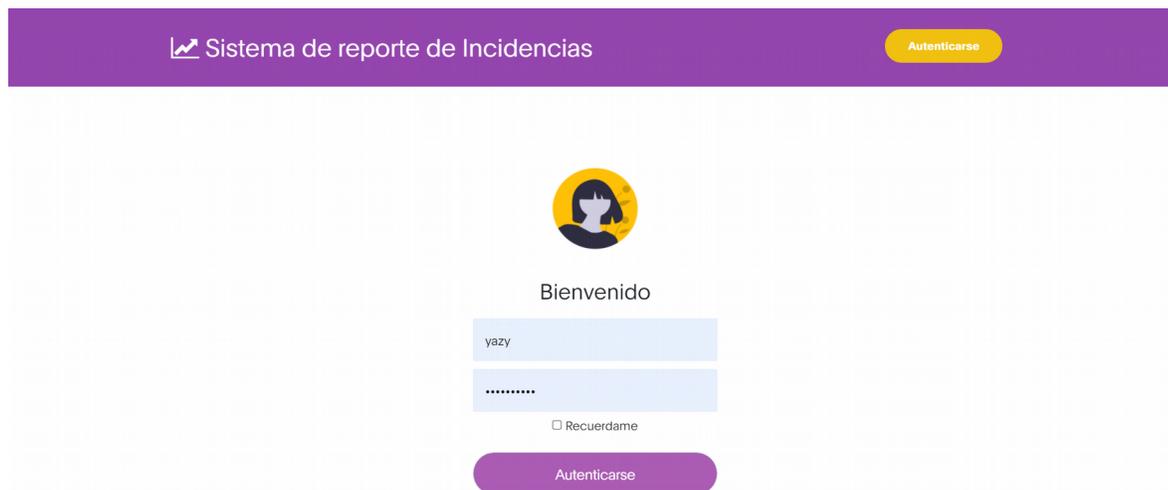


Ilustración 15 Captura de pantalla del sistema (autenticar usuario) (elaboración propia).



Ilustración 16 Captura de pantalla del sistema (pantalla principal) (elaboración propia).

Reporte su **incidencia**

Nombre ▾

Seleccione su área ▾ Seleccione el sistema o equipo a reportar ▾

Descripción

Enviar



CONNECTIONS



ESTI

Departamento
Informático

Ilustración 17 Captura de pantalla del sistema (pantalla para reportar incidencia) (elaboración propia).

Conclusiones del capítulo

En el capítulo se valida la propuesta solución a través de la estrategia de pruebas específicas. En las pruebas se reflejaron que los requisitos establecidos fueron cumplidos según los parámetros de calidad de software que se tuvieron en cuenta y garantizaron determinar las no conformidades, así como la evaluación del sistema hasta la herramienta que satisface los problemas de automatización del negocio. Se arrojan como resultados que el sistema implementado responde a los requerimientos definidos por el cliente.

CONCLUSIONES FINALES

Una vez obtenida una versión más completa de la Aplicación web para la gestión de incidencias tecnológicas en el Equipo de Servicios para Traductores e Intérpretes, se puede concluir que:

1. La análisis, sistematización y fundamentación teórica de los principales conceptos de la investigación, permitió lograr una mayor comprensión del alcance de la investigación, garantizó la elaboración del marco teórico de la investigación científica y del estado actual de las herramientas informáticas para la gestión de reportes de incidencias, posibilitó la definición del ambiente de desarrollo para la implementación de la solución propuesta.
2. La aplicación web garantiza que el sistema se comporte de manera fiel y responda a las necesidades reales del Departamento Informático de la entidad, mejorando los procesos de distribución de incidencias según el área, asegurando la trazabilidad y brindando mejor clasificación.
3. Se realizaron pruebas exploratorias a la aplicación implementada reportando resultados satisfactorios.
4. La Aplicación web para la gestión de incidencias tecnológicas en el Equipo de Servicios para Traductores e Intérpretes, proporciona una mejor capacidad de gestión de incidencias, mejora la búsqueda de las tareas de análisis, garantiza mejores estimaciones e involucra a los usuarios en un proceso más cómodo y seguro, lo que influye positivamente en la toma de decisiones.
5. La solución fue validada a partir de la definición correcta de una estrategia de pruebas, que permitió comprobar el correcto funcionamiento de la Aplicación web para la gestión de incidencias tecnológicas en el Equipo de Servicio de Traductores e Intérpretes, a partir de los requerimientos definidos por el cliente

RECOMENDACIONES

Se recomienda lo siguiente:

1. Se recomienda en otras versiones de la aplicación: Incluir un módulo para la automatización de la asignación de incidencias.
2. Implementar una funcionalidad que permita trabajar con el directorio activo de la entidad.
3. Implementar una funcionalidad para las notificaciones del sistema, mediante correo electrónico.

REFERENCIAS BIBLIOGRÁFICAS

1. Conceptos y principios ITIL | Las buenas prácticas de ITIL. Online. [Accessed 2 November 2022]. Available from: <https://www.servicetonico.com/es/itil/3-itil-conceptos-y-principios/>
2. OLIVA, Rodrigo, CARVAJA, Karina and CATALDO, Alejandro. Impacto TI en las pequeñas y medianas empresas ¿es su efecto moderado por la intensidad de uso de ti de la industria? . 2018. Vol. 13, no. 82–93.
3. Historia | Universidad de las Ciencias Informáticas. Online. 2018. [Accessed 27 October 2022]. Available from: <https://www.uci.cu/universidad/historia>
4. ESTI | Historia. *ESTI*. Online. 2022. [Accessed 26 May 2022]. Available from: <https://esti.cu/paginas/historia>
5. ¿Qué es un sistema web? Online. 2018. Available from: <http://www.addappto.com/que-es-un-sistema-web/>
6. NARANJO, Francisco José. Sistemas de Gestión: Valor Estratégico de las Organizaciones - Consultoría & Consultores. Online. 2015. [Accessed 26 May 2022]. Available from: <https://consultoria-consultores.es/articulos/articulo-consultoria-sistemas-de-gestion-valor-estrategico-de-las-organizaciones/>
7. VARGAS, Yohannia López and CHÁVEZ, Alejandro Vázquez. La Gestión de Servicios de soporte técnico en el ciclo de vida del desarrollo de software. . 2016. Vol. 10, p. 15.
8. MARTÍNEZ LÓPEZ, esteban. QUE ES UN REPORTE Y CUAL ES SU USO. *Prezi*. Online. 2015. [Accessed 26 May 2022]. Available from: <https://prezi.com/-zoqwmyhq2ls/que-es-un-reporte-y-cual-es-su-uso/>
9. ORGANIZACIÓN MUNDIAL DE LA SALUD. *Sistema de gestión de la calidad en el laboratorio: manual*. Online. Geneva : Organización Mundial de la Salud, 2016. [Accessed 26 October 2022]. ISBN 978-92-4-354827-2. Available from: <https://apps.who.int/iris/handle/10665/252631>
10. WESTREICHER, Guillermo. Mantenimiento - Qué es, definición y concepto | 2022 | Economipedia. *Economipedia*. Online. 2020. [Accessed 27 May 2022]. Available from: <https://economipedia.-com/definiciones/mantenimiento.html>
11. PÉREZ, Mariana. ¿Qué es Metodología? » Definición, Tipos y Ejemplos 2021. *Concepto definicion*. Online. 2021. [Accessed 15 May 2022]. Available from: <https://conceptodefinicion.de/metodologia/>
12. LETELIER, P and PENADÉS, M.C. Metodologías ágiles para el desarrollo de software. Online. 2006. Vol. 6, no. 26, p. 17. Available from: <http://www.jstor.org/stable/3541599?origin=crossref>
13. OpenUp. *EcuRed*. Online. 2008. [Accessed 26 October 2022]. Available from: <https://www.ecured.cu/OpenUp>
14. SALGADO, Santiago Ríos, RAZA, Ing Cecilia Hinojosa and RODRÍGUEZ, Ing Ramiro Delgado. APLICACIÓN DE LA METODOLOGÍA OPENUP EN EL DESARROLLO DEL SISTEMA DE DIFU-

SIÓN DE GESTIÓN DEL CONOCIMIENTO DE LA ESPE. . P. 11. This article explains how the methodology OpenUP was applied in the development of the web application “Sistema de Difusión de Gestión del Conocimiento de la ESPE”, which is considered strategic in the process of national and international accreditation of the Escuela Politécnica del Ejército. OpenUP was considered because it uses a pragmatic, agile philosophy approach that focuses on the collaborative nature of the software development process, in addition it offers a high degree of adaptability to the needs of a particular project because of its iterative kind.

15. Software de mantenimiento preventivo y correctivo - Protecnius. *Protecnius*. Online. 2022. [Accessed 26 May 2022]. Available from: <https://www.protecnius.com/>

16. OTRS | Soluciones de software para Servicio al Cliente, ITSM, ISMS, SOAR. Online. 2022. [Accessed 26 October 2022]. Available from: <https://otrs.com/es/home/>

17. HDKey: Software gestión incidencias y tickets tipo Help Desk - KMKey. *KM KEY*. Online. [Accessed 26 May 2022]. Available from: https://www.kmkey.com/software_help_desk/

18. SUITECRM SOFTWARE - SuiteCRM | Software CRM. *Suite crm*. Online. [Accessed 26 May 2022]. Available from: <https://suitecrm.es/>

19. DIRRECCIÓN GENERAL DE INFORMATICA, MINISTERIO DE COMUNICACIÓN. *Relación de aplicaciones informáticas inscritas en el Sistema de Inscripción de Programas y Aplicaciones Informáticas*. la Habana, 2020.

20. RODRIGUEZ, Susana Antón. ¿Cómo se gestionan los incidentes de ciberseguridad en Cuba? *Granma*. Online. 2021. [Accessed 26 September 2022]. Available from: <https://www.granma.cu/cuba/2021-08-29/como-se-gestionan-los-incidentes-de-ciberseguridad-en-cuba-video>

21. What is Modeling Language? - Definition from Techopedia. *Techopedia*. Online. 2016. [Accessed 26 June 2022]. Available from: <https://www.techopedia.com/definition/20810/modeling-language>

22. FOWLER, Martin. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. . Pearson Education, 2018. ISBN 978-0-13-486512-6. Google-Books-ID: VTdtDwAAQBAJ

23. Herramienta ideal de modelado y diagramación para la colaboración ágil en equipos. *Visual paradigm*. Online. 2018. [Accessed 27 October 2022]. Available from: <https://www.visual-paradigm.com/>

24. MENÉMDEZ, Rafael. Cursos de JavaScript. Online. 2021. [Accessed 26 May 2022]. Available from: <https://www.lawebdelprogramador.com/cursos/JavaScript/index1.html>

25. ALVAREZ, Alberto Cuevas. *Programar con Python 3*. . Lulu.com, 2019. ISBN 978-0-359-68516-5. Libro sobre programación en Python 3, con más de 300 códigos de ejemplo. Cubre, desde cero, los aspectos fundamentales de Python, así como otros que podemos considerar avanzados.- Google-Books-ID: e4K4DwAAQBAJ

26. NIETO, Tito Fernando. Introducción al HTML 5. Online. 2010. [Accessed 26 May 2022]. Available from: <https://www.monografias.com/trabajos89/introduccion-al-html-5/introduccion-al-html-5>

27. Bootstrap: qué es y cómo funciona este framework **【Usos】** . *Axarnet*. Online. [Accessed 26 May 2022]. Available from: <https://axarnet.es/blog/bootstrap>
28. Sistemas gestores de bases de datos. Online. 2007. [Accessed 26 May 2022]. Available from: <https://desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>
29. PRESSMAN, Roger S. *INGENIERIA DE SOFTWARE*. . McGraw-Hill Interamericana de España S.L., 2010. ISBN 978-607-15-0314-5. Google-Books-ID: deuwCQAACAAJ
30. SOMMERVILLE, Ian. *Ingeniería de Software*. Online. Novena. México, 2011. ISBN 978-607-32-0603-7. Available from: https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9_compressed.pdf
31. La Oficina de Proyectos de Informática. Online. [Accessed 26 October 2022]. Available from: <http://www.pmoinformatica.com/>
32. PRESSMAN, Roger S. and MAXIM, Bruce R. *Ingeniería del software: un enfoque práctico*. . McGraw Hill Interamericana Editores, 2021. ISBN 978-1-4562-8772-6. Google-Books-ID: sQZ6zgEACAAJ
33. VALLE, Jose Guillermo and GILDARDO GUTIERREZ, Jame. DEFINICIÓN ARQUITECTURA CLIENTE SERVIDOR. . No. 2005, p. 10.
34. Modelo cliente servidor: ¿Qué es? Características, Ventajas y Desventajas. Online. [Accessed 2 November 2022]. Available from: <https://blog.infranetworking.com/modelo-cliente-servidor/>
35. SALAS-ZÁRATE, María del Pilar, ALOR-HERNÁNDEZ, Giner, VALENCIA-GARCÍA, Rafael, RODRÍGUEZ-MAZAHUA, Lisbeth, RODRÍGUEZ-GONZÁLEZ, Alejandro and LÓPEZ CUADRADO, José Luis. Analyzing best practices on Web development frameworks: The lift approach. *Science of Computer Programming*. Online. 1 May 2015. Vol. 102, p. 1–19. [Accessed 26 October 2022]. DOI 10.1016/j.scico.2014.12.004.
36. Patrones de Diseño Contenido ? Patrones GRASP ? Experto ? Creador ? Bajo Acoplamiento | antonio hernandez - Academia.edu. *Academia*. Online. 2018. [Accessed 1 October 2022]. Available from: https://www.academia.edu/4726489/Patrones_de_Dise%C3%B1o_Contento_Patrones_GRASP_Experto_Creador_Bajo_Acoplamiento

ANEXOS

Anexo 1. Entrevista al cliente para conocer la necesidad de la propuesta de solución y definir los requisitos funcionales y no funcionales.

Estimado: Se necesita de su cooperación en una investigación para una tesis de pregrado. Por ello, sería de gran ayuda que respondiera lo siguiente:

1. ¿Considera que a día de hoy el servicio de mantenimiento tecnológico cumple con su objetivo de satisfacer al personal al que se atiende, teniendo en cuenta la cantidad de departamentos que tiene el ESTI?
2. ¿Cree usted que cuenta con los recursos y medios necesarios para dar una respuesta a la persona afectada?
3. ¿Existe alguna herramienta en la entidad para la gestión de incidencias tecnológicas?
4. ¿Considera que el sistema deba permitir que el sistema muestre al usuario autor de una incidencia siempre el estado actualizado de la misma desde que este la crea hasta que se cierra?
5. ¿Considera el sistema deba generar reportes, para una mejor gestión de las incidencias y contribución a la toma de decisiones?
6. ¿Qué otras características, considera que deba presentar el sistema, en cuanto usabilidad, seguridad, interfaz u otro aspecto que garantice su calidad?

Anexo 2: Especificación Caso de uso Gestionar Usuario

Caso de Uso del Sis- *Gestionar usuario* tema

Objetivo	Permitir añadir, modificar, listar y eliminar incidencias
Actores	Usuario administrador
Resumen	El caso de uso se inicia cuando el jefe de informática o administrador del sistema gestionar un usuario, a partir de las necesidades de la entidad el sistema debe permitir añadir nuevos usuarios modificar su información, o eliminar usuarios del sistema.
Complejidad	media
Prioridad	Alta
Precondiciones	El usuario debe estar autenticado y con permisos de gestionar usuario
PostCondiciones	Se adiciona un nuevo usuario al sistema

Acción del actor	Respuesta del Sistema
-------------------------	------------------------------

1. El usuario accede mediante al la página principal del administrador o del jefe de informática.	2. El sistema muestra una lista de funcionalidades que realiza el sistema.
---	--

3. El usuario selecciona la opción de adicionar, modificar o eliminar.	
--	--

Sección: Adicionar Usuario	
-----------------------------------	--

3. El usuario selecciona la opción de adicionar usuario.	4. El sistema muestra un formulario solicitando los datos del usuario a registrar.
--	--

5. El usuario introduce los datos necesarios para reportar una incidencia.	6. El sistema verifica si los datos están correctos. En caso de faltar algún dato se pasa al flujo alternativo A, en caso de que el reporte ya exista se pasa al flujo alternativo B.
--	---

	8. Registra un nuevo usuario.
--	-------------------------------

	9. Muestra un mensaje indicando el éxito de la operación.
--	---

Flujo alternativo A

El sistema señala los campos vacíos.

Flujo alternativo B

Muestra el mensaje "El usuario ya esta registrado en el sistema".

Sección Modificar usuario	
----------------------------------	--

3. El usuario selecciona la opción de modificar incidencia.	4. El sistema muestra un formulario
---	-------------------------------------

	5. El sistema muestra un formulario con los datos del usuario.
--	--

	6. El sistema verifica si los datos están correctos, en caso de faltar algún dato se pasa al flujo alternativo A.
--	---

	8. Modifica el usuario seleccionado.
--	--------------------------------------

	9. Muestra un mensaje indicando el éxito de la operación.
--	---

Flujo alternativo A

El sistema señala los campos vacíos.

Sección eliminar usuario

3. El usuario selecciona la opción de eliminar incidencia.
4. El sistema verifica que el usuario este registrado en el sistema.
5. El usuario queda eliminada.
6. El listado de usuarios registrados en el sistema actualizado.

Prototipo de Interfaz



Anexo 3: Especificación Caso de uso Mostrar reporte de incidencia

Caso de Uso del Sis- *Mostrar reporte de incidencia* tema

Objetivo	<i>Mostrar un reporte de incidencias realizadas</i>
Actores	<i>Jefe de informática</i>
Resumen	<i>El caso de uso se inicia cuando el jefe de informática desea ver un informe de las incidencias reportadas en la entidad.</i>
Complejidad	<i>Media</i>
Prioridad	<i>Alta</i>
Precondiciones	<i>El usuario debe estar autenticado en el sistema y con permiso para realizar reporte.</i>
PostCondiciones	<i>Se muestra un reporte con toda la información de las incidencias.</i>
Acción del actor	Respuesta del Sistema

1. El usuario accede mediante la página principal del jefe de informática.
2. El sistema muestra una lista de funcionalidades que realiza el sistema.
3. El usuario selecciona la opción de generar reporte.

Acción del actor	Respuesta del Sistema
	El sistema muestra un mensaje indicando “El usuario introducido no está registrado en el sistema” o “la contraseña es incorrecta”



Anexo 4: Diagrama de secuencia Caso de uso Gestionar Incidencia. Escenario Modificar incidencia (elaboración propia).

sd modificar incidencia

