

UNIVERSIDAD PÚBLICA DE EL ALTO
CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO
“SISTEMA DE INFORMACION WEB PARA EL REGISTRO Y CONTROL DE EMERGENCIAS EN INFRAESTRUCTURAS DE LAS UNIDADES EDUCATIVAS EN LA CIUDAD DE EL ALTO”

CASO: GOBIERNO AUTONOMO MUNICIPAL DE EL ALTO G.A.M.E.A

Para Optar al Título de Licenciatura en Ingeniería de Sistemas
MENCIÓN: GESTIÓN Y PRODUCCIÓN

Postulante: Marco Antonio Yujra Mendoza

Tutor Metodológico: M.Sc. Lic. Ing. Marisol Arguedas Balladares

Tutor Especialista: Lic. Margarita Bernarda López Mariaca

Tutor Revisor: Lic. Nelson Egberto Tarqui Carpio

**EL ALTO – BOLIVIA
2024**

DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Marco Antonio Yujra Mendoza, estudiante con C.I. 6111752 LP mediante la presente:

declaro de manera pública que la propuesta del PROYECTO DE GRADO titulado "SISTEMA DE INFORMACION WEB PARA EL REGISTRO Y CONTROL DE EMERGENCIAS EN INFRAESTRUCTURAS DE LAS UNIDADES EDUCATIVAS EN LA CIUDAD DE EL ALTO", Caso: GOBIERNO AUTONOMO MUNICIPAL DE EL ALTO G.A.M.E.A es original, siendo resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados, Autorizo la publicación del resumen de mi propuesta en internet y me comprometo a responder a todos los cuestionamientos que se desprenden de su lectura.

Asimismo, me hago responsable ante la universidad o terceros, de cualquiera irregularidad o daño que pudiera ocasionar, por el incumplimiento de lo declarado. De identificarse falsificación, plagio, fraude, o que el PROYECTO DE GRADO haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas legales que se deriven de ello sometiéndome a las normas establecidas y vigentes de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

El Alto, noviembre de 2024

Univ. Marco Antonio Yujra Mendoza

C.I. 6111752 L.P.

e-mail: marcoantonio.yujramendoza@gmail.com

Dedicatoria

Con Mucho Cariño:

A mis padres, por ser mi principal inspiración y el pilar de mi formación

Ruego siempre tener la fortaleza para honrar su esfuerzo y dedicación

A mis amigos, por creer en mí y acompañarme en este camino

Cada instante compartido ha sido un recordatorio de que nunca camino solo

En este proyecto, se refleja el apoyo incondicional que he recibido

Llevaré siempre en mi corazón los valores y lecciones que me han brindado

Y a todos, les agradezco por ser parte de este logro

Agradecimientos

En primer lugar, darle gracias a Dios, por estar siempre conmigo,

guiando mis pasos y fortaleciendo mi espíritu.

Agradecerles el apoyo a mis padres, y en especial a mi madre, por su apoyo

incondicional y por ser mi mayor motivación para salir adelante.

A si mismo agradecer a la carrera de Ingeniería de Sistemas y a los docentes que, con

sus enseñanzas y guía, se convirtieron en piezas fundamentales en mi crecimiento

personal y profesional.

En cada logro alcanzado, reconozco que es el resultado de las lecciones, el apoyo y la

inspiración que he recibido de quienes me rodean.

INDICE

CAPITULO I.....	1
1. INTRODUCCION.....	1
1.2 ANTECEDENTES.....	2
1.2.1. Antecedentes Institucionales.....	2
1.2.1.1 Misión del Gobierno Autónomo Municipal de El Alto.....	2
1.2.1.2 Visión del Gobierno Autónomo Municipal de El Alto.....	2
1.2.1.2 Objetivo.....	2
1.2.1.3 Organigrama.....	3
1.2.2 Antecedentes afines al proyecto de grado.....	4
1.2.2.1 Antecedentes Internacionales.....	4
1.2.2.2 Antecedentes Nacionales.....	5
1.3 PLANTEAMIENTO DEL PROBLEMA.....	6
1.3.1 Problema Principal.....	6
1.3.2 Problemas secundarios.....	6
1.3.3 Formulación del problema.....	7
1.4 OBJETIVOS.....	7
1.4.1 Objetivo General.....	7
1.4.2 Objetivos Específicos.....	7
1.5 JUSTIFICACION.....	8

1.5.1 Justificación Técnica.....	8
1.5.2 Justificación Económica.....	9
1.5.3 Justificación Social.....	9
1.6 METODOLOGIA.....	10
1.6.1 Metodología UWE.....	10
1.6.2 Métricas de Calidad.....	11
1.6.3 Costos.....	12
1.6.4 Seguridad.....	12
1.6.5 Pruebas del Sistema de Información Web.....	13
1.6.5.1 Modelo V.....	13
1.7 HERRAMIENTAS.....	15
Visual Studio Code.....	15
DBeaver.....	15
Balsamiq Mockups.....	15
Laravel.....	16
Html y Css.....	16
XAMPP.....	16
Ajax.....	16
1.7.1 Gestor de Base de datos:.....	17
1.8 LIMITES Y ALCANCES.....	17
1.8.1 Limites.....	17

1.8.2 Alcances.....	18
1.9 APORTES.....	19
CAPITULO II.....	20
2. MARCO TEORICO.....	20
2.1 INTRODUCCION.....	20
2.2 SISTEMA.....	20
2.3 SISTEMA DE INFORMACION.....	20
2.4 SISTEMA WEB.....	21
2.5 GESTION Y CONTROL DE EMERGENCIAS EN INFRAESTRUCTURAS DE LAS UNIDADES EDUCATIVAS.....	21
2.6 INGENIERIA DE SOFTWARE.....	23
2.7 METODOLOGIA UWE.....	24
2.7.1 Características.....	24
2.7.2 Fases de la Metodología UWE.....	25
2.8 ARQUITECTURA DEL SOFTWARE.....	35
2.8.1 Patrón Modelo Vista Controlador.....	37
2.8.2 Ciclo de Vida.....	39
2.9.1 Gestor de base de datos.....	45
2.9.2 Lenguaje de programación.....	47
2.9.3 Framework.....	50
2.9.4 Herramientas de diseño.....	53
2.10 CALIDAD DEL SOFTWARE.....	55

2.10.1 Estándar ISO/IEC 25000.....	57
2.11.1 COCOMO II.....	62
2.11.1.1 MODELOS DE COCOMO II.....	62
2.11.1.2 SECTORIZACION DE APLICACIONES EN COCOMO II.....	64
2.12 SEGURIDAD DE LA INFORMACION.....	65
2.13 PRUEBAS DE FUNCIONALIDAD DEL SOFTWARE.....	67
2.13.1 Pruebas Unitarias.....	67
2.13.2. Pruebas de Integración.....	69
2.13.3. Pruebas de sistema.....	71
2.14.4 Pruebas de aceptación.....	73
CAPITULO III.....	76
3.1 INTRODUCCION.....	76
3.2 ANALISIS DE LA SITUACION ACTUAL.....	76
3.3 INGENIERIA DE REQUERIMIENTOS.....	78
3.3.1 Obtención de requisitos.....	78
3.3.1.1 Roles de Administración Funcional del Sistema.....	79
3.3.1.2 Definición de actores.....	80
3.3.1.2. Lista de requerimiento del sistema.....	81
3.3.1.2.1. Requerimientos funcionales.....	82
3.3.1.2.2 Requerimientos no Funcionales.....	83
3.3.2 Especificación de requerimientos para la aplicación.....	83

3.4 ANALISIS DE REQUERIMIENTO Y DISEÑO DEL SISTEMA.....	84
3.4.1 Modelo de caso de uso.....	84
3.4.2 Modelo de contenido.....	93
3.4.3. Modelo de Navegación.....	96
3.4.4. Modelo de Presentación.....	97
3.4.4.1 Modelo de Autenticación de Usuario.....	97
3.4.4.2 Modelo de Presentación de la Página Principal.....	97
3.4.4.3 Modelo de Presentación Listado de Registros y Acciones.....	98
3.5. IMPLEMENTACION DEL SISTEMA.....	99
CAPITULO IV.....	107
4. INTRODUCCION.....	107
4.1. METRICAS DE CALIDAD ISO 25000.....	107
4.1.1. Funcionalidad.....	107
4.1.2 Usabilidad.....	112
4.1.3. Eficiencia.....	114
4.1.4 Mantenibilidad.....	116
4.1.5 Portabilidad.....	117
4.1.5. Confiabilidad.....	118
4.1.7 Resultados.....	119
4.2 ANALISIS DE COSTOS.....	119
4.3.1 Seguridad Lógica.....	123

4.3.2 Seguridad Física.....	124
4.4 PRUEBA.....	125
4.4.1. Prueba Unitarias.....	125
4.4.2. Prueba de Integración.....	127
CAPITULO V.....	134
5.CONCLUSIONES Y RECOMENDACIONES.....	134
5.1 CONCLUSIONES.....	134
5.2. RECOMENDACIONES.....	134
REFERENCIAS BIBLIOGRAFICAS.....	137

INDICE DE TABLAS

Tabla 1. Definición de actores.....	80
Tabla 2. Lista de requerimiento del sistema.....	82
Tabla 3. Requerimientos no funcionales.....	83
Tabla 4. Requerimientos para la aplicación.....	83
Tabla 5. Caso de uso de Administrador Principal.....	87
Tabla 6. Caso de uso Inspector Encargado.....	90
Tabla 7. Caso de uso: Encargado de Almacén.....	92
Tabla 8. Tabla Caso de uso Personal de Cuadrilla.....	93
Tabla 9. Número de entradas de Usuario.....	108
Tabla 10. Salidas de Usuario.....	108
Tabla 11. Número de Peticiones del Usuario.....	108
Tabla 12. Parámetros de medición de punto de función.....	109
Tabla 13. Punto de función sin ajustar.....	109
Tabla 14. Factores de Complejidad.....	110
Tabla 15. Parámetros de medición de usabilidad.....	113
Tabla 16. Usabilidad del sistema.....	113
Tabla 17. Escala de valores de eficiencia.....	114
Tabla 18. Valoración de la eficiencia del sistema.....	115
Tabla 19. Resultados de evaluación de calidad.....	119
Tabla 20. Tabla de Linas de Código.....	120
Tabla 21. Coeficientes de a,b,c,d de COCOMO II.....	121
Tabla 22. Ecuaciones del Modelo COCOMO II.....	121
Tabla 23. Conductores para COCOMO II.....	121
Tabla 24. Tabla comparativa.....	135

ÍNDICE DE FIGURAS

Figura 1. Organigrama Institucional del Área en Cuestión.....	3
Figura 2. Diagrama de Casos de Uso.....	29
Figura 3. Modelo de Contenido (UWE): MHC-PMS.....	30
Figura 4. Modelo de Navegación (UWE): MHC-PMS.....	31
Figura 5. Modelo de Estructura de navegación (UWE).....	33
Figura 6. Modelo de Presentación (UWE): MHC-PMS.....	34
Figura 7. Modelo de Presentación Acceso al sistema (UWE).....	34
Figura 8. Flujo Modelo Vista Controlador.....	38
Figura 9. Ciclo de vida de un MVC.....	40
Figura 10. Modelo de ISO 250010 ISO/IEC/ 25000.....	59
Figura 11. Sectorización de aplicaciones en COCOMO II.....	64
Figura 12. Modelo en V.....	67
Figura 13. Modelo de diagrama de caso de uso General del sistema.....	85
Figura 14. Diagrama de caso de uso: Administrador Principal.....	86
Figura 15. Diagrama de caso de uso Inspector Encargado.....	89
Figura 16. Caso de Uso de Encargado Almacén.....	92
Figura 17. Diagrama de Caso de Uso Personal de Cuadrillas.....	93
Figura 18. Modelo de Contenido Base de Datos.....	95
Figura 19. Modelo de Navegación.....	96
Figura 20. Modelo de Navegación Ingreso al sistema.....	97
Figura 21. Modelo de Presentación de Pagina Maestra.....	98
Figura 22. Modelo de Listado de Registros y Acciones.....	99
Figura 23. Interfaz de Inicio de Sesión.....	100

Figura 24. Inicio Principal.....	100
Figura 25. Hoja de Ruta.....	101
Figura 26. Inspecciones.....	101
Figura 27. Requerimientos de Material.....	102
Figura 28. Cronogramas.....	102
Figura 29. Intervenciones.....	103
Figura 30. Almacén.....	103
Figura 31. Trabajos Cuadrillas.....	104
Figura 32. Listado de Personal de la Unidad.....	104
Figura 33. Cuadrillas.....	105
Figura 34. Unidades Educativas.....	105
Figura 35. Acta de Inspección.....	106
Figura 36. Ficha Técnica.....	106
Figura 37. Pruebas Unitarias Personales.....	125
Figura 38. Pruebas Unitarias Unidades Educativas Parte I.....	126
Figura 39. Prueba Unitaria Unidad Educativa Parte II.....	127
Figura 40. Prueba Integración Raíz del Proyecto Parte I.....	128
Figura 41. Prueba Integración Raíz del Proyecto Parte II.....	129
Figura 42. Prueba Integración Raíz del Proyecto Parte III.....	130
Figura 43. Prueba de Integración de Hoja de Ruta Parte I.....	131
Figura 44. Prueba de Integración de Hoja de Ruta Parte II.....	132
Figura 45. Prueba de Integración Intervenciones.....	133

INDICE DE ECUACIONES

(1). Ecuación Punto de Función.....	111
(2). Ecuación Punto de Función Máximo.....	112
(3). Ecuación de Funcionalidad.....	112
(4). Ecuación de Usabilidad.....	112
(5). Ecuación de Eficiencia.....	115
(6). Ecuación de Mantenibilidad.....	116
(7). Portabilidad.....	118
(8). Esfuerzo.....	122
(9). Tiempo de Desarrollo del Sistema.....	122

No se encuentran elementos de tabla de ilustraciones.

LISTADO DE ABREVIATURAS

UWE: Ingeniería web basada en UML.

MOF: Manual de Organización y Funciones.

ISO: Organización Internacional de Normalización que se aplica a los productos y servicios.

HTML: Lenguaje de Marcado.

CSS: Cascading Styles Sheets (Hojas de estilo en cascada)

M.V.C: Modelo vista controlador.

COCOMO: Constructive Cost Model (Modelo constructivo de costos)

RESUMEN

En el Gobierno Autónomo Municipal de El Alto, la Unidad de Mantenimiento enfrenta importantes desafíos en la gestión de solicitudes de emergencia relacionadas con problemas de infraestructura en las unidades educativas. Los procesos manuales, la dispersión de la información y las demoras significativas en la atención de solicitudes han generado insatisfacción y riesgos en la comunidad educativa. Este proyecto tiene como objetivo desarrollar un sistema de información web centralizado que permita registrar y gestionar eficientemente las emergencias de infraestructura. La metodología utilizada para el desarrollo del sistema es UWE (UML-based Web Engineering), que ofrece un enfoque iterativo y sistemático para garantizar un diseño robusto y adaptable a las necesidades del usuario. Además, el sistema empleará tecnologías modernas como PostgreSQL para la gestión eficiente de bases de datos y PHP para garantizar un desarrollo escalable y seguro. El sistema incluye módulos para registrar solicitudes de emergencia, asignar recursos, monitorear el progreso de las intervenciones y generar informes detallados. Estas funcionalidades no solo mejoran la eficiencia operativa, sino que también refuerzan la confianza de la comunidad educativa y promueven un entorno seguro para estudiantes y personal. Este proyecto representa un avance significativo hacia la modernización de la gestión de emergencias en El Alto, ofreciendo un sistema confiable y sostenible que responde a las necesidades de la Unidad de Mantenimiento y contribuye al bienestar de la comunidad educativa.

ABSTRACT

In the Autonomous Municipal Government of El Alto, the Maintenance Unit faces significant challenges in managing emergency requests related to infrastructure issues in educational institutions. Manual processes, dispersed information, and considerable delays in addressing requests have caused dissatisfaction and posed risks to the educational community. This project aims to develop a centralized web-based information system to efficiently record and manage infrastructure emergencies. The methodology used for system development is UWE (UML-based Web Engineering), which provides an iterative and systematic approach to ensure a robust and adaptable design tailored to user needs. Additionally, the system will utilize modern technologies such as PostgreSQL for efficient database management and PHP to guarantee scalable and secure development. The system includes modules for registering emergency requests, allocating resources, monitoring the progress of interventions, and generating detailed reports. These functionalities not only enhance operational efficiency but also strengthen the trust of the educational community and promote a safe environment for students and staff. This project represents a significant step toward modernizing emergency management in El Alto, offering a reliable and sustainable system that meets the needs of the Maintenance Unit and contributes to the well-being of the educational community.

CAPITULO I

Marco Preliminar

INGENIERIA
DE SISTEMAS
UNIVERSIDAD PUBLICA DE EL ALTO



CAPITULO I

1. INTRODUCCION

En el contexto de las unidades educativas de El Alto, la gestión de emergencias de infraestructura presenta numerosos desafíos que requieren una solución tecnológica eficiente y robusta. El actual registro y control de emergencias de infraestructura dificulta la coordinación y respuesta rápida ante solicitudes críticas, como daños estructurales, fallas en sistemas eléctricos o problemas en las instalaciones sanitarias. Esta situación expone a estudiantes y personal educativo a riesgos significativos.

La gestión actual de emergencias en las unidades educativas de El Alto enfrenta importantes desafíos en términos de coordinación y respuesta oportuna. La dispersión de los registros y la dependencia de procesos manuales dificultan la comunicación efectiva entre el personal educativo, las autoridades y los servicios de emergencia. Esta fragmentación puede generar retrasos en la respuesta, exponiendo tanto a estudiantes como al personal a riesgos innecesarios. Además, el actual acceso a información en dificulta la gestión de solicitudes quedando solicitudes inconclusas, limita la capacidad de tomar decisiones informadas y de planificar acciones preventivas de manera eficiente.

Para abordar estos problemas, se desarrollará un sistema de información web centralizado para el registro, monitoreo y control eficiente de emergencias de infraestructura en las unidades educativas de El Alto, que permitirá registrar incidentes de manera rápida y organizada, facilitando el acceso a información crucial en tiempo real. Asimismo, mejorará los servicios de mantenimiento, disminuyendo el número de casos no atendidos.

Para gestionar el desarrollo del sistema, se utilizará la metodología UWE (UML-based Web Engineering), que permitira una entrega continua de valor y la adaptación a cambios en los requisitos a través de iteraciones. El sistema se basará en tecnologías modernas para asegurar

su eficiencia y seguridad. PostgreSQL como gestor de base de datos que centralizará el registro de solicitudes, permitiendo un acceso rápido y organizado a la información.

1.2 ANTECEDENTES

1.2.1. Antecedentes Institucionales

El Gobierno Autónomo Municipal de El Alto G.A.M.E.A como Institución Pública, tiene como una de sus finalidades la satisfacción de las necesidades de la población y la mejora de sus condiciones de vida. Para lograr dicha finalidad se necesita crear y construir una Estructura Organizativa que le permita responder de forma efectiva a las demandas de la población. Es así, que el Manual de Organización y Funciones (MOF) como herramienta administrativa contribuye al logro del objetivo institucional al establecer las funciones las relaciones de autoridad, dependencia, supervisión y coordinación entre las distintas Unidades Organizacionales del Órgano Ejecutivo del Gobierno Autónomo Municipal de El Alto.

1.2.1.1 Misión del Gobierno Autónomo Municipal de El Alto

El Alto ciudad emergente con alma y espíritu de unidad, con planificación del bienestar social, participativo, desconcentrado, incluyente que garantiza derechos, seguridad del espacio público, con acceso a la salud, educación y protagonista del desarrollo nacional.

1.2.1.2 Visión del Gobierno Autónomo Municipal de El Alto

Gobierno autónomo municipal de el alto, con identidad, ciudad ajayu con derecho a la salud, educación y sin violencia, con empleo, modernidad, industrias y emprendimientos. para el desarrollo sustentable y planificado, articulador de la región metropolitana.

1.2.1.2 Objetivo

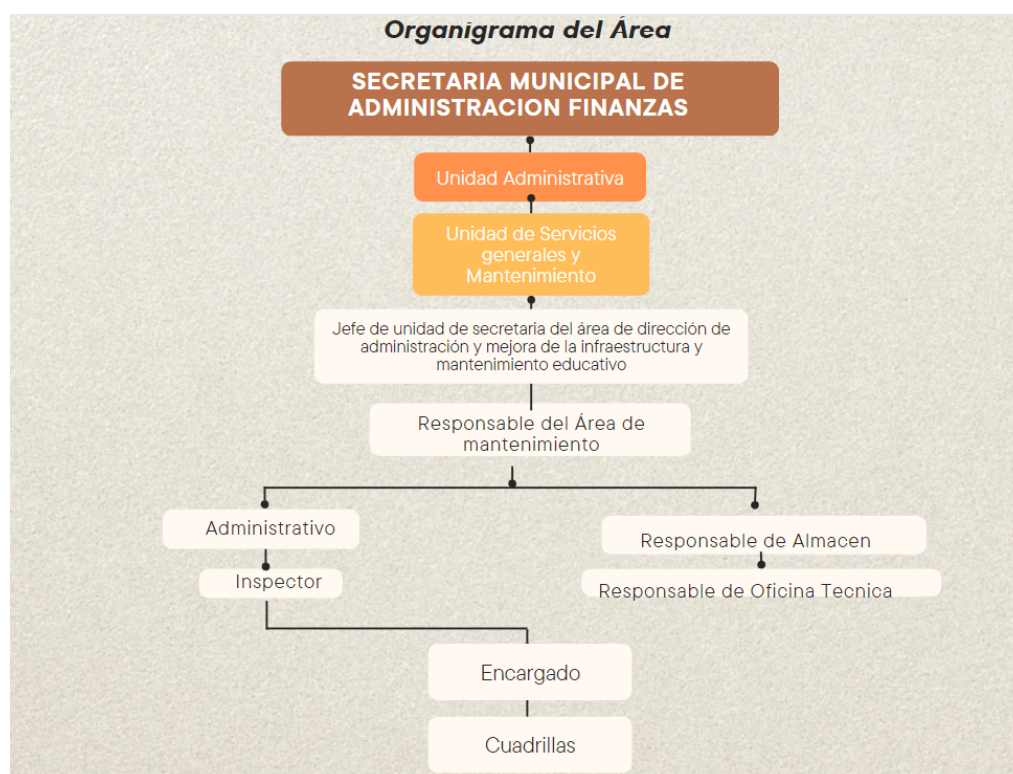
El Gobierno Autónomo Municipal de El Alto tiene como objetivo fundamental promover el desarrollo humano integral de la población, enfocándose en educación, salud, cultura, deporte y recreación, velando por la ciudadanía alteña por la juventud y los estudiantes de las unidades

educativas. Además, busca fomentar un desarrollo económico sostenible mediante la diversificación productiva, el emprendimiento y la generación de empleo digno. En materia de seguridad ciudadana, trabaja para fortalecer la prevención, reducir la criminalidad y promover una convivencia pacífica. Asimismo, prioriza la gestión eficiente y transparente de los servicios públicos, asegurando calidad, inclusividad y sostenibilidad bajo principios de transparencia y compromiso social. Finalmente, apuesta por la preservación del medio ambiente y el desarrollo sostenible, protegiendo y gestionando los recursos de manera responsable.

1.2.1.3 Organigrama

Figura 1.

Organigrama Institucional del Área en Cuestión



1.2.2 Antecedentes afines al proyecto de grado

1.2.2.1 Antecedentes Internacionales

- (Luigui, Chipulina Puelles, 2018) "Sistema web para la gestión de incidencias en la empresa consultit s.a.c."

Objetivo General: El objetivo principal fue establecer el flujo de un sistema web en la gestión de incidencias en la empresa Consultit esta exploración es de tipo de estudio fue aplicada y el diseño de investigación es pre – experimental. El método de investigación a usar fue aplicado – deductivo. La metodología elegida a usar fue SCRUM, el software se desarrolló en PHP, teniendo como motor de base de datos MariaDB, los indicadores a evaluar fueron, la ratio de resolución de incidencias y la ratio de incidencias reabiertas. universidad cesar vallejo facultad de ingeniería escuela académico profesional de ingeniería de sistemas.

- (Buitrago, Daniel Ramirez Duque y Jhonatan Cardona, 2021) "Desarrollo de software para la atención de siniestros por parte de entes estatales en la ciudad de Pereira".

Objetivo General: Desarrollar una aplicación de atención de siniestros por parte de entes estatales en la ciudad de Pereira. Universidad Católica de Pereira programa de ingeniería en sistemas y telecomunicaciones facultad de ciencias básicas e ingeniería.

- (Natividad, Bach. Boggio Chanduvi Olga, 2017) "Diseño de sistema de gestión de incidencias informáticas en de Ingeniería SAC. Lima - Perú"

Objetivo General: Diseñar el sistema de gestión de incidencias informáticas en la empresa DSE Ingeniería SAC – Lima. Universidad Nacional Pedro Ruiz Gallo Facultad de Ingeniería Civil, Sistemas y Arquitectura Escuela Profesional de Ingeniería de Sistemas

- (Barbara Bron Foncesa, Katy Mestre y William Simon, 2019) “Sistema informático para la gestión de reportes de incidencias de mantenimiento en la Facultad de Ciencias y Tecnologías Computacionales”

Objetivo Generales: desarrollar un sistema informático que gestione los reportes de incidencias de mantenimiento generados en la facultad, para que sean atendidos de manera interna. Para el análisis, diseño e implementación del sistema web, la autora optó por la utilización de la metodología AUP-UCI en su escenario número cuatro, se utilizó la herramienta Visual-Paradigm para el modelado. Como lenguajes de programación PHP y Java Script; Symfony como marco de trabajo, NetBeans como Entorno de Desarrollo Integrado, Apache como servidor web y MySQL como sistema gestor de base de datos. Como resultado, se obtuvo un sistema capaz de gestionar los reportes de incidencias, generados dentro de la facultad. Universidad de las Ciencias Informáticas. La Habana, Cuba

1.2.2.2 Antecedentes Nacionales

- (Claudia, Cuevas Coyo Gladys, 2020) “Sistema de Información Web para el control y seguimiento académico”.

Objetivo General: Desarrollar un “Sistema de Información Web para el Control y Seguimiento Académico” para el desenvolvimiento del Estudiante en el año escolar, cumpliendo todas las exigencias de la Unidad Educativa “Huayna Potosí - Mañana”. Brindando información actualizada al Padre de Familia. Universidad Pública de El Alto carrera Ingeniería de Sistemas

- (Gahuapaza, Heriberto Condori, 2020) Sistema Web de gestión y control de almacenes” caso: Unidad de servicios generales y mantenimiento (GAMEA).

Objetivo General: Desarrollar un sistema Web de gestión y control que coadyuve

con la sistematización de información y el seguimiento de almacenes para la Unidad de Servicios Generales y Mantenimiento dependiente del GAMEA, que permita integrar los procesos de entradas, salidas, control de stock, fechas de vencimiento de los materiales y poder generar información en tiempo real cuando se lo requiera. Universidad Pública de El Alto carrera Ingeniería de Sistemas

1.3 PLANTEAMIENTO DEL PROBLEMA

En las unidades educativas de la ciudad de El Alto, la gestión de emergencias de infraestructura presenta múltiples desafíos críticos que afectan la seguridad y el bienestar de estudiantes y personal. Actualmente se tiene la información dispersa, lo que dificulta el seguimiento de las solicitudes críticas según información del responsable de área de mantenimiento de infraestructuras los casos de incidentes aumentan en temporada de lluvias.

1.3.1 Problema Principal

En el Gobierno Autónomo Municipal de El Alto, la Unidad de Mantenimiento enfrenta importantes desafíos en la gestión y el control de solicitudes relacionadas con problemas de infraestructura en las unidades educativas. Los procesos manuales, el seguimiento a las solicitudes dispersa, las demoras significativas en la atención de las solicitudes, incluso llegando a no atender solicitudes, han generado insatisfacción y riesgos en las unidades educativas ya que si la solicitud no sería atendida los estudiantes corren riesgos. Esta situación ha llevado a que la unidad enfrente demandas debido a respuestas inconclusas a las solicitudes recibidas.

1.3.2 Problemas secundarios

- El control no es óptimo sobre los trabajos realizados, la gestión actual de las intervenciones de emergencias en las unidades educativas presenta deficiencias en el procesamiento de información, lo cual genera demoras en la atención de

solicitudes.

- Proceso manual la manipulación constante de documentos en formato físico dificulta su búsqueda y acceso, por lo que el seguimiento es tardío y muchas veces sin respuesta.
- Perdida de la confianza por parte de los representantes de las Unidades educativas debido a la demoras o inconclusión del trabajo.
- La emisión de informes incompletos y la pérdida de información son problemas comunes derivados de la dispersión de los datos, una situación que se agrava aún más cuando los documentos físicos se extravían, sin contar con respaldos que permitan su recuperación o consulta.

1.3.3 Formulación del problema

¿Cómo coadyuvar para una eficiente gestión de documentos optimizar el seguimiento de las solicitudes de las hojas de rutas en relación a las emergencias en unidades educativas y mejorar la toma de decisiones para la unidad de emergencias del Gobierno Autónomo de El Alto?

1.4 OBJETIVOS

1.4.1 Objetivo General

Desarrollar un Sistema Web para el Registro, Control y Seguimiento de Solicitudes Relacionadas con Problemas de Infraestructura en las Unidades Educativas de la Ciudad de El Alto, que optimice los procesos, genere información confiable, para la correcta toma de decisiones en el Gobierno Autónomo Municipal de El Alto.

1.4.2 Objetivos Específicos

- Realizar un análisis de la situación actual del proceso de seguimiento de solicitudes relacionadas con problemas de infraestructura para determinar los

requerimientos.

- Centralizar la información relevante en una base de datos para responder a las solicitudes y el proceso de trabajo de la Unidad de Mantenimiento de forma efectiva.
- Sistematizar los procesos operativos¹ para su respectivo seguimiento.
- Generar informes detallados de los procesos operativos en respuesta a las solicitudes recibidas, de forma oportuna.

1.5 JUSTIFICACION

1.5.1 Justificación Técnica

La implementación de un Sistema de Información Web para la gestión de emergencias de infraestructura en las unidades educativas de El Alto responde a la necesidad de contar con un sistema optimo.

Para desarrollar el sistema se implementó herramientas como ser PHP y PostgreSQL las mismas que permiten el acceso y la manipulación de la información de manera fácil y segura.

Las metodologías y herramientas a emplear para el estudio y desarrollo de un sistema de información, son recursos disponibles y al alcance de cualquier institución o persona en cuanto a las herramientas de desarrollo de software están disponibles por ser gratuitos y de código abierto y es posible adecuar a las necesidades requerimientos y disponibilidades que cuenta la institución.

El GAMEA dispone de hardware y software para la implementación técnica de este

¹ Los procesos operativos refieren al registro de solicitudes de mantenimiento, inspecciones requerimientos de material e intervenciones.

proyecto, destaca la necesidad de un sistema integrado que optimice la respuesta a emergencias de infraestructura y la gestión de incidentes en las unidades educativas de El Alto. La utilización de servicios en la nube garantizará la escalabilidad y disponibilidad del sistema, mientras que la capacitación del personal educativo y el soporte técnico continuo asegurarán su correcta implementación y operación, este proyecto busca no solo mejorar la gestión de emergencias de infraestructura, sino también contribuir a la creación de un entorno educativo más seguro y preparado para enfrentar situaciones críticas.

1.5.2 Justificación Económica

Desde una perspectiva económica, la implementación de un sistema de información web para la gestión de emergencias representa una inversión estratégica a largo plazo. Si bien la creación y puesta en marcha inicial del sistema implicarán costos asociados al desarrollo de software. Asimismo, la mejora en la coordinación y comunicación entre los actores involucrados permitirá una asignación más eficiente de recursos, evitando duplicidades y optimizando el uso de personal y equipos. Además, la centralización del registro de incidentes facilitará la generación de reportes y análisis, proporcionando datos valiosos para la toma de decisiones y la planificación evitando posibles desastres. En consecuencia, la inversión inicial en el desarrollo e implementación del sistema se traducirá en ahorros significativos a largo plazo y en una gestión más eficiente y efectiva de emergencias en el ámbito educativo.

1.5.3 Justificación Social

Proporcionar un entorno seguro y preparado para responder ante cualquier situación de emergencia es fundamental para promover el bienestar y desarrollo integral de estudiantes, personal educativo y familias. La mejora en la capacidad de respuesta ante emergencias contribuirá a generar confianza y tranquilidad en la comunidad, fortaleciendo el vínculo entre la institución educativa y la sociedad. Además, la promoción de la participación y colaboración de la comunidad y los padres de familia en la gestión de emergencias fomentará una cultura de

prevención y solidaridad, donde todos los actores se involucren activamente en la protección y seguridad de los estudiantes, la implementación de este sistema no solo tiene un impacto directo en la seguridad y bienestar de la comunidad educativa, sino que también contribuye a fortalecer el tejido social y promover una cultura de prevención y resiliencia ante situaciones de emergencia.

1.6 METODOLOGIA

El desarrollo del sistema es una tarea compleja que requiere una planificación y organización estructurada para lograrlo, se utilizan metodologías que facilitan la creación, implementación y depuración del sistema la elección adecuada de estas metodologías proporciona una estructura y enfoque sistemático mejorando el proceso y facilitando la interacción con los usuarios y demás involucrados, En este caso se emplean las siguientes metodologías.

1.6.1 Metodología UWE

La metodología UWE (UML-based Web Engineering) es una metodología para el desarrollo de aplicaciones web que se basa en el uso del Lenguaje de Modelado Unificado (UML). UWE proporciona un enfoque sistemático y orientado a modelos para el diseño, implementación y mantenimiento de aplicaciones web. UWE es una metodología de ingeniería web que utiliza UML para modelar y diseñar aplicaciones web. Se centra en varios aspectos clave del desarrollo web, incluyendo la navegación, la presentación, la lógica de negocios y la adaptación. La metodología UWE sigue un enfoque iterativo e incremental, permitiendo una evolución progresiva del sistema web.

Componentes de UWE

- **Modelo de Requisitos:** Define los requisitos funcionales y no funcionales del sistema. Incluye casos de uso y escenarios que describen la interacción del

usuario con el sistema.

- **Modelo de Contenidos:** Especifica los datos y la estructura de la información que se gestionará en la aplicación web. Utiliza diagramas de clases para representar entidades y relaciones.
- **Modelo de Navegación:** Describe la estructura de navegación de la aplicación, mostrando cómo los usuarios pueden moverse entre diferentes partes del sistema. Utiliza diagramas de navegación que incluyen nodos de navegación y enlaces.
- **Modelo de Presentación:** Define la apariencia y el diseño de la interfaz de usuario. Utiliza diagramas de clases y diagramas de estado para modelar las vistas y las interacciones de la interfaz.
- **Modelo de Proceso:** Detalla los procesos de negocio y las reglas de negocio que la aplicación debe implementar. Utiliza diagramas de actividad y diagramas de secuencia para modelar los procesos dinámicos.
- **Modelo de Adaptación:** Se ocupa de la personalización y adaptación del sistema web para diferentes usuarios o contextos de uso. Incluye reglas de adaptación y variabilidad.

1.6.2 Métricas de Calidad

La medición de la calidad del software es esencial para el sistema desarrollado ya que evaluara su rendimiento y asegura que cumpla con los estándares de expectativa establecidos.

Modelo de Calidad ISO/IEC 25010: El modelo de calidad ISO/IEC 25010 se adopta en el desarrollo del sistema de información web para la gestión de emergencias en unidades

educativas de El Alto para asegurar un software de alta calidad y rendimiento. Este modelo evalúa la adecuación funcional, garantizando que las funcionalidades cumplan con los requisitos; la fiabilidad, asegurando la madurez, disponibilidad y tolerancia a fallos del sistema; y la usabilidad, midiendo la facilidad de entendimiento, aprendizaje y operabilidad para los usuarios. Además, se considera la eficiencia de desempeño, enfocándose en el comportamiento temporal y la utilización de recursos; la mantenibilidad, facilitando el análisis, cambio y estabilidad del sistema; y la seguridad, protegiendo la confidencialidad, integridad y autenticidad de los datos. Finalmente, se evalúa la compatibilidad, asegurando la coexistencia e interoperabilidad con otros sistemas. Estas métricas proporcionan una evaluación integral y continua, garantizando que el sistema sea robusto, eficiente y seguro, contribuyendo a un entorno educativo más seguro y gestionado eficazmente en El Alto.

1.6.3 Costos

El método COCOMO II (Constructive Cost Model) se empleará para estimar los costos del desarrollo del sistema de información web para la gestión de emergencias en unidades educativas de El Alto, proporcionando una estimación basada en el tamaño del proyecto, la complejidad del software y otros factores ajustables. Este modelo cuantifica el esfuerzo de desarrollo a través de ecuaciones que consideran aspectos como el tamaño estimado del código (en KLOC, líneas de código) y factores de ajuste que reflejan características del proyecto y el equipo, como la experiencia y el uso de herramientas. Al utilizar COCOMO II, se obtiene una estimación detallada del esfuerzo y tiempo de desarrollo, facilitando la planificación de recursos, la asignación de presupuesto y la gestión de plazos, lo cual contribuye a una relación costo-beneficio equilibrada y permite una toma de decisiones informada durante todo el ciclo de vida del proyecto.

1.6.4 Seguridad

La norma ISO/IEC 27001 (A.16) se centra en la gestión de incidentes de seguridad de la

información, proporcionando un marco esencial para garantizar la protección de datos y la continuidad del servicio en el desarrollo del sistema de información web para la gestión de emergencias en unidades educativas de El Alto. Implementar esta norma implica establecer políticas y procedimientos claros para la identificación, reporte, análisis y resolución de incidentes, así como definir roles y responsabilidades específicas. El uso de herramientas avanzadas de monitoreo y sistemas de alerta temprana asegura la detección en tiempo real de actividades sospechosas, mientras que la comunicación eficaz y la notificación a todas las partes relevantes garantizan una respuesta coordinada. Además, mantener un registro detallado y analizar los incidentes facilita la mejora continua de las medidas de seguridad. Capacitar al personal y fomentar una cultura de seguridad dentro de la comunidad educativa son fundamentales para prevenir incidentes y proteger la información sensible, asegurando así un entorno seguro y resiliente.

La norma ISO/IEC 27002 proporciona directrices prácticas para la implementación de controles de seguridad de la información, ofreciendo un conjunto de buenas prácticas que refuerzan la protección de datos y la gestión de riesgos en el sistema de información web para emergencias en unidades educativas de El Alto. Esta norma detalla medidas específicas para garantizar la seguridad en áreas clave como el control de accesos, la gestión de activos de información, la protección de la infraestructura y la seguridad en las comunicaciones. Al aplicar estos controles, el sistema se fortalece frente a posibles vulnerabilidades y amenazas, optimizando el manejo de datos sensibles y promoviendo la resiliencia operativa. La norma también subraya la importancia de la concienciación en seguridad y la formación del personal, asegurando que todos los actores involucrados comprendan y sigan las prácticas de protección de la información, contribuyendo a un entorno seguro y confiable.

1.6.5 Pruebas del Sistema de Información Web

1.6.5.1 Modelo V

El modelo V o modelo en cuatro niveles es un modelo empleado en diversos procesos de desarrollo, por ejemplo, en el desarrollo de software. En los años 90 apareció su primera versión, pero con el tiempo se ha ido perfeccionando y adaptando a los métodos modernos de desarrollo. La idea básica, sin embargo, se remonta a los años 70 y fue concebida como una especie de desarrollo posterior del modelo de cascada.

Además de las fases de desarrollo de un proyecto, el modelo V también define los procedimientos de gestión de la calidad que lo acompañan y describe cómo pueden interactuar estas fases individuales entre sí. Su nombre se debe a su estructura, que se asemeja a la letra V.

En primer lugar, el modelo V define el curso de un proyecto en fases individuales cada vez más detalladas:

Al principio del proyecto, el modelo prevé un análisis de las especificaciones del sistema planificado (fase de especificaciones).

El proyecto se completa después con requisitos funcionales y no funcionales para la arquitectura del sistema (fase funcional).

A esta fase le sigue el diseño del sistema, en el que se planifican los componentes y las interfaces de este (fase de diseño).

Una vez completadas estas fases, se puede diseñar en detalle la arquitectura del software (codificación).

Es ahora cuando, de acuerdo con estos planes, comienza el desarrollo en sí del software. A continuación, tendrán lugar las fases de control de la calidad, también llamadas de

verificación o validación, que siempre están relacionadas con cada una de las fases de desarrollo. El método V abarca las siguientes tareas:

- Pruebas de unidad
- Pruebas de integración
- Pruebas del sistema
- Pruebas de aceptación

1.7 HERRAMIENTAS

El proceso de desarrollo de sistemas web se aplican diferentes tecnologías de desarrollo, asimismo herramientas de software libre.

Visual Studio Code es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con Git, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación (Microsoft, s.f.).

DBeaver es una herramienta para el desarrollo backend es aquella que nos permite administrar bases de datos mediante interfaz gráfica. DBeaver. (s.f.).

DBeaver es una herramienta gratuita y multiplataforma para trabajar con las bases de datos. Soporta todas las bases de datos como MySQL, PostgreSQL, Oracle, MS SQL Server, MS Access, SQLite y otras.

Balsamiq Mockups es una herramienta para crear prototipos, bocetos o wireframes es Balsamiq Mockups.

Es un programa de escritorio, es decir, solo tienes que registrarte para poder empezar a utilizarlo sin ningún tipo de descarga.

Laravel un framework PHP de código abierto que se destaca por su elegancia, eficiencia y facilidad de uso en el desarrollo de aplicaciones web modernas. Basado en el patrón de arquitectura MVC, Laravel ofrece una amplia gama de características poderosas, como enrutamiento flexible, sistema de plantillas Blade intuitivo y poderoso, ORM Eloquent para la interacción con bases de datos, sistema de migraciones para el control de versiones de la base de datos, y una sólida capa de seguridad que incluye protección CSRF, autenticación y autorización avanzadas. Además, Laravel cuenta con una comunidad activa y una amplia documentación que hacen que sea fácil aprender y utilizar, lo que lo convierte en una opción popular para el desarrollo rápido y eficiente de aplicaciones web escalables y seguras.

Html y Css en conjunto, HTML y CSS son fundamentales para crear páginas web visualmente atractivas, funcionales y responsivas, permitiendo estructurar el contenido y darle el estilo deseado para ofrecer una experiencia de usuario agradable y efectiva.

XAMPP es un paquete de software que facilita la creación de entornos de desarrollo web local. Incluye varios componentes esenciales para el desarrollo web, como Apache como servidor web, MySQL como sistema de gestión de bases de datos, PHP como lenguaje de programación para el desarrollo dinámico de páginas web, y Perl como otro lenguaje de programación común en entornos web. Además de estos componentes principales, XAMPP también puede incluir otros como phpMyAdmin para la gestión de bases de datos MySQL de manera gráfica y sencilla, así como OpenSSL para la seguridad en las comunicaciones web, XAMPP proporciona un entorno de desarrollo local completo y fácil de configurar para desarrolladores web, permitiendo la creación y prueba de aplicaciones web de manera eficiente y segura antes de su despliegue en entornos de producción en línea.

Ajax que significa Asynchronous JavaScript and XML (JavaScript asíncrono y XML), es una técnica de programación web que permite la comunicación entre el navegador web y el servidor de forma asíncrona, sin necesidad de recargar la página completa. Esto se logra utilizando una combinación de tecnologías como HTML, CSS, JavaScript y XML (aunque hoy en día se utiliza más comúnmente JSON en lugar de XML).

Es una herramienta poderosa para crear aplicaciones web más dinámicas, interactivas y eficientes, mejorando la experiencia del usuario al proporcionar respuestas rápidas y actualizaciones de contenido sin recargar toda la página.

Leaflet es una biblioteca de JavaScript de código abierto utilizada para crear mapas interactivos en aplicaciones web. Es ligera, simple y flexible, lo que la hace ideal para una amplia variedad de proyectos de mapas web. Leaflet soporta características como zoom, desplazamiento, superposición de capas, y la adición de marcadores y popups. Es compatible con muchos servicios de mapas populares, como OpenStreetMap, y puede integrarse fácilmente con otras bibliotecas y tecnologías web. Agafonkin, V. (2023)

1.7.1 Gestor de Base de datos:

PostgreSQL es un sistema de gestión de bases de datos relacionales de código abierto que destaca por su soporte SQL estándar, integridad de datos, transacciones ACID, capacidad para funciones y procedimientos almacenados, escalabilidad, extensiones y complementos, así como características de seguridad avanzadas. Es una opción confiable y eficiente para el almacenamiento y administración de datos en diversas aplicaciones, ofreciendo robustez, escalabilidad y una amplia gama de funcionalidades (The PostgreSQL Global Development Group, s.f.).

1.8 LIMITES Y ALCANCES

1.8.1 Limites

El Sistema Web tendrá las siguientes limitaciones:

- El presente proyecto se limita a optimizar el control y seguimiento de las solicitudes, así como el proceso de trabajo hasta tu culminación del mismo.
- El sistema no contendrá la parte contable o financiera.
- El sistema no podrá realizar solicitudes directas en línea por parte de los solicitantes representantes de unidades educativas.

1.8.2 Alcances

Los alcances del proyecto son los siguientes:

- Registro detallado de solicitudes de emergencias en infraestructuras el sistema permitirá registrar información detallada sobre diferentes tipos de emergencias, incluyendo descripción, ubicación y otros datos relevantes.
- Gestión de recursos se desarrollarán herramientas para asignar y gestionar recursos necesarios para cada emergencia, como personal y material de requerimiento.
- Análisis de datos se incluirán herramientas de análisis de datos para identificar patrones permitiendo tomar decisiones informadas y planificar acciones preventivas.
- Generación de informes detallados sobre las solicitudes atendidas, problemas encontrados, materiales requeridos, tipo de trabajo y resultados obtenidos.

- Seguridad se implementarán medidas de seguridad para proteger los datos registrados y garantizar el acceso autorizado a la información.
- Interfaz amigable desarrollo de una interfaz intuitiva y fácil de usar para que los usuarios puedan registrar, gestionar y acceder a información relevante de manera rápida y eficiente.

1.9 APORTES

- El proyecto de desarrollo del sistema de información web para el registro y control de emergencias en unidades educativas de El Alto tiene como objetivo principal contribuir significativamente en varios aspectos:
- Mejora en la respuesta a solicitudes de emergencias en infraestructura: Al integrar un sistema eficiente de reportesin y gestión, se espera una notable mejora en la capacidad de respuesta ante situaciones críticas, lo que puede reducir el tiempo de actuación y minimizar los impactos negativos.
- Optimización de recursos: La gestión eficaz de recursos como personal, equipos y material, facilitada por el sistema, conlleva a una mejor asignación y utilización de los mismos, mejorando así la eficiencia operativa.
- Análisis y mejora continua: Los análisis de datos y la generación de informes proporcionarán información valiosa para identificar áreas de mejora, patrones de emergencias y tendencias, lo que facilitará la toma de decisiones informadas y la planificación estratégica para futuras situaciones de emergencia.

Estos aportes buscan no solo mejorar la gestión de emergencias en las unidades educativas, sino también promover un entorno más seguro, eficiente y preparado

para enfrentar situaciones de manera efectiva.

CAPITULO II

Marco Teorico



CAPITULO II

2. MARCO TEORICO

2.1 INTRODUCCION

El presente proyecto de grado se inicia con la recopilación de la información posible sobre sistemas web definiciones y conceptos empleadas para el desarrollo del presente proyecto.

2.2 SISTEMA

Un sistema es un conjunto de elementos o componentes interrelacionados que interactúan entre sí para lograr un objetivo común. Estos elementos pueden ser personas, procesos, datos, equipos o cualquier otro recurso que trabaje en conjunto dentro de un entorno definido. Los sistemas suelen tener entradas, procesos y salidas, y están organizados para alcanzar un fin específico de manera eficiente.

De acuerdo con Ludwig von Bertalanffy, uno de los principales exponentes de la Teoría General de Sistemas (TGS), un sistema es “un conjunto de elementos interrelacionados entre sí que forman un todo complejo”. Según (Bertalanffy, 1968), un sistema no solo consiste en la suma de sus partes, sino también en las interacciones entre ellas, lo que le otorga una mayor complejidad y capacidad para adaptarse a su entorno. En el contexto de este proyecto, el sistema de información web propuesto tiene como objetivo integrar diferentes procesos para gestionar eficientemente las emergencias en infraestructuras educativas, permitiendo una mayor coordinación y respuesta oportuna.

2.3 SISTEMA DE INFORMACION

Un sistema de información es un conjunto de componentes interrelacionados que recopilan, procesan, almacenan y distribuyen información para apoyar la toma de decisiones, la coordinación y el control dentro de una organización. Además, los sistemas de información ayudan a los gerentes y empleados a analizar problemas, visualizar temas complejos y crear

nuevos productos ((Laudon, 2012):

2.4 SISTEMA WEB

Un sistema de información es un conjunto integrado de componentes que trabajan en conjunto para recolectar, procesar, almacenar y distribuir información. Estos sistemas combinan elementos tecnológicos (hardware y software) con personas y procedimientos para convertir los datos en información útil, facilitando la toma de decisiones y la ejecución eficiente de actividades dentro de una organización.

Según (Montilva, 1999) un sistema de información es un sistema hombre-máquina diseñado para registrar y procesar datos relacionados con las transacciones y entidades de una organización, proporcionando información crítica para su funcionamiento. Por su parte, (James A. Senn, 1978) describe el sistema de información como un sistema basado en computadoras que organiza los datos de tal forma que puedan ser utilizados eficazmente en la toma de decisiones.

En el contexto actual, los sistemas de información son esenciales para las organizaciones debido al creciente volumen y complejidad de los datos que manejan. Estos sistemas no necesariamente deben estar basados en computadoras, aunque con frecuencia lo están, ya que la tecnología permite mejorar significativamente la eficiencia en el manejo de grandes volúmenes de información y en la realización de procesos complejos.

Para este proyecto, un sistema de información web permite automatizar tareas que antes se realizaban de manera manual, como la generación de listas, la recepción de documentos, y la emisión de informes, logrando que estos procesos se realicen de manera más rápida y eficiente, y brindando resultados confiables tanto para el personal como para la comunidad.

2.5 GESTION Y CONTROL DE EMERGENCIAS EN INFRAESTRUCTURAS DE LAS UNIDADES EDUCATIVAS

La gestión involucra varios conceptos como:

REGISTRO El concepto de registro hace referencia a la acción de documentar o dejar constancia de un evento, situación o acción que se ha llevado a cabo. En el contexto del sistema de gestión de emergencias, el registro implica documentar cualquier incidente relacionado con la infraestructura educativa, como fallos estructurales, problemas de servicios o desastres naturales, asegurando que se disponga de una base de datos completa y actualizada para su posterior análisis y seguimiento.

Un registro eficiente proporciona un historial de emergencias que facilita la toma de decisiones y la asignación de recursos de manera efectiva. Esto es clave para la planificación de intervenciones rápidas y la mejora continua de la infraestructura, permitiendo prevenir futuras emergencias o minimizar su impacto. Según (Ifonso Barca Lozano, 2008) registrar un incidente permite un mayor control sobre la información, lo cual es esencial en la gestión de emergencias, ya que cada detalle registrado puede ser crucial para coordinar la respuesta adecuada.

CONTROL El concepto de control se refiere a las acciones y mecanismos implementados para asegurar que los protocolos y procedimientos diseñados para gestionar emergencias se lleven a cabo de manera adecuada. El control en el ámbito de las emergencias de infraestructuras educativas implica supervisar, inspeccionar y regular las acciones tomadas para enfrentar y solucionar cualquier incidente reportado, con el fin de garantizar que se cumplan los estándares de seguridad y eficiencia.

El objetivo principal del control es asegurar que las emergencias se gestionen de manera efectiva, minimizando daños y riesgos. También busca identificar cualquier falla en los procesos o desviaciones de los planes establecidos para corregirlos a tiempo, evitando así que

se repitan en futuras situaciones el control permite la regulación y fiscalización de los procedimientos, asegurando que las decisiones se tomen de acuerdo con las normativas y objetivos establecidos.

INFRAESTRUCTURA La infraestructura en el contexto de este proyecto se refiere a los edificios, servicios y estructuras físicas que componen las unidades educativas. El mantenimiento y supervisión de esta infraestructura es clave para prevenir situaciones de emergencia que puedan afectar la seguridad de los estudiantes y el personal.

El control de la infraestructura educativa incluye la monitorización constante del estado de los edificios, los servicios de agua, electricidad, y demás recursos críticos que afectan el funcionamiento seguro de la escuela. Las emergencias relacionadas con infraestructura pueden ser tanto de tipo estructural (daños en los edificios) como de servicios (falta de agua, electricidad, plomería, albañilería, techos, etc.).

2.6 INGENIERIA DE SOFTWARE

La Ingeniería de Software es una rama de la informática que se enfoca en la aplicación de principios, metodologías y técnicas de ingeniería para el diseño, desarrollo, mantenimiento y gestión de sistemas de software de manera eficiente y confiable. Su principal objetivo es crear software de alta calidad que cumpla con los requisitos establecidos, tanto funcionales como no funcionales, y que sea fácil de mantener y evolucionar a lo largo del tiempo.

La Ingeniería de Software no solo abarca el proceso de programación, sino que también incluye la planificación, diseño, pruebas, implementación y mantenimiento de sistemas de software. Para lograr esto, se apoya en disciplinas como las ciencias de la computación, las matemáticas, la ingeniería y la gestión de proyectos.

De acuerdo con (Zelkowitz, 1978) la Ingeniería de Software es el estudio de principios y

metodologías que permiten desarrollar y mantener sistemas de software de manera eficiente. (Boehm, 1976) también la define como la aplicación práctica del conocimiento científico para el diseño y construcción de programas de computadora, junto con la documentación necesaria para su operación y mantenimiento.

Esta disciplina se centra en garantizar que el software sea confiable, escalable, seguro y que cumpla con los estándares de calidad necesarios, proporcionando soporte tanto en la fase de desarrollo como en la de operación y mantenimiento.

2.7 METODOLOGIA UWE

La Metodología UWE (UML-based Web Engineering) es una metodología de desarrollo de aplicaciones web utilizada principalmente en el ámbito de la ingeniería web. Su objetivo es sistematizar y personalizar el diseño y desarrollo de sistemas adaptativos, es decir, sistemas que se ajustan a las necesidades específicas de los usuarios y al contexto en el que se utilizan.

UWE se basa en el paradigma orientado a objetos y sigue un enfoque iterativo e incremental, lo que permite la evolución del sistema a lo largo del tiempo. Esta metodología es especialmente útil para el desarrollo de aplicaciones web debido a su capacidad para gestionar modelos complejos y su flexibilidad en la creación de sistemas adaptativos. UWE incluye flujos de trabajo bien definidos y puntos de control que permiten garantizar que cada fase del desarrollo se ajuste a los objetivos y requisitos establecidos.

El lenguaje UWE se deriva del Lenguaje de Modelado Unificado (UML), una herramienta ampliamente utilizada en la ingeniería de software para modelar sistemas complejos. Gracias a esta herencia, UWE ofrece una gran flexibilidad para diseñar modelos web de manera estructurada, permitiendo que sean fácilmente comprensibles para los desarrolladores que ya están familiarizados con UML.

2.7.1 Características

Las principales características que definen la Metodología UWE son:

Uso de notación estándar (UML): UWE utiliza el lenguaje UML como base para todos sus modelos, lo que permite que los diagramas y representaciones gráficas del sistema sean estándar y fáciles de entender.

Definición de métodos claros: UWE proporciona una serie de pasos bien definidos para la creación de diferentes modelos web, lo que facilita la sistematización del proceso de desarrollo.

Especificación de restricciones: UWE permite aplicar restricciones claras en el desarrollo del sistema, asegurando que se cumplan los requisitos y limitaciones del proyecto desde las primeras fases del diseño.

UWE es una metodología específica para el modelado de aplicaciones web que combina la flexibilidad del paradigma orientado a objetos con la precisión y estandarización de UML. Esto garantiza que los desarrolladores puedan crear modelos eficientes y adaptativos para sistemas web complejos, facilitando el mantenimiento y la evolución continua del software.

2.7.2 Fases de la Metodología UWE

La metodología UWE (UML-based Web Engineering) sigue un enfoque iterativo y sistemático para el desarrollo de aplicaciones web. A continuación, se describen las fases más importantes del proceso:

1) Captura, análisis y especificación de requisitos

Esta fase consiste en la identificación, recopilación y definición de los requisitos funcionales (qué debe hacer el sistema) y los requisitos no funcionales (calidad del sistema,

tiempos de respuesta, seguridad, etc.) que deberá cumplir la aplicación web. Durante esta etapa, se detallan las características que el sistema deberá tener para cumplir con las necesidades del usuario y del negocio. El análisis detallado de los requisitos es fundamental para asegurar que la aplicación web sea efectiva y cumpla con las expectativas de los usuarios.

2) Diseño del sistema

En la fase de diseño, se crea la estructura técnica de la aplicación web. Con base en los requisitos previamente definidos, se desarrollan los modelos de la arquitectura del sistema, donde se detallan los componentes y su interacción. Este diseño incluye tanto la estructura de datos, como la interacción del usuario con la aplicación. El objetivo es definir cómo se cumplirán los requisitos establecidos y preparar un modelo que sirva como referencia para la implementación.

3) Codificación del software

Durante la codificación, el diseño se transforma en código ejecutable. Aquí se utilizan lenguajes de programación como PHP, JavaScript o cualquier otro que haya sido elegido en la fase de diseño. Los desarrolladores escriben el código fuente de la aplicación web, implementando cada funcionalidad especificada en el diseño. Esta fase es crucial porque la calidad del código influirá directamente en el rendimiento, seguridad y mantenimiento del sistema.

4) Pruebas

Las pruebas son esenciales para garantizar que el sistema funcione de acuerdo con los requisitos. En esta fase se verifican tanto los componentes individuales del sistema como la integración entre ellos, asegurando que no existan errores críticos o defectos que afecten su operación. Se realizan pruebas unitarias (a nivel de componentes), pruebas de integración

(verificación del funcionamiento conjunto) y pruebas de aceptación (para comprobar que el sistema cumple con los requisitos del usuario).

5) Implementación y lanzamiento

La implementación implica trasladar el software a un entorno de producción, configurarlo y asegurarse de que esté listo para ser utilizado por los usuarios. Una vez que la aplicación ha sido probada y aprobada, se despliega en el servidor web. Para garantizar la compatibilidad y escalabilidad futuras, se recomienda el uso de estándares web como HTML, HTML y CSS. Tras la implementación, el sitio web se lanza públicamente y comienza a ser utilizado por los usuarios finales.

6) Mantenimiento

El mantenimiento es una fase continua que asegura que el sistema siga funcionando correctamente a lo largo del tiempo. Implica la actualización de contenido, la corrección de errores no detectados durante las pruebas, y la mejora de la aplicación según los comentarios y necesidades emergentes de los usuarios. El mantenimiento también incluye la optimización del sistema para garantizar que siga siendo eficiente y seguro en función de las demandas cambiantes del entorno web.

❖ Mantenimiento y seguimiento

Una vez puesta la Pagina Web a Disposición de los usuarios hay que ir cambiando datos y mantener este sitio actualizado, ya que esta página no puede permanecer estática. Los problemas de uso no detectados durante el proceso de desarrollo pueden descubrirse a través de varios métodos, principalmente a través de los mensajes, opiniones de los usuarios, el comportamiento y uso del sitio. Según, (Pressman, 2010)

❖ UWE Basado en UML

La Ingeniería Web basada en UML (UWE) fue propuesta por Nora Koch en el año 2000. Esta metodología utiliza un enfoque orientado a objetos y está enfocada en el usuario, lo que permite crear aplicaciones web centradas en sus necesidades. UWE utiliza una extensión del perfil UML para adaptar este lenguaje a las aplicaciones web, lo que permite modelar de manera precisa los componentes del sistema.

Los modelos de UWE incluyen:

- ▶ **Modelo lógico-conceptual:** Define cómo están estructurados los datos en el sistema.
- ▶ **Modelo de navegación:** Muestra cómo los usuarios se moverán a través de la aplicación.
- ▶ **Modelo de presentación:** Diseña la interfaz de usuario de la aplicación.

Estas vistas están representadas gráficamente mediante diagramas UML, lo que facilita el entendimiento y la organización del sistema.

Principales Actividades del Modelado en UWE

Las actividades clave en UWE incluyen:

- ▶ **Análisis de requerimientos:** Identificar las necesidades del sistema.
- ▶ **Modelo conceptual:** Definir cómo los contenidos y datos están relacionados dentro del sistema.
- ▶ **Modelo de navegación:** Establecer cómo las diferentes páginas y secciones del

sitio estarán enlazadas.

- ▶ **Modelo de presentación:** Crear el diseño de la interfaz de usuario.

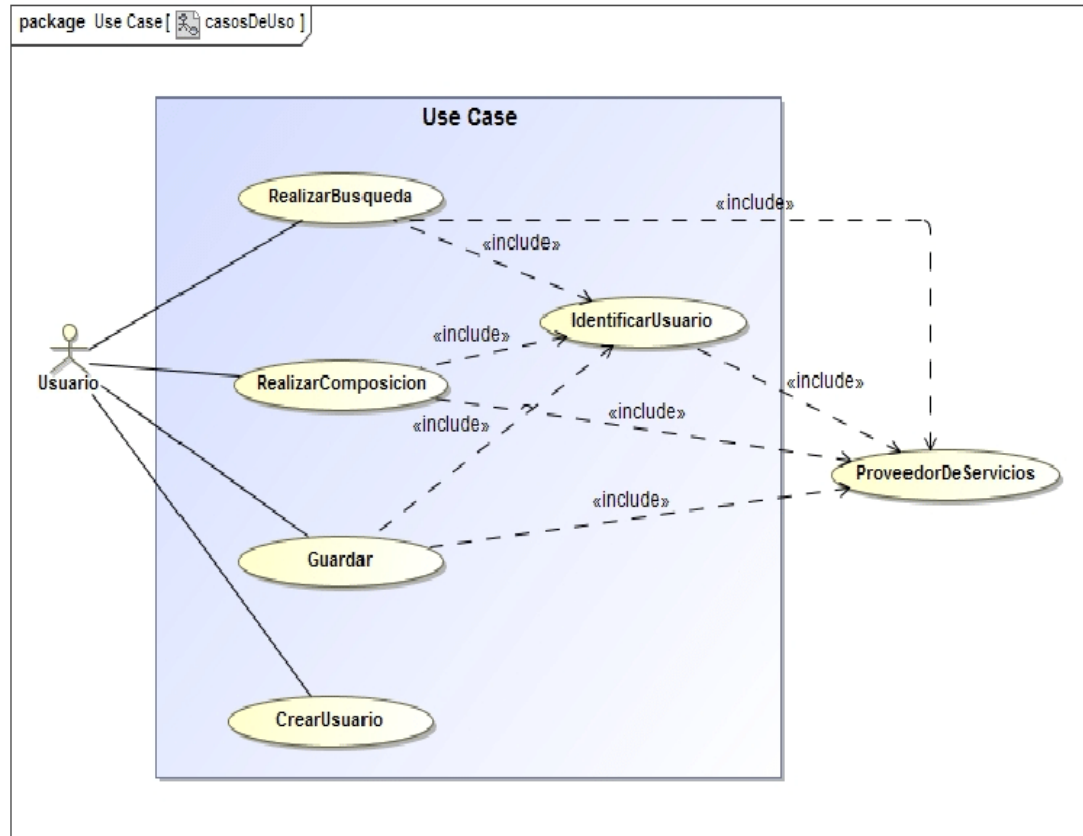
Cada uno de estos modelos proporciona una vista especializada del sistema y asegura que todas las partes se integren de manera coherente.

❖ **Modelo de casos de Uso**

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario.

Los diagramas de caso de uso modelan la funcionalidad del sistema usando actores y casos de uso. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios.

Figura 2.

Diagrama de Casos de Uso

Nota. Sistema Caso de uso Usuario, (Presman 2010)

❖ Modelo Conceptual

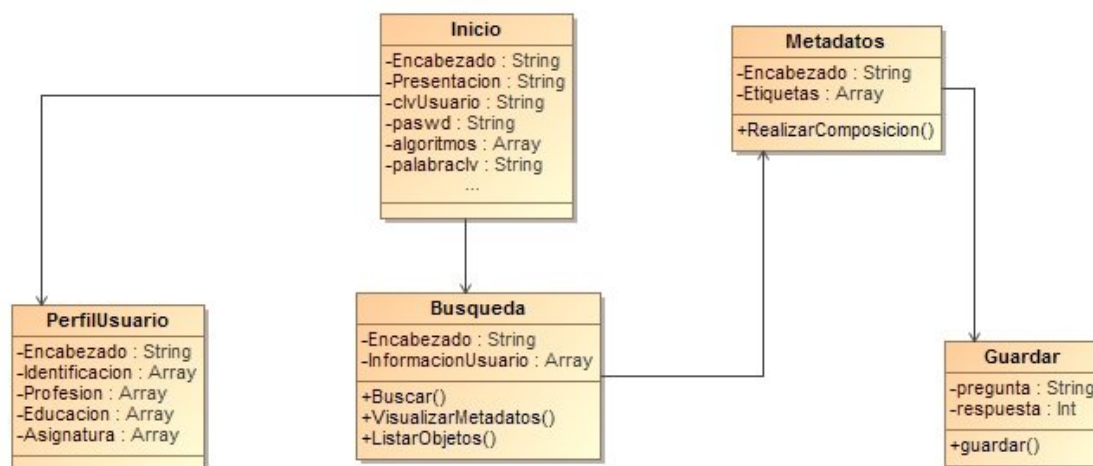
El modelo conceptual en la metodología UWE es una representación abstracta que describe la estructura de los datos en el dominio de la aplicación web. Este modelo especifica cómo los diferentes contenidos del sistema están relacionados entre sí, y cómo los datos serán organizados y gestionados en el sitio web. Su propósito es asegurar que la estructura de los datos cumpla con los requisitos establecidos en la fase de análisis de requerimientos.

Este modelo es esencial porque traduce los requisitos funcionales y no funcionales en un esquema comprensible y organizado, permitiendo que los desarrolladores tengan una visión clara de cómo deben estructurarse los datos y cómo interactúan las entidades dentro del sistema.

Su objetivo es construir un modelo conceptual del dominio de la aplicación considerando los requisitos reflejados en los casos de uso. Da como resultado un diagrama de clases de dominio. (Pressman, 2010)

Figura 3.

Modelo de Contenido (UWE): MHC-PMS



Nota. Adaptado de Pressman, modelo de Contenido UWE de (Pressman, 2010)

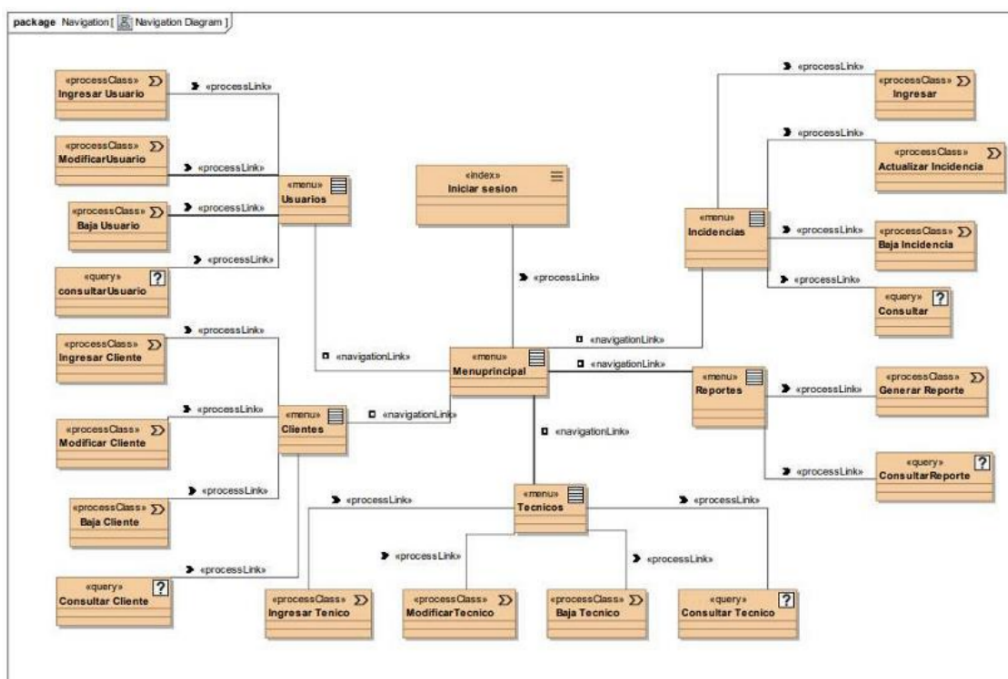
❖ Modelo de Navegación

Tiene como objetivo la representación de nodos y enlaces de la estructura de hipertexto, y el diseño de las rutas de navegación mediante diagramas de clases UML.

En la figura 5 se presenta el modelo de navegación del MHC-PMS, donde los distintos nodos, enlaces de la estructura de hipertexto, el diseño de las rutas de navegación y la relación que existe entre los distintos nodos del sistema, se encuentran caracterizados por estereotipos definidos por UWE. Dichos nodos se encuentran agrupados en paquetes para poder clasificarlos según al tipo de gestión del sistema al que pertenece, el cual puede ser de paciente o de usuarios.

Figura 4.

Modelo de Navegación (UWE): MHC-PMS



Nota. Diagrama de navegación basado en el diseño de usuario (Pressman, 2010)

❖ Modelo estructura de navegación

Este refinamiento se enfoca en mejorar el modelo de espacio de navegación de una aplicación web, añadiendo estructuras de acceso que optimizan la experiencia del usuario al interactuar con el sitio. Estas estructuras incluyen elementos como índices, guías de ruta, consultas y menús, que facilitan la navegación y permiten al usuario acceder de manera más eficiente a la información.

<<index>> Proporciona un acceso directo a todas las instancias de un destino específico, mostrando una lista completa de elementos desde los cuales el usuario puede seleccionar para navegar dentro de la aplicación web.

<<guided tour>> Ofrece un recorrido secuencial a través de las distintas instancias de un nodo destino, guiando al usuario de manera estructurada a través del contenido del sistema.

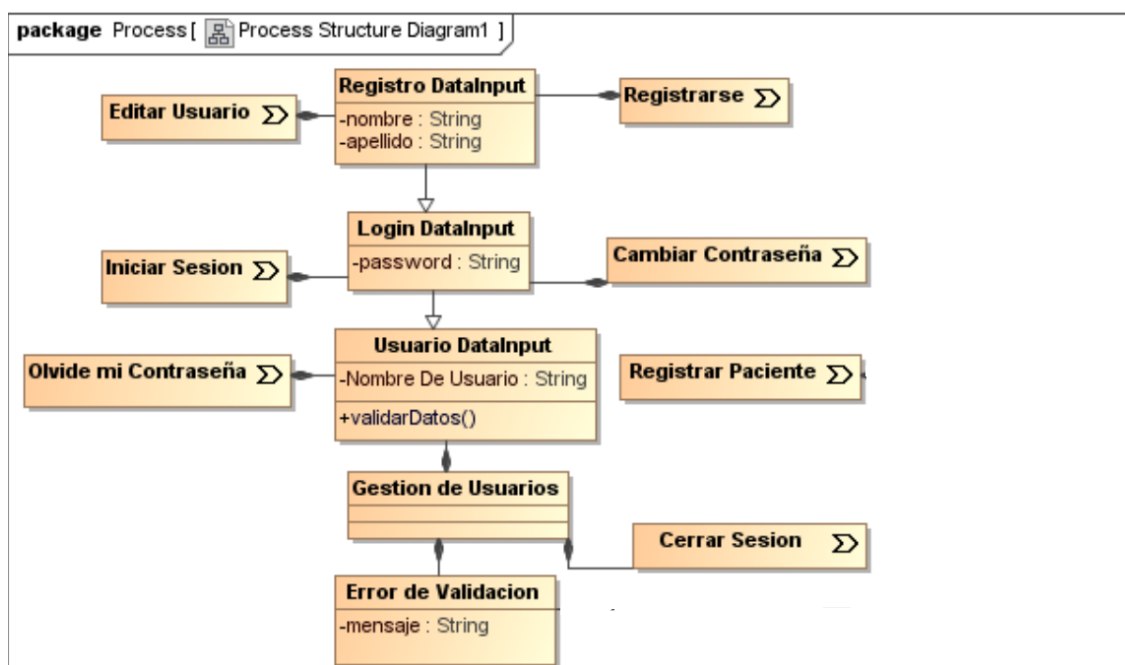
<<query>> Representa la funcionalidad de búsqueda, permitiendo al usuario buscar instancias específicas dentro de un nodo destino, lo que facilita encontrar información relevante de manera rápida y precisa.

<<menu>> Organiza y estructura los enlaces de salida desde un nodo, proporcionando una forma ordenada y clara de acceder a distintas secciones o funcionalidades de la aplicación.

Estos componentes mejoran la usabilidad del sistema al ofrecer diferentes formas de acceder y navegar por la información, asegurando una experiencia de usuario más fluida y eficiente. (Pressman, 2010)

Figura 5.

Modelo de Estructura de navegación (UWE)



Nota. Diagrama estructural de procesos del sistema, basado en lineamientos de diseño de usuario de Pressman (2010)

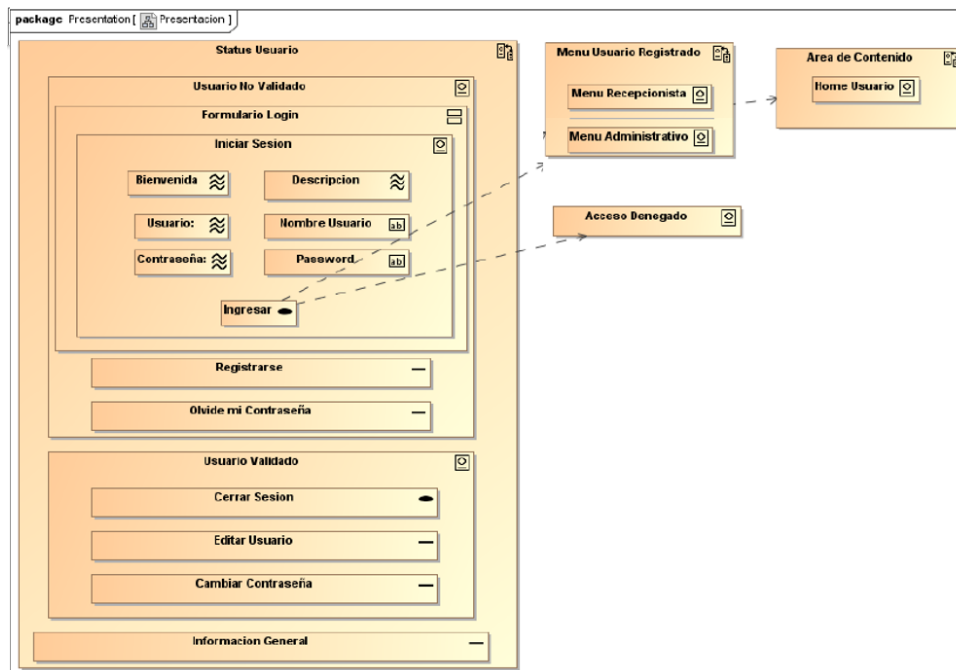
❖ Modelo de Presentación

Representa el layout subyacente a los modelos de navegación y procesos mediante una presentación abstracta, ya que una presentación concreta requeriría la especificación de propiedades físicas adicionales. Utiliza un diagrama de clases de UML para modelar la estructura.

En las figuras 6 y 7 se presentan los modelos de presentación para el caso de estudio tratado, donde se pretende mostrar las clases de navegación y de procesos a las que pertenece a cada página web. Se puede observar, también, cómo se relacionan los elementos de distintas páginas, como llegar hasta ellas y los mensajes de error que pueden darse durante la interacción.

Figura 6.

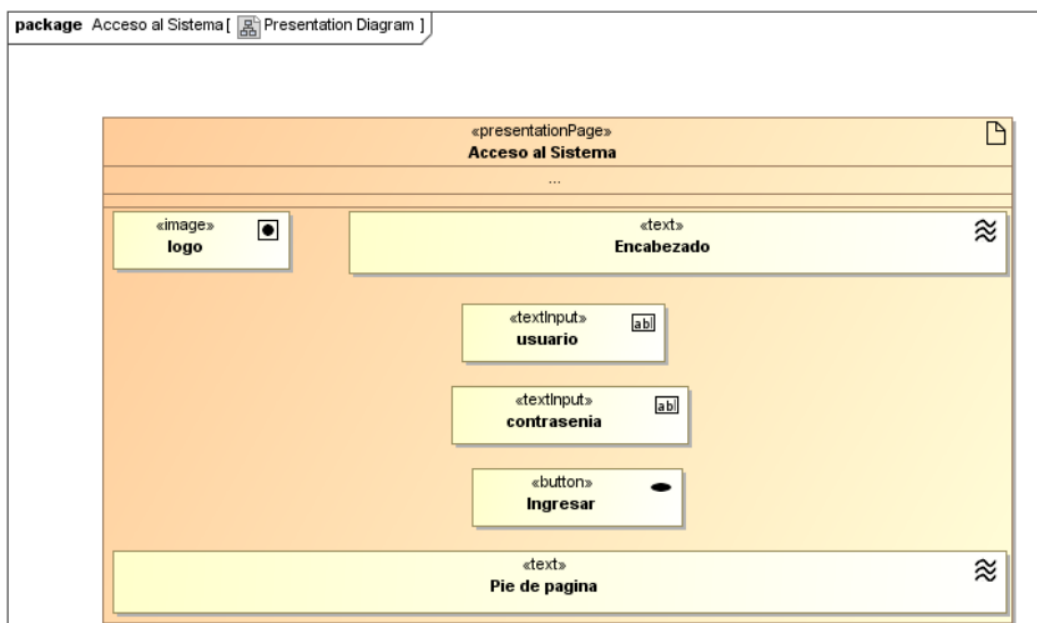
Modelo de Presentación (UWE): MHC-PMS



Nota. Modelo de presentación, basado en lineamientos de diseño interfaz de usuario de Pressman (2010)

Figura 7.

Modelo de Presentación Acceso al sistema (UWE)



Nota. Modelo del acceso al sistema, basado en lineamientos de diseño de usuario de Pressman (2010)

2.8 ARQUITECTURA DEL SOFTWARE

Hoy en día, es común escuchar sobre la Arquitectura de Software y el rol fundamental que desempeñan los arquitectos en la creación de soluciones tecnológicas. Sin embargo, ¿sabemos realmente qué es la Arquitectura de Software? A continuación, se ofrece un panorama claro y detallado de este concepto clave en el desarrollo de software.

La Arquitectura de Software es el proceso mediante el cual se define una solución que responde a los requisitos técnicos y operacionales de un sistema. En otras palabras, la arquitectura establece qué componentes conforman el software, cómo interactúan entre sí y de qué manera estas interacciones logran cumplir con la funcionalidad requerida, respetando criterios como la seguridad, la disponibilidad, la eficiencia y la usabilidad.

El diseño de la arquitectura aborda cuestiones cruciales que pueden determinar el éxito o fracaso de un proyecto de software. Para ello, es necesario hacerse preguntas clave:

- ¿En qué entorno se implementará el software?
- ¿Cómo se llevará a producción?
- ¿De qué manera interactuarán los usuarios con el sistema?
- ¿Existen requisitos adicionales como seguridad, rendimiento o disponibilidad?
- ¿Qué cambios en la arquitectura podrían afectar al sistema en el futuro?

Al diseñar la arquitectura, es esencial tener en cuenta los intereses de todas las partes involucradas: los usuarios, el software en sí y los objetivos del negocio. Cada uno de estos agentes establece requisitos y restricciones que pueden entrar en conflicto. Por ejemplo, los usuarios podrían desear una respuesta rápida y fluida, mientras que la empresa puede priorizar la reducción de costos.

El papel del arquitecto es precisamente equilibrar estas necesidades, identificando los escenarios y requisitos de calidad críticos para cada agente. El arquitecto debe prever tanto los objetivos clave que el sistema debe cumplir como las acciones o eventos que deben evitarse para garantizar el éxito del software.

El objetivo final de la arquitectura es identificar los requisitos que impactan la estructura del software y minimizar los riesgos asociados a su construcción. Una buena arquitectura debe estar preparada para adaptarse a los cambios futuros, tanto en software y hardware como en las nuevas funcionalidades que los clientes puedan requerir. Además, el arquitecto debe evaluar las implicaciones de sus decisiones de diseño y encontrar compromisos entre las diversas necesidades de calidad, asegurando que el software, los usuarios y los objetivos del negocio queden satisfechos.

Principales Características de la Arquitectura de Software

La arquitectura de software debe cumplir con ciertas capacidades fundamentales, tales como:

- ▶ Mostrar la estructura general del sistema, ocultando los detalles innecesarios.
- ▶ Definir los casos de uso clave que guiarán el diseño del sistema.
- ▶ Satisfacer los intereses de los usuarios, del software y del negocio en la mayor medida posible.
- ▶ Manejar los requisitos funcionales y no funcionales, asegurando la calidad y rendimiento del sistema.
- ▶ Determinar el tipo de software que se desarrollará, así como los estilos arquitecturales más adecuados.
- ▶ Gestionar cuestiones transversales, como la seguridad, el rendimiento y la escalabilidad.

La Arquitectura de Software no solo define la estructura del sistema, sino que también sienta las bases para su éxito futuro al equilibrar las necesidades de todos los agentes involucrados. Su capacidad para soportar cambios, optimizar recursos y minimizar riesgos es esencial en el desarrollo de software moderno. (Sifuentes, 2018)

2.8.1 Patrón Modelo Vista Controlador

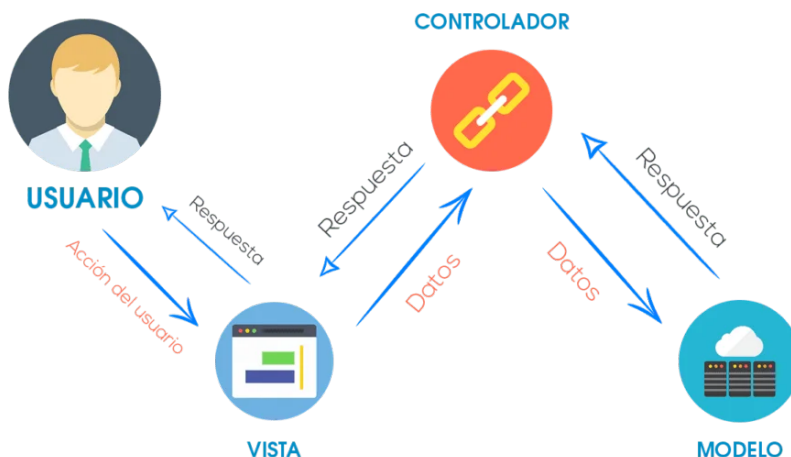
El MVC o Modelo Vista Controlador es un patrón arquitectónico ampliamente utilizado en el desarrollo de software que busca separar las responsabilidades del sistema en tres componentes principales: Modelo, Vista y Controlador. Esta división facilita la gestión y el mantenimiento del código al separar la lógica del negocio, la interfaz de usuario y las interacciones del sistema. El objetivo del MVC es organizar mejor el flujo de datos y la

interacción del usuario con el sistema, proporcionando una estructura clara y modular. (Fontan, 2012)

- **Modelo** Es la capa encargada de gestionar los datos y la lógica del negocio. Es la parte del sistema que interactúa directamente con la base de datos u otras fuentes de información, y se encarga de realizar operaciones como la creación, lectura, actualización y eliminación de los datos (CRUD). Además, el modelo es responsable de actualizar su estado según los cambios realizados a través del controlador. En otras palabras, el modelo es el núcleo del sistema, donde se definen las reglas de negocio y se almacena la información.
- **Vista** Es la capa que gestiona la presentación de la información al usuario. Esta capa contiene el código que permite generar las interfaces visuales de la aplicación, convirtiendo los datos del modelo en una forma comprensible y navegable para el usuario, generalmente a través de HTML, CSS o tecnologías similares. La vista es esencialmente la parte visual de la aplicación y es responsable de mostrar los estados de la aplicación al usuario final, pero no realiza ninguna lógica de negocio ni gestiona los datos directamente.
- **Controlador** Es el que actúa como intermediario entre el modelo y la vista. Se encarga de gestionar las solicitudes del usuario y traducirlas en acciones que el modelo debe ejecutar, como actualizar datos, realizar búsquedas o procesar una transacción. Una vez que el controlador interactúa con el modelo, actualiza la vista con los nuevos datos o estados. En resumen, el controlador es el responsable de coordinar el flujo de información y controlar las interacciones entre el usuario, la vista y el modelo.

Figura 8.

Flujo Modelo Vista Controlador



Nota. Flujo Modelo Vista Controlador, basado en Precognis (2022)

Aunque el auge de este término durante los últimos tiempos pueda indicar lo contrario, MVC no es un concepto nuevo, ya que el patrón fue descrito en el año 1979 por Trygve Reenskaug, hoy en día profesor emérito de informática de la Universidad de Oslo, mientras trabajaba con el equipo de Smalltalk en los laboratorios Xerox PARC.

Por tanto, teniendo en cuenta su antigüedad, es obvio que ni siquiera fue ideado expresamente para sistemas web, aunque ahora se use mucho en ellas. Existen implementaciones para todo tipo de sistemas (escritorio, clientes y servidores web, servicios web, Single Page Applications o SPA, etc.) y lenguajes (Smalltalk, Java, Ruby, C++, Python, PHP, JavaScript, NodeJS, etc.).

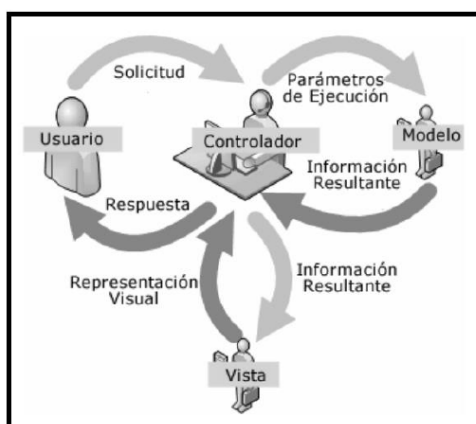
La arquitectura MVC propone, independientemente de las tecnologías o entornos en los que se base el sistema a desarrollar, la separación de los componentes de una aplicación en tres grupos (o capas) principales: el modelo, la vista, y el controlador, y describe cómo se relacionarán entre ellos para mantener una estructura organizada, limpia y con un acoplamiento mínimo entre las distintas capas.

2.8.2 Ciclo de Vida

El ciclo de vida del patrón MVC se organiza en tres capas fundamentales: el Modelo, la Vista y el Controlador, que interactúan entre sí para gestionar las solicitudes y operaciones del sistema. El cliente o usuario es quien inicia este ciclo con una acción o solicitud. A continuación, se explica el flujo del ciclo de vida en términos más detallados. (Alvarez, 2014)

Figura 9.

Ciclo de vida de un MVC



Nota. Ciclo de vida de un Modelo, Vista, Controlador, Álvarez, 2014

❖ Inicio de la Solicitud

El ciclo comienza cuando el usuario realiza una acción o envía una solicitud. Esta solicitud, que puede ser una acción como enviar un formulario, hacer clic en un botón o solicitar información, es dirigida al Controlador. La tarea del controlador en este primer paso es recibir la solicitud y determinar qué acción específica debe tomar el sistema en respuesta a la entrada del usuario.

❖ Procesamiento en el Controlador

Una vez recibida la solicitud, el Controlador decide qué acción debe ejecutarse. Esta decisión incluye identificar qué parte del Modelo es relevante para la solicitud del usuario. El

controlador, en este sentido, actúa como un coordinador que determina qué datos se necesitan y qué operaciones deben realizarse para responder a la solicitud.

❖ Operaciones del Modelo

Después de delegar la tarea al Modelo, este comienza a procesar la información solicitada. El Modelo es responsable de interactuar con la base de datos o las fuentes de datos, realizar las operaciones necesarias como consultar, actualizar o eliminar información, y finalmente devolver los resultados al Controlador. El Modelo es la capa que gestiona la lógica de negocio y las reglas del sistema.

❖ Retorno al Controlador

Una vez que el Modelo ha completado sus operaciones y obtenido los resultados, pasa esa información de vuelta al Controlador. El Controlador recibe los datos procesados y decide cómo presentarlos al usuario a través de la Vista.

❖ Generación de la Vista

El Controlador entonces delega a la Vista la tarea de convertir los datos en una representación visual comprensible para el usuario. La Vista toma los datos proporcionados por el Controlador y los organiza en un formato que el usuario puede entender, generalmente en forma de una página web (HTML, CSS, etc.), una interfaz gráfica o cualquier otra salida visual.

❖ Respuesta al Usuario

Finalmente, la Vista presenta los resultados visuales al Controlador, quien se encarga de enviarlos al usuario. El usuario recibe la información solicitada en un formato visual entendible. Este ciclo puede repetirse indefinidamente cada vez que el usuario realiza una nueva acción o solicitud, permitiendo una interacción fluida entre el sistema y el usuario.

Este ciclo se repite cada vez que el usuario interactúa con el sistema, permitiendo una separación clara entre los datos, la lógica de negocio y la interfaz gráfica, lo que facilita el mantenimiento, escalabilidad y evolución del sistema.

❖ **Ventajas del patrón MVC**

El uso del patrón MVC ofrece múltiples ventajas sobre otras maneras de desarrollar aplicaciones con interfaz de usuario, y en especial para la Web. Sin entrar en detalles, a continuación, veremos algunas de ellas:

- La clara separación de responsabilidades impuesta por el uso del patrón MVC hace que los componentes de nuestras aplicaciones tengan sus misiones bien definidas. Por lo tanto, nuestros sistemas serán más limpios, simples, más fácilmente mantenibles y, por tanto, más robustos.
- Mayor velocidad de desarrollo en equipo, que es consecuencia de lo anterior, ya que, al estar separado en tres partes tan diferenciadas, diferentes programadores pueden ocuparse de cada parte en paralelo. Esto la hace ideal para el desarrollo de aplicaciones grandes.
- Múltiples vistas a partir del mismo modelo, pudiendo reaprovechar mucho mejor los desarrollos y asegurando consistencia entre ellas.
- Facilidad para realización de pruebas unitarias.

❖ **Desventajas del patrón MVC**

Pero, por supuesto, no todo pueden ser siempre ventajas, así que su uso presenta también algunas desventajas, entre las que cabe destacar:

- Hay que ceñirse a las convenciones y al patrón. El uso de las convenciones

impuestas por el framework y la estructura propuesta por el patrón arquitectural MVC nos obliga a ceñirnos a las mismas, lo que puede resultar a veces algo tedioso si lo comparamos con la forma habitual de trabajar con otros frameworks que dan más libertad al desarrollador.

- La división impuesta por el patrón MVC obliga a mantener un mayor número de archivos, incluso para tareas simples.
- Curva de aprendizaje. Dependiendo del punto de partida, el salto a MVC puede resultar un cambio radical y su adopción requerirá cierto esfuerzo. Además, utilizarlo implica conocer bien las tecnologías subyacentes con las que se implemente: la plataforma de programación utilizada, además de la tecnología utilizada para la interfaz de usuario (HTML, CSS, JavaScript en el caso de la Web).

En cualquier caso, el uso del patrón MVC y sus variantes está claro que ha triunfado en todo tipo de desarrollos Web, móvil, de escritorio etc. y en todo tipo de plataformas rara es la plataforma actual que no lo implementa para uno o varios tipos de desarrollos. (Precognis,2022)

2.9 HERRAMIENTAS DE DESARROLLO

❖ Servidor Apache

Los servidores web son esenciales en el funcionamiento de Internet tal como lo conocemos hoy en día. A través de ellos, se entregan todas las páginas web y el contenido multimedia que consumimos a diario. Además, son parte integral de la infraestructura que soporta aplicaciones móviles y servicios en la nube.

Cada vez que un usuario introduce una dirección web en su navegador, este genera una petición HTTP hacia un servidor remoto. El servidor, a través de un servicio llamado servidor

web, procesa esta solicitud y responde entregando la página web solicitada. Este proceso complejo permite que los usuarios interactúen con sitios web y accedan al contenido que desean visualizar. (Leon, 2019)

Entre los servidores web más reconocidos se encuentra Apache, uno de los servidores más longevos y ampliamente utilizados en la historia de Internet. A pesar de que en los últimos años han surgido alternativas más modernas y eficientes, como NGINX, Apache sigue siendo un servidor de referencia en la industria.

Apache comenzó como un conjunto de parches aplicados al servidor NCSA HTTPd, pero eventualmente se reescribió por completo, eliminando el código original. A lo largo del tiempo, Apache ha evolucionado para convertirse en un servidor web robusto y confiable, que sigue siendo muy popular entre desarrolladores y administradores de sistemas.

En esencia, Apache es un servidor HTTP que permite responder a las solicitudes realizadas por los navegadores de los usuarios, entregando contenido web (como páginas HTML, imágenes, videos, etc.). Su capacidad para gestionar múltiples tipos de contenido y solicitudes lo convierte en una opción versátil para servidores web.

Aunque en la actualidad han surgido alternativas que optimizan ciertos aspectos del rendimiento, como la capacidad para manejar grandes volúmenes de tráfico simultáneo, Apache sigue siendo una opción sólida gracias a su flexibilidad, comunidad activa y arquitectura modular. Esto le permite ser personalizable para una amplia variedad de necesidades y entornos, lo que lo convierte en una herramienta adaptable tanto para pequeñas aplicaciones web como para grandes infraestructuras.

Apache sigue siendo un servidor popular debido a su facilidad de uso, su amplio soporte en diferentes plataformas y su capacidad de adaptación a nuevas tecnologías. Permite a los

desarrolladores agregar o quitar módulos según las necesidades de su aplicación, lo que lo hace muy flexible. Además, su código abierto ha permitido que una comunidad global de desarrolladores continúe mejorándolo.

A pesar de que alternativas como NGINX han ganado popularidad en proyectos que requieren mayor escalabilidad y manejo de alto tráfico, Apache sigue siendo una herramienta confiable y preferida en muchos contextos, especialmente en configuraciones donde se valora la flexibilidad y personalización. (Leon, J. 2019).

2.9.1 Gestor de base de datos

❖ Base de datos

Una base de datos es un conjunto organizado de información que se almacena de manera estructurada, permitiendo que su contenido sea accedido, gestionado y actualizado de forma eficiente. El término "base" proviene del latín basis, que hace referencia al sostén o fundamento, mientras que un "dato" se refiere a una pieza de información concreta, un testimonio o una prueba que puede ser utilizada para tomar decisiones.

En el contexto de la informática, una base de datos se define como una colección de datos organizados de tal manera que permiten su procesamiento y análisis de manera rápida y precisa. A través de la estructura organizada de las bases de datos, los usuarios pueden localizar y manipular fácilmente la información, logrando mayor eficiencia en comparación con sistemas donde los datos estarían desorganizados o dispersos.

Las bases de datos suelen organizarse de acuerdo con diferentes criterios, como el tipo de información que contienen o la relación entre los datos. Esta estructura permite que los usuarios encuentren la información de manera rápida y sencilla, ya que el sistema está diseñado para almacenar y recuperar datos en base a consultas definidas. Por ejemplo, en una

base de datos relacional, los datos están dispuestos en tablas, lo que facilita las operaciones de búsqueda, actualización y análisis de la información.

❖ PostgreSQL

PostgreSQL es uno de los sistemas de gestión de bases de datos relacionales más populares y avanzados, desarrollado y mantenido por la comunidad de código abierto. A diferencia de almacenar todos los datos en un solo lugar, PostgreSQL organiza la información en tablas separadas y relacionadas entre sí, lo que incrementa tanto la flexibilidad como la velocidad de las consultas.

El término relacional se refiere a la capacidad de PostgreSQL para crear relaciones entre tablas, facilitando la combinación de datos de diferentes tablas cuando es necesario. Esto lo convierte en una herramienta muy eficiente y rápida para gestionar grandes volúmenes de datos y realizar consultas complejas mediante SQL (Lenguaje de Consulta Estructurado), el estándar de la industria para gestionar bases de datos relacionales.

PostgreSQL se ha consolidado como una de las mejores opciones de bases de datos para proyectos que requieren alta fiabilidad, flexibilidad y rendimiento. Ya sea para una aplicación web de gran envergadura, un sistema financiero o una solución empresarial, PostgreSQL proporciona todas las herramientas necesarias para gestionar datos de forma eficiente y segura, además de contar con una comunidad activa que continuamente mejora sus capacidades y funcionalidades. (PostgreSQL Global Development Group)

❖ PostgreSQL es Open Source

Al igual que otros sistemas populares, PostgreSQL es software de código abierto, lo que significa que cualquiera puede usarlo, modificarlo y distribuirlo sin costo alguno. Está disponible bajo la Licencia Pública General PostgreSQL, que permite a los usuarios descargar,

estudiar y modificar el código fuente de acuerdo a sus necesidades. Esto lo convierte en una opción extremadamente flexible para desarrolladores y empresas que necesitan personalizar su entorno de bases de datos.

Además, aunque PostgreSQL es gratuito y de código abierto, también existe la posibilidad de obtener soporte comercial de varias empresas especializadas en el entorno PostgreSQL. Esto ofrece a las organizaciones la opción de obtener soporte profesional, si así lo desean, para integraciones o proyectos específicos de gran envergadura. (PostgreSQL Global Development Group.)

2.9.2 Lenguaje de programación

❖ PHP

Es un lenguaje de programación de uso general que permite la generación dinámica de contenidos en un servidor web. Según Gil (2001, p. 2), “es un lenguaje de script que permite la generación dinámica de contenidos en un servidor. Entre sus principales características se destacan su potencia, alto rendimiento y facilidad de aprendizaje.”

PHP fue diseñado originalmente para el desarrollo de sitios web con contenido dinámico y es ampliamente reconocido por su capacidad para ser incrustado directamente en el código HTML, lo que simplifica la creación de páginas dinámicas sin la necesidad de llamar a archivos externos para el procesamiento de datos. El código PHP es interpretado por un servidor web que tiene un módulo específico de procesamiento de PHP, y este genera la página web resultante que se mostrará al usuario.

Con el paso del tiempo, PHP ha evolucionado y ahora incluye una interfaz de línea de comandos, lo que lo hace útil no solo para aplicaciones web, sino también para aplicaciones gráficas independientes. Una de sus grandes ventajas es que puede ser usado en la mayoría de

los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno, lo que lo convierte en una opción versátil para los desarrolladores.

Este lenguaje es conocido por su flexibilidad, potencia y alto rendimiento, características que han atraído a grandes plataformas con una alta demanda de tráfico, como Facebook, para implementarlo como su tecnología de servidor. PHP fue creado originalmente por Rasmus Lerdorf en 1995, y desde entonces, ha sido desarrollado y mejorado por la comunidad de desarrolladores del grupo PHP, agregando continuamente nuevas funcionalidades.

PHP es parte del software libre, publicado bajo la Licencia PHP, que aunque es muy similar a otras licencias de software libre, es incompatible con la Licencia Pública General de GNU debido a ciertas restricciones en el uso del término "PHP". (Gil, 2001).

❖ JavaScript

Según Flanagan (2002, p. 7), JavaScript es un lenguaje de programación interpretado que constituye un dialecto del estándar ECMAScript. Se caracteriza por ser orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Fue desarrollado originalmente por Brendan Eich en Netscape, bajo el nombre de Mocha, luego renombrado a LiveScript, hasta que finalmente se estableció como JavaScript.

JavaScript se utiliza predominantemente en su forma client-side (del lado del cliente), como parte integral de los navegadores web, permitiendo la creación de interfaces de usuario mejoradas y la generación de páginas web dinámicas. Sin embargo, su uso no se limita solo a la web, ya que también se emplea en otras aplicaciones, como la manipulación de documentos PDF y el desarrollo de aplicaciones de escritorio (como widgets).

Aunque JavaScript fue diseñado con una sintaxis similar a C y adopta ciertos nombres y convenciones de Java, ambos lenguajes son fundamentalmente diferentes en su semántica y

propósito. Es importante destacar que Java y JavaScript no están relacionados, a pesar de las similitudes en sus nombres.

En todos los navegadores modernos, JavaScript es interpretado directamente dentro del código de las páginas web. Para permitir la interacción con estas páginas, JavaScript incluye una implementación del Document Object Model (DOM), que facilita el acceso y manipulación de la estructura de un documento HTML o XML.

❖ jQuery

jQuery es una biblioteca de JavaScript diseñada para simplificar el proceso de programación en este lenguaje. Se utiliza principalmente para añadir elementos interactivos a una página web sin necesidad de escribir grandes cantidades de código, lo que facilita el desarrollo y mejora la experiencia del usuario.

Una de las principales ventajas de jQuery es que permite a los desarrolladores manipular fácilmente el DOM (Document Object Model), manejar eventos, crear animaciones y realizar peticiones AJAX, todo ello con unas pocas líneas de código. Esto reduce considerablemente el tiempo de desarrollo en comparación con escribir código JavaScript puro, ya que jQuery abstrae gran parte de la complejidad, proporcionando una sintaxis más simple y concisa.

jQuery es de código abierto, lo que significa que cualquiera puede contribuir a su desarrollo, ya que el acceso al código fuente es completamente libre. Al estar basada en JavaScript, jQuery es compatible con todos los navegadores modernos, lo que asegura que las funcionalidades creadas con esta biblioteca funcionen correctamente en diferentes plataformas.

Entre sus características más destacadas se incluyen:

- Selección de elementos DOM jQuery simplifica la búsqueda y manipulación de elementos dentro de una página web mediante selectores que se basan en la sintaxis de CSS.
- Manejo de eventos proporciona una manera sencilla de asignar eventos a los elementos de una página, como clics, movimientos de ratón o pulsaciones de teclas.
- Animaciones y efectos permite agregar animaciones y efectos visuales, como mostrar, ocultar o deslizar elementos, sin necesidad de recurrir a complejas líneas de código.
- AJAX simplificado facilita la integración de peticiones AJAX para interactuar de forma asíncrona con servidores web, permitiendo la actualización dinámica de contenido sin recargar la página.

Debido a su versatilidad y simplicidad, jQuery se ha convertido en una de las bibliotecas más populares en el desarrollo web, especialmente en la creación de interfaces dinámicas y amigables para el usuario. Aunque en la actualidad existen nuevas alternativas como React o Vue.js, jQuery sigue siendo una herramienta ampliamente utilizada, sobre todo en proyectos que requieren soluciones rápidas y eficientes para manipulación del DOM y manejo de eventos.

Según Porto J.P. (2018), jQuery continúa siendo una opción atractiva para desarrolladores que buscan reducir la complejidad del código JavaScript y optimizar el rendimiento de sus aplicaciones web.

2.9.3 Framework

❖ Laravel

Laravel es un framework de PHP diseñado para el desarrollo de aplicaciones web que sigue el patrón MVC (Modelo-Vista-Controlador). Laravel se ha destacado por ofrecer una sintaxis elegante y expresiva que facilita el trabajo de los desarrolladores, permitiéndoles centrarse en los aspectos creativos del desarrollo sin comprometer la funcionalidad ni la escalabilidad del sistema. Laravel proporciona un entorno robusto para la creación de aplicaciones web modernas y escalables.

Ventajas de usar Laravel como framework:

- Ecosistema completo Laravel cuenta con un ecosistema robusto que incluye herramientas integradas como Eloquent ORM (para la manipulación de bases de datos), Artisan (para automatización de tareas), y Blade (su motor de plantillas).
- Autenticación y autorización sencilla Laravel facilita la implementación de sistemas de autenticación, manejo de permisos y protección de rutas, permitiendo a los desarrolladores configurar estos aspectos en poco tiempo.
- Enrutamiento elegante El sistema de enrutamiento en Laravel es intuitivo y fácil de implementar, lo que facilita la creación de rutas RESTful para cualquier aplicación.
- ORM Eloquent Este sistema de mapeo objeto-relacional permite manejar bases de datos con un enfoque orientado a objetos, proporcionando una interacción fluida y sencilla con las tablas de la base de datos.
- Sistema de migraciones y seeds Laravel ofrece migraciones de base de datos que permiten versionar los cambios en el esquema de forma segura y eficiente. Además, facilita la inserción de datos mediante seeds.

- Extensa documentación y comunidad activa: Laravel cuenta con una documentación completa y clara, así como una gran comunidad que respalda el desarrollo y mejora del framework, con foros, tutoriales y paquetes de terceros disponibles.
- Blade Laravel tiene su propio motor de plantillas llamado Blade, que permite el uso de estructuras de control y reutilización de componentes, además de ser rápido y ligero.
- Seguridad avanzada Laravel proporciona varios mecanismos de seguridad como hashing de contraseñas, protección contra CSRF, y prevención de ataques SQL Injection, lo que lo convierte en una opción segura para el desarrollo de aplicaciones. Stauffer, M., & Taylor, O. (2019).

❖ **Bootstrap**

Bootstrap es un framework de código abierto que facilita el desarrollo de sitios web y aplicaciones web responsivas y con un diseño atractivo. Fue creado por desarrolladores de Twitter y lanzado en 2011, con el propósito de ayudar a los programadores a desarrollar rápidamente interfaces de usuario sin necesidad de comenzar desde cero. Bootstrap proporciona una colección de herramientas CSS y JavaScript, que permiten crear diseños modernos y adaptables a diferentes tamaños de pantalla (responsive), como smartphones, tablets y computadoras de escritorio.

Entre las características clave de Bootstrap se destacan:

- Sistema de cuadrícula responsiva: Bootstrap incluye un sistema de cuadrícula de 12 columnas que facilita la creación de diseños adaptables a diferentes resoluciones de pantalla. Esto permite que los sitios web se ajusten

automáticamente según el dispositivo en el que se visualizan.

- **Componentes predefinidos:** Bootstrap ofrece una amplia variedad de componentes de interfaz de usuario ya creados, como botones, menús de navegación, modales, formularios, alertas y más. Estos componentes pueden personalizarse fácilmente y son de uso inmediato.
- **Compatibilidad con navegadores:** Bootstrap asegura que los sitios web creados con este framework sean compatibles con la mayoría de los navegadores modernos, garantizando una experiencia de usuario consistente sin importar el navegador utilizado.
- **Diseño móvil primero:** Bootstrap utiliza un enfoque mobile-first, lo que significa que el diseño y la estructura del sitio se optimizan inicialmente para dispositivos móviles, adaptándose luego a pantallas más grandes.
- **Personalización:** Aunque Bootstrap viene con estilos predeterminados, permite la personalización a través de variables de Sass o Less, lo que facilita la adaptación del diseño a las necesidades específicas del proyecto.
- **Extensiones y plugins de JavaScript:** Además de los componentes de CSS, Bootstrap incluye varios plugins de JavaScript que permiten agregar funcionalidad interactiva como ventanas modales, carruseles de imágenes, menús desplegados, y otros elementos interactivos, todo sin tener que escribir código desde cero. (Bootstrap. (s.f.). Introduction to Bootstrap.).

2.9.4 Herramientas de diseño

❖ HTLM5

HTML5 (HyperText Markup Language 5) es la quinta versión del lenguaje de marcado estándar utilizado para estructurar y presentar contenido en la web. HTML5 introduce nuevas etiquetas, atributos y funcionalidades que permiten crear sitios web más dinámicos, interactivos y multimedia sin necesidad de complementos adicionales como Flash. Esta versión fue diseñada para mejorar la semántica del código, la accesibilidad y ofrecer compatibilidad con dispositivos móviles.

Entre sus principales características destacan:

- Nuevas etiquetas semánticas como `<header>`, `<article>`, `<section>`, y `<footer>`, que permiten estructurar mejor el contenido y mejorar su legibilidad tanto para los desarrolladores como para los motores de búsqueda.
- Soporte para multimedia nativa: HTML5 incluye etiquetas como `<audio>` y `<video>`, que permiten reproducir archivos multimedia directamente en el navegador sin necesidad de instalar plugins externos.
- Mejoras en el manejo de formularios: Nuevos tipos de entrada como `email`, `date`, `number`, y validaciones automáticas de formularios, que simplifican el desarrollo y mejoran la experiencia del usuario.
- Canvas y gráficos: HTML5 introduce la etiqueta `<canvas>`, que permite dibujar gráficos y animaciones directamente en la página web, ideal para juegos y gráficos interactivos.
- Compatibilidad y accesibilidad: HTML5 está diseñado para ser compatible con todos los navegadores modernos y dispositivos móviles, permitiendo crear aplicaciones web responsivas y accesibles para diferentes tamaños de pantalla.

❖ CSS3

CSS3 (Cascading Style Sheets, versión 3) es la última evolución del lenguaje utilizado para diseñar y dar estilo a las páginas web. CSS3 introduce nuevas características y módulos que permiten controlar la apariencia de los elementos HTML de forma más avanzada y con menos código, mejorando la presentación visual de sitios web.

Entre sus principales características destacan:

- Soporte para animaciones y transiciones CSS3 permite animar elementos de la página sin necesidad de usar JavaScript, mejorando la interactividad de las interfaces de usuario.
- Nuevos selectores y pseudo-clases CSS3 facilita la selección precisa de elementos HTML para aplicar estilos, permitiendo diseños más específicos y dinámicos.
- Diseño flexible y responsivo gracias a propiedades como flexbox y grid, CSS3 permite crear diseños adaptables que se ajustan automáticamente a diferentes tamaños de pantalla y dispositivos.
- Soporte para sombras y gradientes CSS3 añade efectos visuales como sombras (box-shadow, text-shadow) y gradientes, mejorando el aspecto gráfico sin necesidad de imágenes externas.
- Media Queries esta característica permite aplicar diferentes estilos según el tamaño de la pantalla o el dispositivo, esencial para el diseño responsivo.

2.10 CALIDAD DEL SOFTWARE

La calidad del software se define como el grado en el que un sistema, componente o

proceso cumple con los requisitos especificados y las expectativas o necesidades del cliente o usuario. Según el IEEE Std. 610-1990, es "la concordancia del software con los requerimientos explícitamente establecidos, los estándares de desarrollo predefinidos y los requerimientos implícitos no formalizados, pero esperados por el usuario". De acuerdo con Pressman (2002), la calidad de un software implica no solo su adecuación a los requisitos técnicos, sino también a las expectativas no expresadas formalmente por el cliente.

- ❖ **CALIDAD:** El concepto de calidad está presente en muchos contextos y busca transmitir una percepción positiva, sugiriendo que algo es mejor. En términos de productos o servicios, la calidad es la percepción que el cliente tiene del mismo, y es una fijación mental que asume conformidad con un estándar. Sin embargo, esta percepción puede cambiar con el tiempo, en función de nuevas necesidades o especificaciones. En resumen, la calidad es la capacidad de un producto o servicio para satisfacer las expectativas del cliente o usuario.

- ❖ **MÉTRICA:** El término "métrica" proviene originalmente de la literatura y la poesía, donde se utilizaba para medir el ritmo y las sílabas de los versos. En informática, sin embargo, la métrica se refiere a la medición del software en base a parámetros predeterminados, como el número de líneas de código o la cantidad de documentación asociada. A veces, las métricas también se conocen como "indicadores".

Entre las métricas más comunes en el desarrollo de software encontramos:

- **Índice de productividad:** $\text{tamaño} / \text{esfuerzo} = \text{líneas de código generadas} / \text{horas trabajadas}$.

- **Tasa de defectos:** $\text{defectos} / \text{tamaño} = \text{número de errores} / \text{líneas de código}$

generadas. (Pressman, 2002)

Las métricas de calidad del software se han centrado tradicionalmente en medir procesos, productos y recursos. Actualmente, las métricas se usan también para monitorear y optimizar aplicaciones en tiempo de ejecución, buscando cuellos de botella y midiendo capacidades máximas.

2.10.1 Estándar ISO/IEC 25000

Este estándar norma ISO/IEC 25000, también conocida como SQuaRE (System and Software Quality Requirements and Evaluation), es una serie de estándares internacionales que proporcionan un marco para la evaluación y gestión de la calidad de productos de software. Su objetivo principal es unificar y enriquecer los procesos de especificación de requisitos de calidad y evaluación del software, apoyándose en la medición de la calidad del mismo.

❖ Métricas de Calidad del Modelo ISO/IEC 25000

La norma ISO/IEC 25000 se enfoca en evaluar la calidad de un software mediante un conjunto de métricas agrupadas en características esenciales. Estas características incluyen funcionalidad, que mide si el software cumple con los requerimientos establecidos en términos de precisión, seguridad y cumplimiento normativo, asegurando su adecuada operación e interacción con otros sistemas. Esta área es fundamental para verificar que el software no solo funcione bien por sí mismo, sino que también lo haga en un contexto más amplio de aplicaciones y servicios.

Otra área crucial es la fiabilidad, que contempla métricas para medir la madurez del sistema, su disponibilidad y su tolerancia a fallos. Además, abarca la usabilidad, la cual evalúa lo intuitivo y fácil que es para los usuarios entender y operar el sistema, proporcionando también una interfaz atractiva y amigable. En este aspecto, se asegura que el software sea

accesible y eficiente en el uso, maximizando así la satisfacción del usuario final.

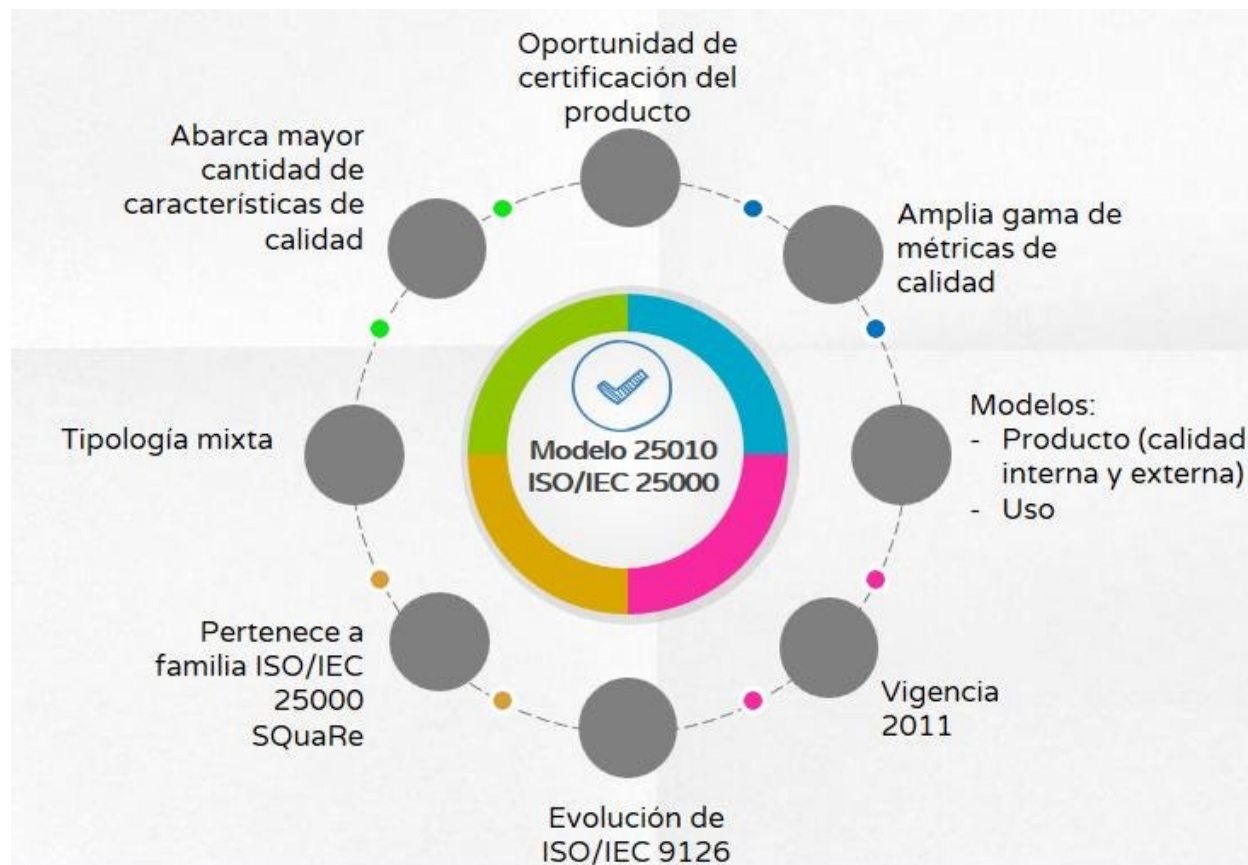
Finalmente, la norma considera métricas de eficiencia, mantenibilidad y portabilidad. La eficiencia evalúa el rendimiento y uso de recursos del sistema, mientras que la mantenibilidad mide la facilidad para realizar modificaciones y reparaciones en el software. La portabilidad, por su parte, se enfoca en la capacidad del software para adaptarse a diferentes entornos y plataformas sin perder funcionalidad. En conjunto, estas métricas permiten que el software sea evaluado de manera integral, asegurando calidad en términos de funcionamiento, rendimiento y adaptabilidad. (Galín, 2004)

❖ Norma de Evaluación ISO/IEC 25000

Esta norma proporciona un marco para la evaluación de la calidad del software mediante un conjunto de estándares estructurados en cinco divisiones clave: gestión de calidad, modelo de calidad, medición de calidad, requisitos de calidad y evaluación de calidad. Esta norma permite a las organizaciones especificar requisitos claros de calidad, así como medir y evaluar aspectos como funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad del software. En conjunto, ISO/IEC 25000 ayuda a asegurar que los productos de software cumplan con altos estándares de calidad, alineándose con las necesidades de los usuarios y los objetivos de negocio. (International Organization for Standardization. 2014).

Figura 10.

Modelo de ISO 250010 ISO/IEC/ 25000



Nota. Modelo de ISO 250010 ISO/IEC, Tulia Vaca, 2018

❖ Evaluación Interna, externa y Calidad de Uso ISO/IEC 25000

La norma ISO/IEC 25000 establece un marco de evaluación de la calidad de software en tres dimensiones: calidad interna, calidad externa y calidad en uso. La calidad interna se refiere a las características del software que pueden evaluarse sin ejecutarlo, como la estructura y el diseño del código. La calidad externa implica medir características observables durante la ejecución, como el rendimiento y la estabilidad, en un entorno de pruebas. Por último, la calidad en uso se enfoca en cómo el software satisface las necesidades del usuario final en términos de efectividad, eficiencia y satisfacción, cuando se usa en condiciones reales. Estos tres niveles aseguran una evaluación completa del software desde su desarrollo hasta su implementación

final. (Standarization.2014, 2014)

- ▶ **Funcionalidad.** Engloba las funciones que satisfacen las necesidades cuando el producto es utilizado correctamente, y se divide en las siguientes subcategorías: completitud funcional, corrección funcional y pertinencia funcional.
- ▶ **Rendimiento.** Se refiere a la eficiencia, medida a través del comportamiento en el tiempo y la utilización de recursos.
- ▶ **Compatibilidad.** La capacidad de que dos o más sistemas compartan el mismo software. Esta categoría incluye dos subcaracterísticas: coexistencia e interoperabilidad.
- ▶ **Fiabilidad.** Un sistema es fiable cuando satisface los requisitos de madurez, disponibilidad, tolerancia a fallos y capacidad de recuperación.
- ▶ **Usabilidad.** Representa la facilidad de aprendizaje y uso del software. Incluye las siguientes subcategorías: capacidad para reconocer su adecuación, capacidad de aprendizaje, capacidad para ser usado, protección contra errores de usuario, estética de la interfaz de usuario y accesibilidad.
- ▶ **Seguridad.** Asegura la protección de los datos y la información para evitar accesos no autorizados. Las subcategorías son: confidencialidad, integridad, responsabilidad, autenticidad y no repudio.
- ▶ **Mantenibilidad.** Incluye aspectos relacionados con la modificación del software. Sus subcategorías son: modularidad, reusabilidad, analizabilidad, capacidad para ser modificado y capacidad para ser probado.

2.11 ANALISIS DE COSTO DE SOFTWARE

La estimación de los costos de desarrollo de software es un elemento crucial en el análisis de proyectos informáticos, ya que permite contar con indicadores clave para medir el costo, garantizando eficiencia, calidad y competitividad. El análisis de costos implica identificar los recursos necesarios para realizar el trabajo o proyecto de manera eficiente.

La evaluación de los costos incluye la determinación de la cantidad y calidad de los recursos en términos de dinero, esfuerzo, capacidades, conocimientos y tiempo, factores que influyen directamente en la gestión empresarial.

Actualmente, existen diversas métricas que pueden ser aplicadas a cualquier tipo de proyecto de software para calcular los costos, aunque muchas no se utilizan frecuentemente. Este análisis propone diseñar un conjunto de métricas para calcular el costo en el proceso de desarrollo de software, tomando en cuenta las características específicas del proyecto.

Hoy en día, existen múltiples herramientas y metodologías que permiten realizar estimaciones de costos, aunque hay diversos factores que pueden afectar dichas estimaciones, tales como:

- Incertidumbre en los requerimientos del proyecto.
- Términos contractuales inflexibles.
- Factores financieros que afectan la toma de decisiones (por ejemplo, ganar licitaciones sacrificando costo y tiempo).
- Falta de experiencia con ciertas tecnologías o herramientas.

2.11.1 COCOMO II

El modelo COCOMO (Constructive Cost Model) es una metodología de estimación de costos para el desarrollo de software, siendo una de las ampliaciones más

importantes de su versión original, COCOMO 81, publicada en 1981. Este modelo toma en cuenta diferentes enfoques y niveles en el desarrollo de software, proporcionando estimaciones detalladas de manera incremental. Además, COCOMO apoya el uso del modelo de desarrollo en espiral, que permite iteraciones continuas para refinar el producto.

$$E = a(KLDC)^b, \text{persona-mes}$$

$$D = c(E)^d, \text{meses}$$

$$P = \frac{E}{D}, \text{personas}$$

Para calcular el punto de función se utiliza la siguiente relación:

$$PF = \text{Cuenta Total} * (X + \text{Min}(Y) * \sum Fi)$$

La fórmula para el cálculo de LDC (Líneas de Código) es la siguiente ecuación:

$$LDC = PFNA * \text{Factor LDC/PFNA}$$

La fórmula para el cálculo de KLDC (Miles de Líneas de Código) está dado por:

$$KLDC = LDC / 1000$$

2.11.1.1 MODELOS DE COCOMO II

Los tres modelos de COCOMO II están diseñados para adaptarse a las necesidades específicas de diversos sectores y para utilizar la información disponible en cada etapa del ciclo de vida del desarrollo de software. Estos modelos permiten manejar diferentes niveles de granularidad de la información. Los tres enfoques principales de COCOMO II son:

- **Modelo de Composición de Aplicación:** Este modelo se emplea durante las primeras etapas de la ingeniería del software. Es especialmente útil cuando se

están realizando tareas como el prototipado de interfaces de usuario, la interacción entre el sistema y el software, la evaluación del rendimiento y la madurez de la tecnología utilizada. En esta fase, la rapidez en la toma de decisiones y la flexibilidad son clave para ajustar el diseño y las funcionalidades iniciales.

- **Modelo de Fase de Diseño Previo:** Se utiliza una vez que los requisitos del software se han estabilizado y la arquitectura básica del sistema se ha definido. Este modelo proporciona estimaciones más detalladas, enfocándose en la estructura y funcionalidad del software conforme a su diseño final. Es adecuado para proyectos en los que los requisitos ya están bien establecidos y se necesita una planificación más precisa antes de comenzar la implementación.
- **Modelo de Fase Posterior a la Arquitectura:** Este modelo se aplica durante la construcción del software, cuando el diseño ya está definido y se necesita estimar los costos y el esfuerzo durante las etapas de implementación y prueba. Aporta un mayor grado de precisión en las estimaciones y se adapta a los proyectos más avanzados donde la arquitectura y el diseño ya están completamente definidos.

Cada uno de estos modelos se orienta a etapas específicas del ciclo de vida del desarrollo de software, lo que permite ajustarse a las necesidades y características particulares del proyecto en cada fase.

2.11.1.2 SECTORIZACION DE APLICACIONES EN COCOMO II

Figura 11.

Sectorización de aplicaciones en COCOMO II

Aplicaciones desarrolladas por usuarios finales		
Generadores de Aplicaciones	Aplicaciones con Componentes	Sistemas Integrados
Infraestructura		

Nota. COCOMO II, Gracia, 2012

APLICACIONES DESARROLLADAS POR USUARIOS FINALES: En este sector se encuentran las aplicaciones que los usuarios finales desarrollan directamente utilizando herramientas como hojas de cálculo, sistemas de consultas y otros generadores de aplicaciones. Estas aplicaciones suelen surgir debido al uso masivo de estas herramientas, así como a la necesidad de soluciones rápidas y flexibles que satisfagan demandas inmediatas. Este tipo de desarrollo se caracteriza por su facilidad de uso y rápida implementación.

GENERADORES DE APLICACIONES: Aquí se incluyen los módulos preempaquetados que pueden ser utilizados tanto por usuarios finales como por programadores. Estos generadores permiten la creación rápida de aplicaciones mediante la configuración o modificación de módulos ya existentes, sin la necesidad de desarrollar desde cero. Son muy útiles para proyectos que requieren rapidez en la entrega de soluciones.

APLICACIONES CON COMPONENTES: Este sector abarca aplicaciones construidas mediante la integración de componentes preempaquetados. Aunque son soluciones simples, estas aplicaciones pueden ser construidas rápidamente a partir de componentes interoperables, como interfaces gráficas, administradores de bases de datos y buscadores de datos. Por lo general, estas aplicaciones son desarrolladas por equipos pequeños y en plazos de semanas o pocos meses.

SISTEMAS INTEGRADOS: En este sector se encuentran los sistemas de gran escala, altamente integrados y sin antecedentes previos que puedan ser utilizados como referencia. Estos sistemas suelen implicar un alto grado de complejidad y pueden incluir la combinación de aplicaciones preexistentes con componentes nuevos. El desarrollo de estos sistemas puede abarcar múltiples equipos y se extiende durante largos períodos.

INFRAESTRUCTURA: Este sector abarca el desarrollo de sistemas operativos, protocolos de redes y sistemas administradores de bases de datos. Estos sistemas son esenciales para el funcionamiento de otras aplicaciones y su desarrollo requiere una gran inversión en términos de tiempo y recursos, ya que constituyen la base sobre la cual se ejecutan otros sistemas y aplicaciones

2.12 SEGURIDAD DE LA INFORMACION

La seguridad de la información, también conocida como ciberseguridad o seguridad de tecnología de la información, se enfoca en la protección de la infraestructura tecnológica y los datos que circulan a través de redes y sistemas computacionales. Este campo abarca tanto la protección de hardware, software, redes, como de la información valiosa de las organizaciones, que podría ser utilizada indebidamente si llegase a manos no autorizadas.

El concepto de seguridad informática no debe confundirse con seguridad de la información. Mientras que la seguridad informática se limita a la protección en el ámbito digital, la seguridad de la información considera la protección de los datos en cualquier forma, ya sea en medios electrónicos o físicos.

En resumen, la seguridad en un entorno de red implica identificar y eliminar vulnerabilidades para proteger tanto la infraestructura como la información valiosa de una organización. Cada empresa debe definir una política de seguridad que cubra sus necesidades específicas, asegurando una adecuada coordinación y control de los recursos. Según

(Wikipedia, 2015).

❖ Estándar ISO/IEC 27000

El estándar ISO/IEC 27000 exige niveles de seguridad informática adecuados para empresas que trabajan en entornos digitales, como el comercio electrónico. Para mantener una gestión de la seguridad de la información efectiva, este estándar se actualiza regularmente, abordando nuevas brechas de seguridad y proporcionando pautas que las empresas deben seguir para minimizar riesgos. La certificación ISO 27000 es esencial para organizaciones que desean garantizar niveles altos de protección de sus datos. (Pandiani, 2005)

❖ Estándar ISO/IEC 27002

El estándar ISO/IEC 27002 ofrece directrices sobre las mejores prácticas para gestionar la seguridad de la información, enfocándose en la preservación de la confidencialidad, integridad y disponibilidad de los datos. Este estándar establece un código de buenas prácticas para la implementación de un Sistema de Gestión de Seguridad de la Información (SGSI) dentro de las organizaciones.

El objetivo principal de ISO/IEC 27002 es proporcionar principios y guías generales para iniciar, implementar, mantener y mejorar la seguridad de la información. Además, ayuda en la selección y administración de controles basados en la evaluación de riesgos, lo cual es vital para la protección de los activos tecnológicos más valiosos de una organización. (Pandini W., 2005).

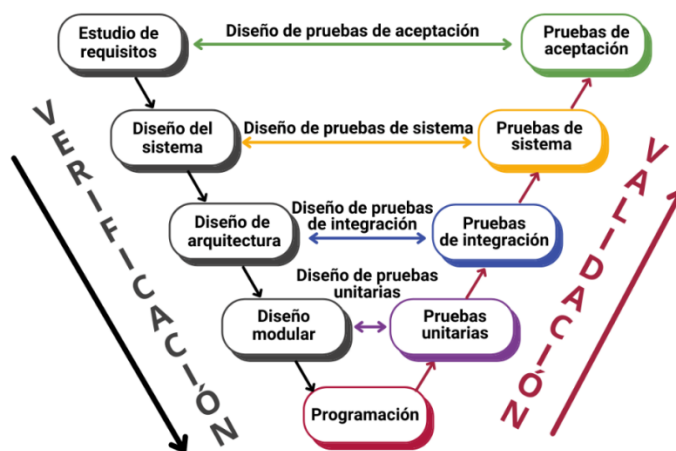
2.13 PRUEBAS DE FUNCIONALIDAD DEL SOFTWARE

El software puede ser evaluado mediante pruebas principales. La idea principal del modelo en V, (V-Model) es que las actividades de desarrollo y de prueba se corresponden entre sí. Cada fase de desarrollo debe completarse con su propia actividad de prueba, cubriendo cada una de ellas un nivel de abstracción diferente: los componentes de software, la integración de

componentes, el sistema de software completo y la aceptación del usuario. Este modelo es, actualmente, uno de los estándares de SQA (aseguramiento de la calidad de software) y se utiliza para describir las actividades de prueba como parte del proceso de desarrollo de software. El V-Model muestra la relación entre cada fase de desarrollo y su correspondiente actividad de prueba. Consiguiendo la calidad de forma interactiva y escalonada, validando y verificando cada uno de los pasos del desarrollo del software. (SIPSA, 2023)

Figura 12.

Modelo en V



Nota. Fases del Modelo en V, SIPSA, 2023

2.13.1 Pruebas Unitarias.

Comenzamos por las pruebas unitarias, también conocidas como unit testing.

Las pruebas unitarias normalmente las realizan los programadores, mediante las cuales se prueban unidades individuales de código fuente para determinar si son aptas para su uso.

Estas pruebas verifican que los componentes más pequeños del programa funcionan correctamente. Dependiendo del lenguaje de programación, un componente puede ser un módulo, una unidad o una clase.

Según esto, las pruebas se llamarán pruebas de módulo, pruebas de unidad o pruebas de clase.

Las pruebas de componentes comprueban la salida del componente después de proporcionar una entrada basada en la especificación del componente.

La principal característica de las pruebas de componentes es que los componentes individuales se prueban de forma aislada y sin interactuar con otros componentes. Como no hay otros componentes implicados, es mucho más fácil localizar los defectos.

Normalmente, la implementación de las pruebas se realiza con la ayuda de un framework de pruebas, por ejemplo, JUnit, CPPUNIT o PyUnit, por nombrar algunos de los frameworks de pruebas unitarias populares.

Además de las pruebas funcionales, las pruebas de componentes pueden referirse a aspectos no funcionales como la eficiencia (por ejemplo, el consumo de almacenamiento) y la mantenibilidad (por ejemplo, la complejidad del código).

Las pruebas unitarias tienen como objetivo verificar la funcionalidad y estructura de cada componente individualmente una vez que ha sido codificado.

Los pasos necesarios para llevar a cabo las pruebas unitarias son los siguientes:

- Ejecutar todos los casos de prueba asociados a cada verificación establecida en el plan de pruebas, registrando su resultado. Los casos de prueba deben contemplar tanto las condiciones válidas y esperadas como las inválidas e inesperadas.
- Corregir los errores o defectos encontrados y repetir las pruebas que los detectaron. Si se considera necesario, debido a su implicación o importancia, se

repetirán otros casos de prueba ya realizados con anterioridad.

La prueba unitaria se da por finalizada cuando se hayan realizado todas las verificaciones establecidas y no se encuentre ningún defecto, o bien se determine su suspensión. (SIPSA, 2023).

2.13.2. Pruebas de Integración

Después de completar las pruebas unitarias se realizan las pruebas de integración.

El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probados unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces, tanto internas como externas, cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados en las verificaciones correspondientes.

Las pruebas pueden abarcar sólo dos componentes específicos, grupos de componentes o incluso subsistemas independientes de un sistema de software. Las pruebas de integración suelen realizarse después de que los componentes hayan sido probados en la fase de pruebas de componentes inferiores.

El desarrollo de las pruebas de integración puede basarse en la especificación funcional, la arquitectura del sistema, los casos de uso o las descripciones del flujo de trabajo. Las pruebas de integración se centran en las interfaces de los componentes (o las interfaces de los subsistemas) y tratan de revelar los defectos que se activan a través de su interacción y que, de otro modo, no se activarían al probar los componentes aislados.

Las pruebas de integración también pueden centrarse en las interfaces con el entorno de la aplicación, como los sistemas de software o hardware externos. Esto suele denominarse pruebas de integración de sistemas, en contraste con las pruebas de integración de

componentes.

En las pruebas de integración se examinan las interfaces entre grupos de componentes o subsistemas para asegurar que son llamados cuando es necesario y que los datos o mensajes que se transmiten son los requeridos.

A menudo se combinan las pruebas unitarias con las de integración, ya que en las unitarias es necesario crear módulos auxiliares que simulen las acciones de los componentes invocados por el que se está probando y también se crean a componentes para establecer las precondiciones necesarias, llamar al componente objeto de la prueba y examinar los resultados de la prueba.

Los tipos principales de integración son:

Integración incremental: se combina el siguiente componente que se debe probar con el conjunto de componentes que ya están probados y se va incrementando progresivamente el número de componentes a probar. Con el tipo de prueba incremental lo más probable es que los problemas que surjan al incorporar un nuevo componente o un grupo de componentes previamente probado, sean debidos a este último o a las interfaces entre este y los demás componentes.

Integración no incremental: se prueba cada componente por separado y posteriormente se integran todos a la vez realizando las pruebas pertinentes. Este tipo de integración se denomina también "Big-Bang" (gran explosión).

Las pruebas de integración se realizan en la fase de diseño de la arquitectura. (SIPSA).

2.13.3. Pruebas de sistema

Estas pruebas permiten probar el sistema en su conjunto y con otros sistemas con los

que se relaciona para verificar que las especificaciones funcionales y técnicas se cumplen.

Tienen la finalidad de verificar cómo se comporta el producto tomando como referencia al usuario final y su interacción con el sistema.

Se llevan a cabo cuando se chequea que la integración de los sistemas actúa correctamente, por lo tanto, se comprueba la funcionalidad y se deben realizar en un ambiente similar al real, verificando que todo funcione de acuerdo con las especificaciones y requisitos planteados desde el principio por el cliente. Reflejan una visión muy similar a su comportamiento en el entorno de producción.

La fase de pruebas del sistema abarca el sistema de software en su conjunto. Mientras que las pruebas de integración se basan principalmente en las especificaciones técnicas, las pruebas del sistema se crean teniendo en cuenta el punto de vista del usuario y se centran en los requisitos funcionales y no funcionales de la aplicación. Estas últimas pueden incluir pruebas de rendimiento, carga y estrés.

Aunque los componentes y su integración ya han sido probados, las pruebas del sistema son necesarias para validar que se cumplen los requisitos funcionales del software. Las pruebas del sistema suelen incluir otros documentos, como la documentación del usuario o los documentos de análisis de riesgos.

El entorno de pruebas del sistema debe ser lo más parecido posible al entorno de producción. Si es posible, hay que utilizar el mismo hardware, sistemas operativos, controladores, infraestructura de red o sistemas externos, y evitar en lo posible los marcadores de posición, simplemente para reproducir el comportamiento del sistema de producción. Sin embargo, el entorno de producción no debe utilizarse como entorno de prueba, ya que cualquier defecto provocado por la prueba del sistema podría repercutir en el sistema de producción.

Las pruebas de sistema deberían ser automatizadas para reducir el tiempo asociado a las pruebas manuales.

Tipos de pruebas de sistema:

- **Pruebas funcionales:** asegurar que el sistema realiza correctamente todas las funciones que se han detallado en las especificaciones dadas por el usuario.
- **Pruebas de comunicaciones:** determinan que las interfaces entre los componentes del sistema funcionan adecuadamente, tanto a través de dispositivos remotos, como locales. Asimismo, se han de probar las interfaces hombre/máquina.
- **Pruebas de rendimiento:** comprueban que los tiempos de respuesta están dentro de los intervalos establecidos en las especificaciones del sistema.
- **Pruebas de volumen:** examinan el funcionamiento del sistema cuando está trabajando con grandes volúmenes de datos, simulando las cargas de trabajo esperadas.
- **Pruebas de sobrecarga:** comprueban el funcionamiento del sistema en el umbral límite de los recursos, sometiéndole a cargas masivas. El objetivo es establecer los puntos extremos en los cuales el sistema empieza a operar por debajo de los requisitos establecidos.
- **Pruebas de disponibilidad de datos:** consisten en demostrar que el sistema puede recuperarse ante fallos, tanto de equipo físico como lógico, sin comprometer la integridad de los datos.

- **Pruebas de facilidad de uso:** comprueban la adaptabilidad del sistema a las necesidades de los usuarios, tanto para asegurar que se acomoda a su modo habitual de trabajo, como para determinar las facilidades que aporta al introducir datos en el sistema y obtener los resultados.
- **Pruebas de operación:** comprueban la correcta implementación de los procedimientos de operación, incluyendo la planificación y control de trabajos, arranque y re arranque del sistema, etc.
- **Pruebas de entorno:** verifican las interacciones del sistema con otros sistemas dentro del mismo entorno.
- **Pruebas de seguridad:** verifican los mecanismos de control de acceso al sistema para evitar alteraciones indebidas en los datos. (SIPSA, 2023).

2.14.4 Pruebas de aceptación

Estas pruebas van dirigidas a comprobar que el sistema cumple los requisitos de funcionamiento esperado, recogidos en la definición de requisitos y en los criterios de aceptación con el fin de conseguir la aceptación final del sistema, desde el punto de vista de su funcionalidad y rendimiento, por parte del usuario.

Las pruebas de aceptación son definidas por el usuario del sistema y preparadas por el equipo de testing, aunque la ejecución y aprobación final corresponden al usuario.

El responsable de usuarios debe revisar los criterios de aceptación que se especificaron previamente en el plan de pruebas del sistema y, posteriormente, dirigir las pruebas de aceptación final.

La validación del sistema se consigue mediante la realización de pruebas de caja negra

que demuestran la conformidad con los requisitos y que se recogen en el plan de pruebas.

Las pruebas de aceptación pueden ser:

- Internas, para una versión de un software que aún no se ha lanzado. Estas pruebas de aceptación internas las realizan quienes no participan en el desarrollo o las pruebas.
- Externas, realizadas por la empresa que pidió el desarrollo o los usuarios finales del software.

En el caso de las pruebas de aceptación por parte del cliente, incluso puede ser responsabilidad de éste, parcial o totalmente, decidir si la versión final del software cumple los requisitos de alto nivel. Se crean informes para documentar los resultados de las pruebas y para la discusión entre la entidad desarrolladora y el cliente.

A diferencia de las pruebas de aceptación por parte del cliente, las pruebas de aceptación por parte del usuario (UAT) pueden ser el último paso antes del lanzamiento final del software. Se trata de una prueba centrada en el usuario para verificar que un software ofrece realmente los flujos de trabajo que se pretenden. Estas pruebas pueden llegar a cubrir aspectos de usabilidad y la experiencia general del usuario.

Las pruebas de aceptación suelen ser manuales y, a menudo, sólo se automatiza una pequeña parte de ellas. Pero si pensamos en el arduo trabajo que supone realizar las pruebas de aceptación de numerosas versiones a lo largo del ciclo de vida del software o de las numerosas iteraciones en los proyectos ágiles, a medio y largo plazo, la automatización de pruebas sería la mejor solución. (SIPSA, 2023).

CAPITULO III

Marco Practico

INGENIERIA
DE SISTEMAS
UNIVERSIDAD PUBLICA DE EL ALTO



CAPITULO III

3.1 INTRODUCCION

El propósito de capítulo es poner en práctica lo mencionado en los anteriores capítulos, que se analiza, desarrolla la metodología UWE, que satisfaga las necesidades que se tiene en la unidad de servicios generales y mantenimiento, de GAMEA.

El uso de la metodología UWE implica la realización de actividades por cada uno de sus modelos, a continuación, se describe por cada modelo de la metodología.

3.2 ANALISIS DE LA SITUACION ACTUAL

- **Solicitud de Mantenimiento**

Las Unidades Educativas de la ciudad de El Alto, que abarcan los catorce distritos de la urbe, envían solicitudes para el mantenimiento e instalaciones relacionadas con infraestructura. Estas solicitudes son recibidas y gestionadas por la oficina de la unidad de mantenimiento.

- **Inspección y Requerimiento de Material**

Una vez recibida la solicitud, el inspector de la unidad de Mantenimiento visita la unidad educativa para evaluar el tipo de trabajo necesario y el material requerido para llevarlo a cabo. Los problemas de infraestructura suelen deberse al desgaste por el tiempo, daños causados por estudiantes o por condiciones climáticas adversas.

- **Requerimiento de Materiales**

Tras la inspección y evaluación, una unidad educativa puede presentar diferentes problemas de infraestructura en una misma revisión. Los materiales necesarios

pueden ser proporcionados por la propia unidad educativa, en cuyo caso la junta escolar aporta los materiales y la unidad de mantenimiento proporciona la mano de obra. En situaciones donde la unidad educativa no dispone de materiales, pero cuenta con el presupuesto anual asignado de 1600 bs por la alcaldía, pueden obtener los materiales del almacén de la unidad de mantenimiento de la alcaldía.

- **Cronograma de Trabajo**

Una vez completada la inspección y definidos los materiales necesarios, el inspector asigna cuadrillas de trabajo según el tipo de intervención requerida. Las cuadrillas realizan el trabajo en la unidad educativa en coordinación con la junta escolar.

- **Conclusión e Informe del Trabajo**

Al finalizar el trabajo, se entrega una carta de conformidad firmada por la junta escolar, que confirma la satisfacción con el trabajo realizado. Posteriormente, se elabora un informe detallado sobre la intervención.

- **Análisis del Escenario Actual**

En base a estudios de campo y el modelo operativo, se observa que los inspectores y encargados de la unidad de mantenimiento no cuentan con datos precisos sobre las unidades educativas atendidas o pendientes de atención, debido a una gestión y coordinación inadecuada con la junta escolar. Además, el registro de datos se realiza manualmente y en hojas de Excel, lo cual dificulta la gestión eficiente de la información.

3.3 INGENIERIA DE REQUERIMIENTOS

En esta etapa del desarrollo de software, se identificaron procesos que actualmente se realizan de manera manual, tanto en el registro como en la generación de informes. Esto incluye la gestión de solicitudes recibidas, el estado de intervenciones y los trabajos realizados. Esta metodología manual ocasiona ineficiencias en la administración de los trabajos, lo que prolonga el tiempo necesario para ejecutar estas tareas. Como resultado, se generan retrasos en las inspecciones, en la atención de solicitudes de mantenimiento y, en algunos casos, las unidades educativas quedan desatendidas, afectando la calidad de sus instalaciones y el bienestar del estudiantado.

3.3.1 Obtención de requisitos

La descripción general del escenario, junto con las características específicas del proyecto planteado, constituye la base fundamental para la identificación de los requisitos. Es esencial realizar un análisis detallado de estos requisitos antes de proceder con el diseño de la aplicación.

Los requisitos generales identificados establecen las características que debe tener la aplicación para proporcionar un manejo eficiente y accesible de la información para los distintos usuarios del sistema, en particular el jefe de unidad, el responsable de mantenimiento, las cuadrillas operarias y los encargados de almacén. Estos requisitos incluyen los siguientes puntos clave:

- Entrega de Información Actualizada el sistema debe mantener la información constantemente actualizada para facilitar la toma de decisiones.
- Administración de Información Accesible la interfaz debe facilitar el acceso a la información y proporcionar una presentación clara y adecuada de los contenidos del sistema.

- Interfaz de Actualización Intuitiva la interfaz destinada a la actualización de la información debe ser fácil de usar, permitiendo a los usuarios actualizar datos sin dificultad.
- De forma complementaria a los requisitos generales, el sistema debe cumplir con los siguientes aspectos:
- Accesibilidad Remota debe ser accesible desde cualquier ubicación para permitir la consulta de información en tiempo real.
- Seguridad y Autenticación el sistema web debe proporcionar seguridad en el acceso, con autenticación personal para cada usuario.

3.3.1.1 Roles de Administración Funcional del Sistema

La organización funcional del sistema debe estructurarse en torno a roles específicos de administración:

- **Administración Principal:** Responsable de gestionar los permisos de los usuarios, asignando los niveles de acceso necesarios.
- **Inspector:** Con permisos para gestionar solicitudes de mantenimiento, inspecciones, cronogramas e intervenciones, además de administrar los datos de las unidades educativas solicitantes. El inspector puede registrar y actualizar la información sobre las intervenciones realizadas, así como generar informes completos sobre el estado de intervenciones en curso o finalizadas.
- **Encargado de Almacenes:** Con acceso para registrar entradas y salidas de materiales, generar informes y actualizar datos de cantidades y estados de los materiales.

- **Cuadrillas:** Habilitado para registrar los trabajos realizados en la unidad educativas

3.3.1.2 Definición de actores

Las identificaciones de los actores, nos permite conocer a los personajes involucradas en el proceso del sistema de gestión y control de incidencias de las unidades educativas.

Tabla 1.

Definición de actores

ACTORES	DESCRIPCIONES
Encargado de Unidad de Mantenimiento	Tiene las siguientes Funciones dentro de la unidad: <ul style="list-style-type: none"> · Le llegan las solicitudes de mantenimiento y gestiona las cuadrillas para los trabajos · Coordina las inspecciones y el trabajo con las juntas escolares · Realiza las inspecciones · Realiza los detalles de las intervenciones realizadas
Inspector	Tiene las siguientes Funciones dentro de la unidad: <ul style="list-style-type: none"> · Realiza visitas para evaluar la incidencia de la unidad educativa · Evalúa el material a necesitar para el trabajo · Realiza una coordinaciones con la junta escolar · Designa el tipo de trabajo a las cuadrillas
Encargado de almacenes	Tiene las siguientes Funciones dentro de la unidad: <ul style="list-style-type: none"> · Ingresa pedidos de materiales de unidades educativas · Despacha el material para las unidades educativas · Coordina la entrega de material con el inspector
Personal de Cuadrillas	Tiene las siguientes Funciones dentro de la unidad: <ul style="list-style-type: none"> · Realizan el trabajo de mano de obra de las unidades educativas.

Administrador (Encargado de Unidad de Mantenimiento)

Gestión de usuarios el administrador basado en la lista de encargados de la unidad y los demás usuarios del sistema, esto para que cada usuario tenga diferentes privilegios.

- Usuarios
- Password
- Nivel

Inspector

Gestiona las solicitudes de incidencias que llegan a la unidad de mantenimiento realiza así mismo las visitas a las unidades educativas evalúa el trabajo a realizar, analiza los requerimientos de materiales, realiza los cronogramas para realizar el trabajo, asigna a las cuadrillas para el trabajo junto con las juntas escolares.

Encargado de almacén

Tiene la función principal de distribuir el material a las unidades educativas previa evaluación del inspector despachando el material registrando el material saliente dependiente de cada inspección ya realizada.

Personal de Cuadrillas

La función de las cuadrillas es realizar el trabajo estas cuadrillas se dividen en 4 grupos: tingladistas, albañilería de obra bruta y fina, eléctricos, y plomería, estos son designados por los inspectores para realizar el trabajo en las unidades educativas.

3.3.1.2. Lista de requerimiento del sistema

La obtención correcta del requerimiento que se puede llegar a describir con claridad, en forma consistente, el comportamiento del sistema. Las funciones que deben realizarse se clasifican en tres categorías como se detallan de la siguiente manera:

Evidente: Alerta de confirmación, el usuario debe saber que la acción se ha realizado

Oculto: Aunque no es visible para los usuarios esto aplica a varios módulos donde la información es guardada en un mecanismo persistente de almacenamiento oculto.

Superflua: Esta función es opcional en el cual no repercuten la forma significativa en el costo ni en otras funciones.

3.3.1.2.1. Requerimientos funcionales

Los requisitos fundamentales son las características que necesita el sistema a partir de la información obtenida como parte de la tarea de obtención de requisitos:

Tabla 2.

Lista de requerimiento del sistema

Ref.	Función	Categoría
R.1	Control de acceso seguro y diferenciado por tipo de usuarios o rol.	Evidente
R.2	Gestión de Usuario del Sistema.	Evidente
R.3	Desplegar Vistas y menús de acuerdo al cargo de cada usuario.	Oculto
R.4	Determinar el estado de cada usuario activado.	Evidente
R.5	Registrar al personal con su cargo	Evidente
R.6	Registrar a la cuadrilla	Evidente
R.7	Registrar las solicitudes de incidencias	Evidente
R.8	Registrar solicitantes de las juntas escolares.	Evidente
R.9	Registrar las inspecciones realizadas	Evidente
R.10	Registrar el material a necesitar	Evidente
R.11	Registrar el cronograma	Evidente
R.12	Registrar las intervenciones por unidades educativas.	Evidente
R.13	Registrar reclamos y demandas	Evidente
R.14	Listar por cargo y cuadrilla a la que pertenece	Evidente
R.15	Generar Informes.	Evidente

3.3.1.2.2 Requerimientos no Funcionales

El requerimiento no funcional Muestra su comportamiento en la siguiente tabla.

Tabla 3.

Requerimientos no funcionales

Ref.	Función	Categoría
R.1	El sistema debe visualizarse y funcionar correctamente en cualquier Navegador, Internet Explorer, Mozilla Firefox, Chrome.	Evidente
R.2	El sistema no debe tardar más de diez segundos en mostrar los resultados de una búsqueda.	Evidente
R.3	El sistema debe tener seguridad en el acceso a la información del sistema bajo los roles del usuario.	Evidente

3.3.2 Especificación de requerimientos para la aplicación

Se presenta un análisis de los requerimientos identificados en el lugar de trabajo mediante entrevistas y observación directa de los problemas. Utilizando el estándar IEEE-830, se han especificado los requisitos para el software propuesto. La aplicación web deberá emplear herramientas propias de la web para soportar un modelo distribuido cliente servidor, e incluir componentes esenciales para su diseño, desarrollo e implementación.

Tabla 4.

Requerimientos para la aplicación

Tipo	Nombre	Fase
Lenguaje de programación	PHP	Etapa de desarrollo de la aplicación
Gestor de base de datos	PostgreSQL	Etapa de implementación de la aplicación
Framework	Laravel	Etapa de funcionamiento de la aplicación
Servidor de aplicaciones web	Apache	Funcionamiento de la aplicación
Entorno de desarrollo	PHP	Etapa implementación en la etapa de desarrollo
Diseño Grafico	HTML 5 Css3	Etapa de la implementación en el desarrollo de la aplicación

3.4 ANALISIS DE REQUERIMIENTO Y DISEÑO DEL SISTEMA

3.4.1 Modelo de caso de uso

En esta fase, se analiza el sistema mediante la construcción de diagramas de casos de uso, describiendo los pasos y actividades que conforman los procesos del sistema.

Diagrama de caso de uso

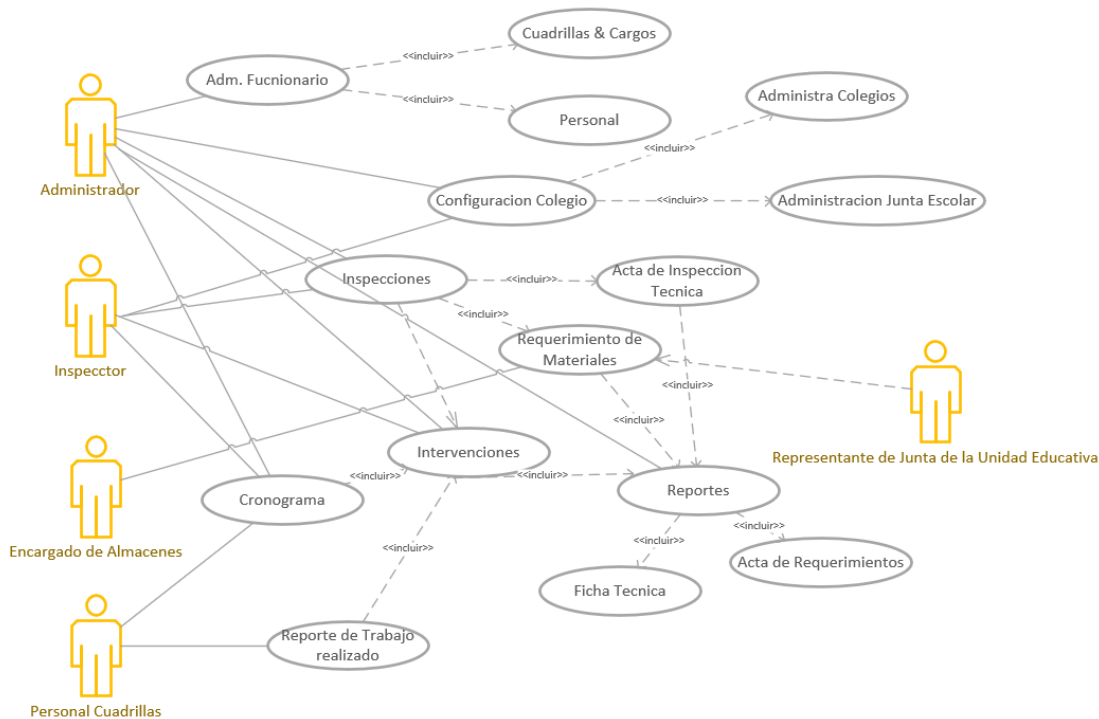
A continuación, se realiza el modelado para ilustrar cómo interactúan los actores con los casos de uso. Se presenta una visión preliminar de los actores mediante diagramas de casos de uso.

- ❖ Modelo de diagrama de caso de uso: General del Sistema
- ❖ Diagrama de caso de uso: Administrador/ Encargado de Unidad de Mantenimiento
- ❖ Diagrama de caso de uso: Inspector
- ❖ Diagrama de caso de uso: Encargado de almacenes
- ❖ Diagrama de caso de uso: Personal de Cuadrillas

a) Modelo de diagrama de casos de uso: General del Sistema

Figura 13.

Modelo de diagrama de caso de uso General del sistema



b) Diagrama de caos de uso Administrador

En esta parte presentaremos los movimientos y privilegios que tiene el administrador principal y las relaciones que tiene con los otros módulos.

Figura 14.

Diagrama de caso de uso: Administrador Principal

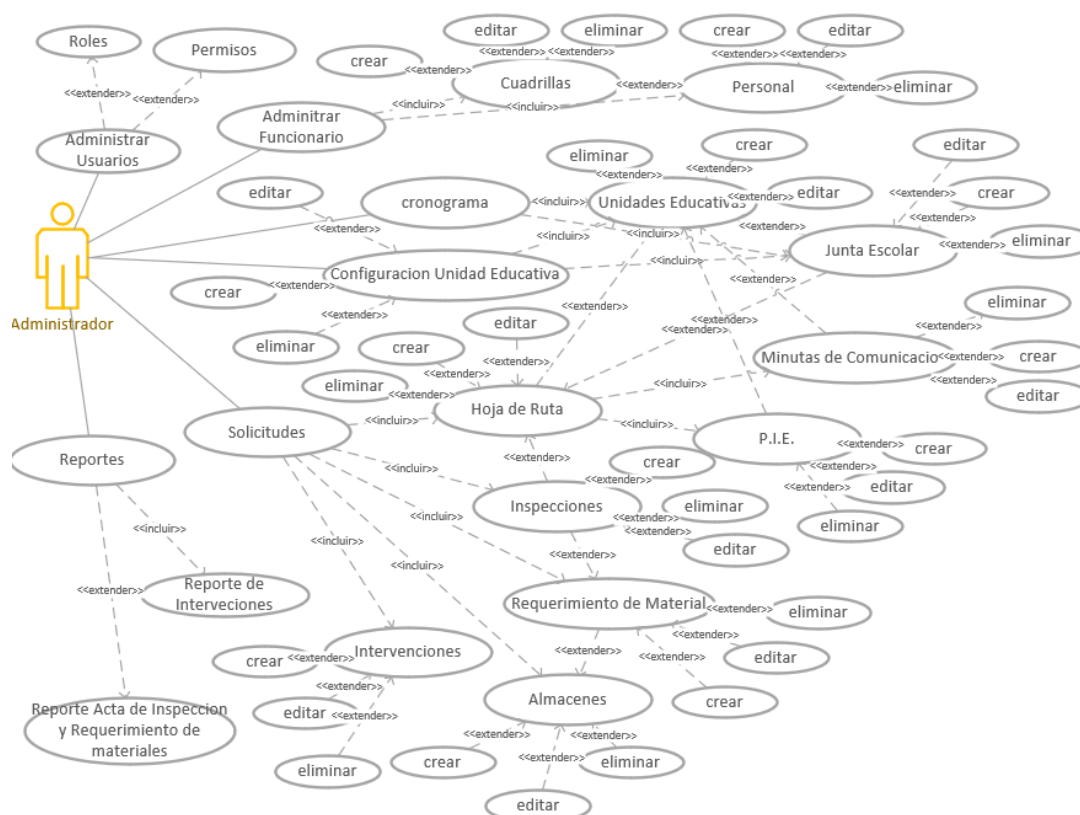


Tabla 5.

Caso de uso de Administrador Principal

Caso de uso: Administrador principal	
Actores:	Administrador
Tipo:	Primario esencial
Descripción:	<p>El Administrador Principal tiene las siguientes opciones de modulo como ser:</p> <p>Módulo de Solicitudes: Este módulo agrupa tres submódulos principales: Hoja de Ruta, Minuta de Comunicación y P.I.E. A continuación, se describen sus funciones:</p> <ul style="list-style-type: none"> • Hoja de Ruta: Permite registrar todas las solicitudes provenientes de las Unidades Educativas. • Minuta de Comunicación y P.I.E.: Ambos submódulos tienen funciones similares y están diseñados para recibir solicitudes y demandas relacionadas con incumplimientos. <p>Cada submódulo permite registrar, editar, eliminar y visualizar registros.</p> <p>Módulo de Mantenimiento: Este módulo incluye cuatro submódulos: Proceso del Trabajo, Inspecciones, Requerimientos de Material, Cronogramas e Intervenciones. A continuación, se detallan sus funciones:</p> <ul style="list-style-type: none"> • Inspecciones: Permite registrar las visitas realizadas a las Unidades Educativas, documentando los problemas encontrados en cada visita. • Requerimientos de Material: Este submódulo registra la solicitud de materiales necesarios según los problemas identificados previamente durante las inspecciones. • Cronogramas: Administra la asignación de fechas para que las cuadrillas realicen el trabajo de mantenimiento programado. • Intervenciones: Registra los trabajos realizados y concluidos en las Unidades Educativas. <p>Cada submódulo permite registrar, editar, eliminar y visualizar registros.</p>

Módulo de Recursos: Este módulo incluye el submódulo de Almacenes, que permite registrar el material entregado a las Unidades Educativas, incluyendo la cantidad y detalles. El submódulo permite registrar, editar y eliminar registros de material.

Módulo de Trabajo de Cuadrillas: Este módulo incluye la opción de Subir Imágenes de Trabajos, donde se registran las cuadrillas que acudieron a realizar trabajos en la Unidad Educativa. Permite registrar detalles como la descripción del trabajo realizado y adjuntar fotografías, las cuales se vinculan con el módulo de Intervenciones. Cada registro puede ser creado, editado, eliminado y visualizado para revisar los detalles del trabajo efectuado.

Módulo de Personal y Cuadrillas: Este módulo incluye dos secciones principales:

- **Personal:** Almacena la información del personal encargado de realizar inspecciones, trabajos, y la gestión de almacenes.
- **Cuadrillas:** Permite asignar al personal existente, especialmente a aquellos en roles de obreros, a una cuadrilla según su cargo.

Ambas secciones permiten registrar, editar y eliminar registros de manera eficiente.

Módulo de Unidades Educativas: Este módulo contiene dos secciones desplegadas:

- **Unidad Educativa:** Permite registrar las Unidades Educativas, incluyendo los datos y detalles importantes de cada una.
- **Junta Escolar:** Registra las juntas escolares de cada Unidad Educativa, con las cuales se coordina para llevar a cabo los trabajos.

Ambas secciones permiten registrar, editar y eliminar registros.

Módulo de Personal y Cuadrillas: Este módulo incluye dos secciones principales:

- **Personal:** Almacena la información del personal encargado de realizar inspecciones, trabajos, y la gestión de almacenes.
- **Cuadrillas:** Permite asignar al personal existente, especialmente a aquellos en roles de obreros, a una cuadrilla según su cargo.

Ambas secciones permiten registrar, editar y eliminar registros de manera eficiente.

Módulo de Administración: Este módulo incluye las secciones de Usuarios, Roles y Permisos. Permite registrar a los usuarios que tendrán acceso al sistema, definir roles y establecer permisos para cada módulo.

c) Diagrama de caso de uso: Inspector Encargado

En este módulo presentaremos los movimientos, privilegios y restricciones de acceso que tiene el Inspector encargado y todas las relaciones que tiene con los otros módulos.

Figura 15.

Diagrama de caso de uso Inspector Encargado



Tabla 6.

Caso de uso Inspector Encargado

Caso de uso: Inspector Encargado	
Actores:	Inspector Encargado
Tipo:	Primario esencial
Descripción:	<p>El Inspector Encargado Principal tiene las siguientes opciones de modulo como ser:</p> <p>Módulo de Solicitudes: Este módulo agrupa tres submódulos principales: Hoja de Ruta, Minuta de Comunicación y P.I.E. A continuación, se describen sus funciones:</p> <ul style="list-style-type: none"> • Hoja de Ruta: Permite registrar todas las solicitudes provenientes de las Unidades Educativas. • Minuta de Comunicación y P.I.E.: Ambos submódulos tienen funciones similares y están diseñados para recibir solicitudes y demandas relacionadas con incumplimientos. <p>Cada submódulo permite registrar, editar, eliminar y visualizar registros.</p> <hr/> <p>Módulo de Mantenimiento: Este módulo incluye cuatro submódulos: Proceso del Trabajo, Inspecciones, Requerimientos de Material, Cronogramas e Intervenciones. A continuación, se detallan sus funciones:</p> <ul style="list-style-type: none"> • Inspecciones: Permite registrar las visitas realizadas a las Unidades Educativas, documentando los problemas encontrados en cada visita. • Requerimientos de Material: Este submódulo registra la solicitud de materiales necesarios según los problemas identificados previamente durante las inspecciones. • Cronogramas: Administra la asignación de fechas para que las cuadrillas realicen el trabajo de mantenimiento programado. • Intervenciones: Registra los trabajos realizados y concluidos en las Unidades Educativas. <p>Cada submódulo permite registrar, editar, eliminar y visualizar registros.</p>

Módulo de Recursos: Este módulo incluye el submódulo de Almacenes, que permite registrar el material entregado a las Unidades Educativas, incluyendo la cantidad y detalles. El submódulo permite registrar, editar y eliminar registros de material.

Módulo de Trabajo de Cuadrillas: Este módulo incluye la opción de Subir Imágenes de Trabajos, donde se registran las cuadrillas que acudieron a realizar trabajos en la Unidad Educativa. Permite registrar detalles como la descripción del trabajo realizado y adjuntar fotografías, las cuales se vinculan con el módulo de Intervenciones. Cada registro puede ser creado, editado, eliminado y visualizado para revisar los detalles del trabajo efectuado.

Módulo de Personal y Cuadrillas: Este módulo incluye dos secciones principales:

- **Personal:** Almacena la información del personal encargado de realizar inspecciones, trabajos, y la gestión de almacenes.
- **Cuadrillas:** Permite asignar al personal existente, especialmente a aquellos en roles de obreros, a una cuadrilla según su cargo.

Ambas secciones permiten registrar, editar y eliminar registros de manera eficiente.

Módulo de Unidades Educativas: Este módulo contiene dos secciones desplegables:

- **Unidad Educativa:** Permite registrar las Unidades Educativas, incluyendo los datos y detalles importantes de cada una.
- **Junta Escolar:** Registra las juntas escolares de cada Unidad Educativa, con las cuales se coordina para llevar a cabo los trabajos.

Ambas secciones permiten registrar, editar y eliminar registros.

Módulo de Personal y Cuadrillas: Este módulo incluye dos secciones principales:

- **Personal:** Almacena la información del personal encargado de realizar inspecciones, trabajos, y la gestión de almacenes.
- **Cuadrillas:** Permite asignar al personal existente, especialmente a aquellos en roles de obreros, a una cuadrilla según su cargo.

Ambas secciones permiten registrar, editar y eliminar registros de manera eficiente.

Módulo de Administración: Este módulo incluye las secciones de Usuarios, Roles y Permisos. Permite registrar a los usuarios que tendrán acceso al sistema, definir roles y establecer permisos para cada módulo.

d) Diagrama de caso de uso: Jefe de unidad.

Este módulo presentamos las acciones que puede realizar y los privilegios accesos que tiene el Encargado de Almacén.

Figura 16.

Caso de Uso de Encargado Almacén

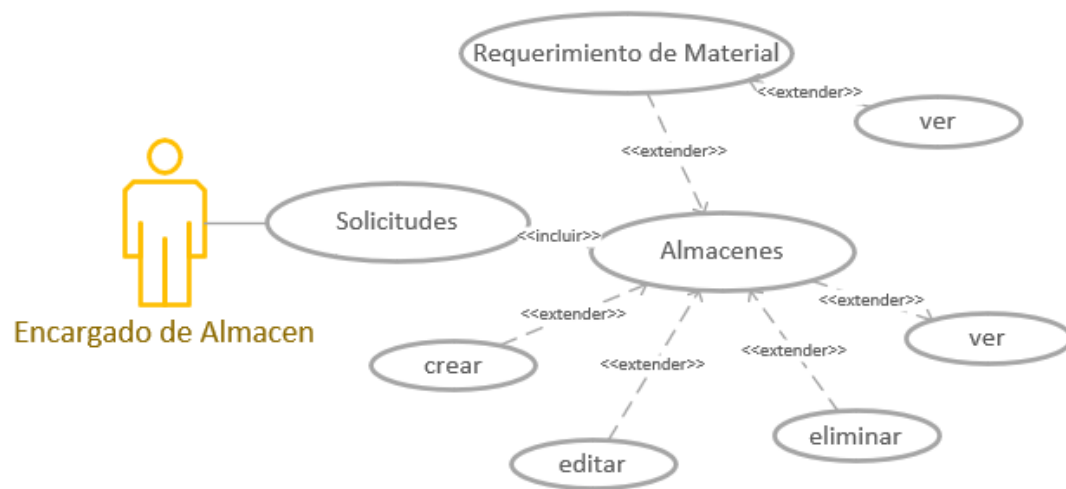


Tabla 7.

Caso de uso: Encargado de Almacén

Caso de uso: Encargado de Almacén	
Actores:	Encargado de Almacén
Tipo:	Primario esencial

Descripción: El Encargado de Almacén tiene las siguientes opciones de modulo como ser:
Módulo de Recursos: Este módulo incluye el submódulo de Almacenes, que permite registrar el material entregado a las Unidades Educativas, incluyendo la cantidad y detalles. El submódulo permite registrar, editar y eliminar registros de material.

e) Diagrama de Caso de Uso Personal de Cuadrilla

En este módulo presentaremos los movimientos, privilegios y restricciones de acceso que tiene el Personal de Cuadrillas.

Figura 17.

Diagrama de Caso de Uso Personal de Cuadrillas

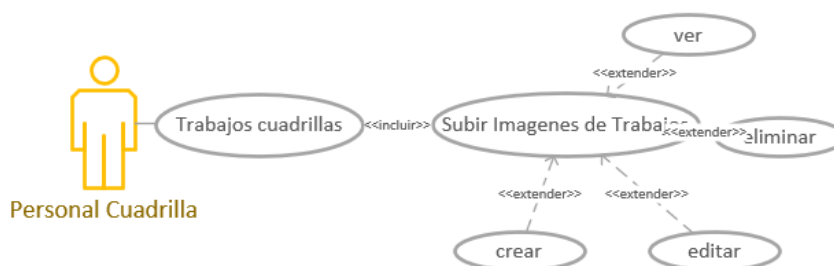


Tabla 8.

Tabla Caso de uso Personal de Cuadrilla

Caso de uso: Personal Carilla	
Actores:	Personal Cuadrilla
Tipo:	Primario esencial
Descripción:	El Personal de Cuadrilla tiene las siguientes opciones de modulo como ser:

Módulo de Trabajo de Cuadrillas: Este módulo incluye la opción de Subir Imágenes de Trabajos, donde se registran las cuadrillas que acudieron a realizar trabajos en la Unidad Educativa. Permite registrar detalles como la descripción del trabajo realizado y adjuntar fotografías, las cuales se vinculan con el módulo de Intervenciones. Cada registro puede ser creado, editado, eliminado y visualizado para revisar los detalles del trabajo efectuado.

3.4.2 Modelo de contenido

El modelo de contenido para aplicaciones web en UWE utiliza modelos estándares de UML para representar estructuras de clases, asociaciones y paquetes. El diagrama de contenido tiene como propósito mostrar las relaciones entre las entidades y la estructura de los datos alojados en el sistema. Este modelo de contenido contiene la información relevante almacenada en el sistema, así como su estructura y las relaciones entre los elementos.

El diseño conceptual se basa en el modelo de análisis y abarca los objetos involucrados en las actividades típicas que los usuarios realizan con la aplicación. El objetivo del modelo de contenido es proporcionar una especificación visual de la información relevante para el dominio del sistema web, centrada principalmente en el contenido de la aplicación.

Figura 18.

Modelo de Contenido Base de Datos

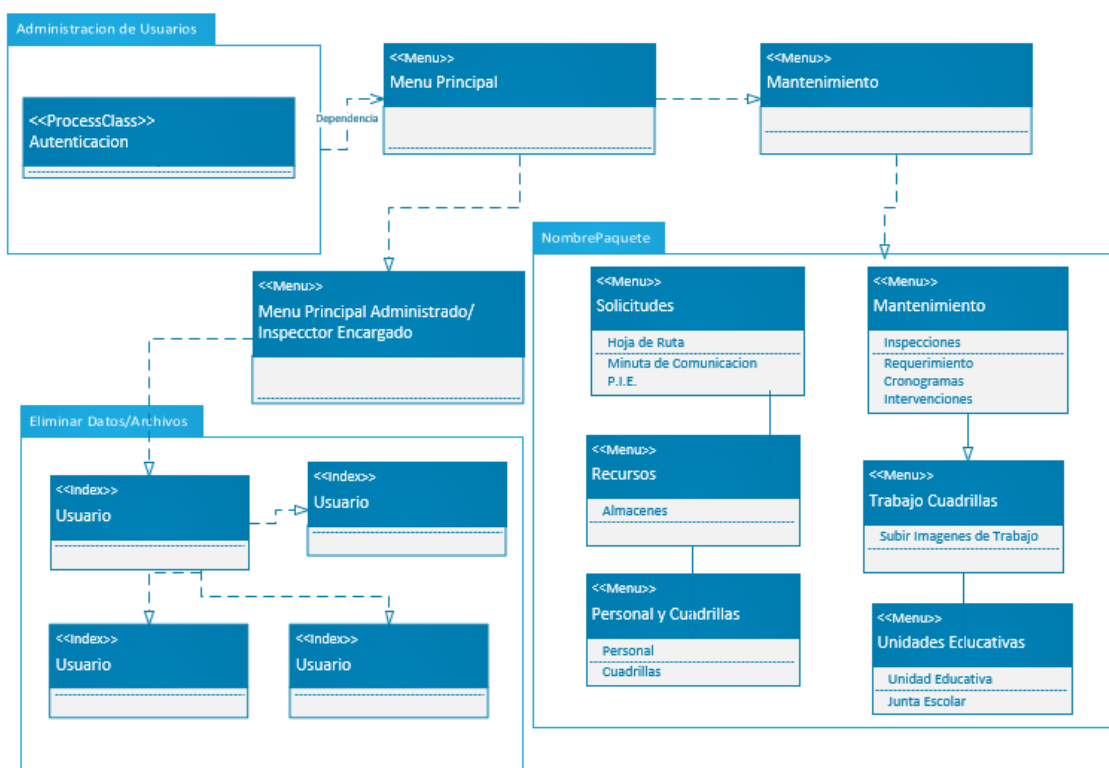


3.4.3. Modelo de Navegación

En la fase de diseño de navegación de la metodología UWE, se emplean diagramas específicos para visualizar y estructurar el dominio de la aplicación web. Los diagramas principales incluyen el diagrama de estructura de navegación, que muestra la relación entre páginas, y el diagrama de clases navegacionales, que introduce las "clases navegacionales" como nodos de acceso en la navegación. También se incluyen elementos de acceso, como enlaces y botones, que facilitan el flujo entre las distintas partes de la aplicación.

Figura 19.

Modelo de Navegación



3.4.4. Modelo de Presentación

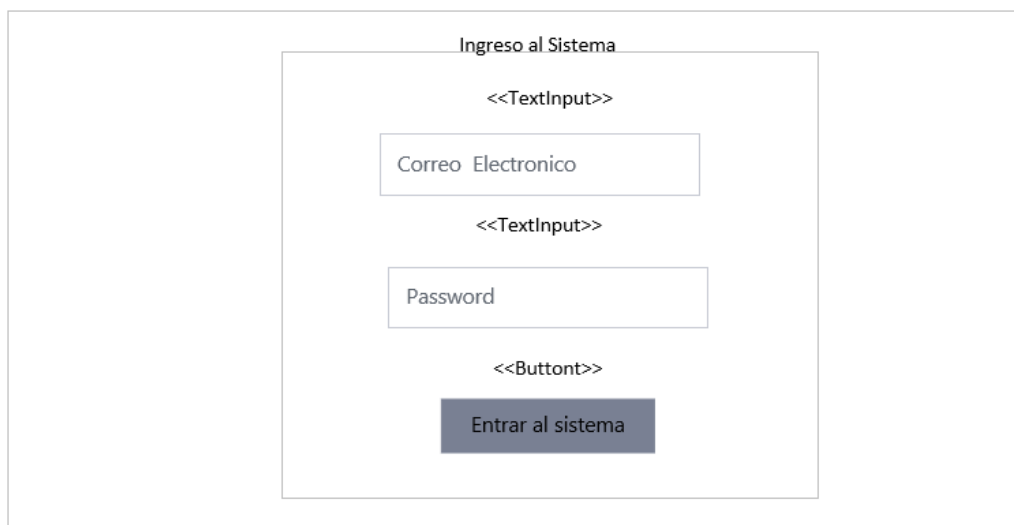
El modelo de presentación UWE (UML-based Web Engineering), busca proporcionar una representación abstracta de la interfaz de usuario final y definir la interacción de las clases navegables con el usuario. Este modelo se basa en el modelo de navegación.

3.4.4.1 Modelo de Autenticación de Usuario

El modelo de presentación nos muestra cómo se verá el Login o Ingreso al sistema.

Figura 20.

Modelo de Navegación Ingreso al sistema

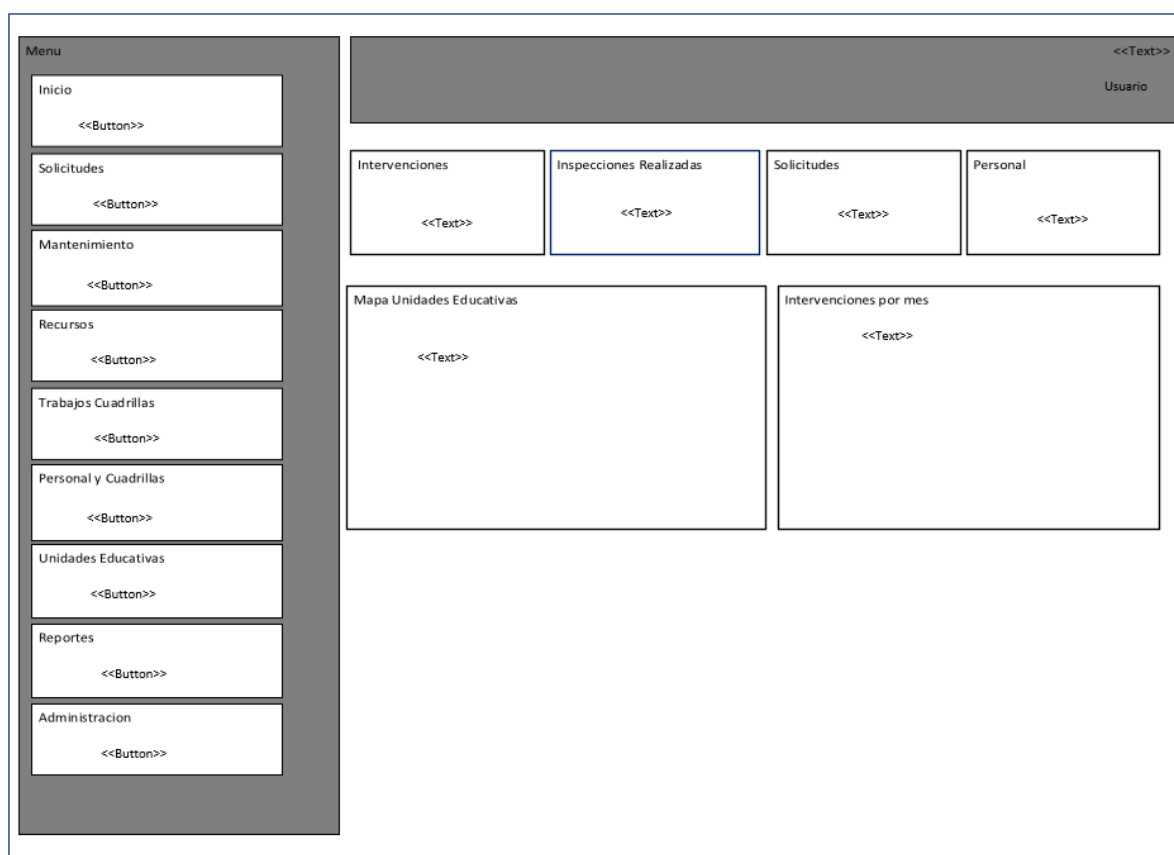


3.4.4.2 Modelo de Presentación de la Página Principal

El modelo de presentación nos va mostrando como se verá el sistema, en la página principal se tiene un esquema de todos los menús en general.

Figura 21.

Modelo de Presentación de Pagina Maestra

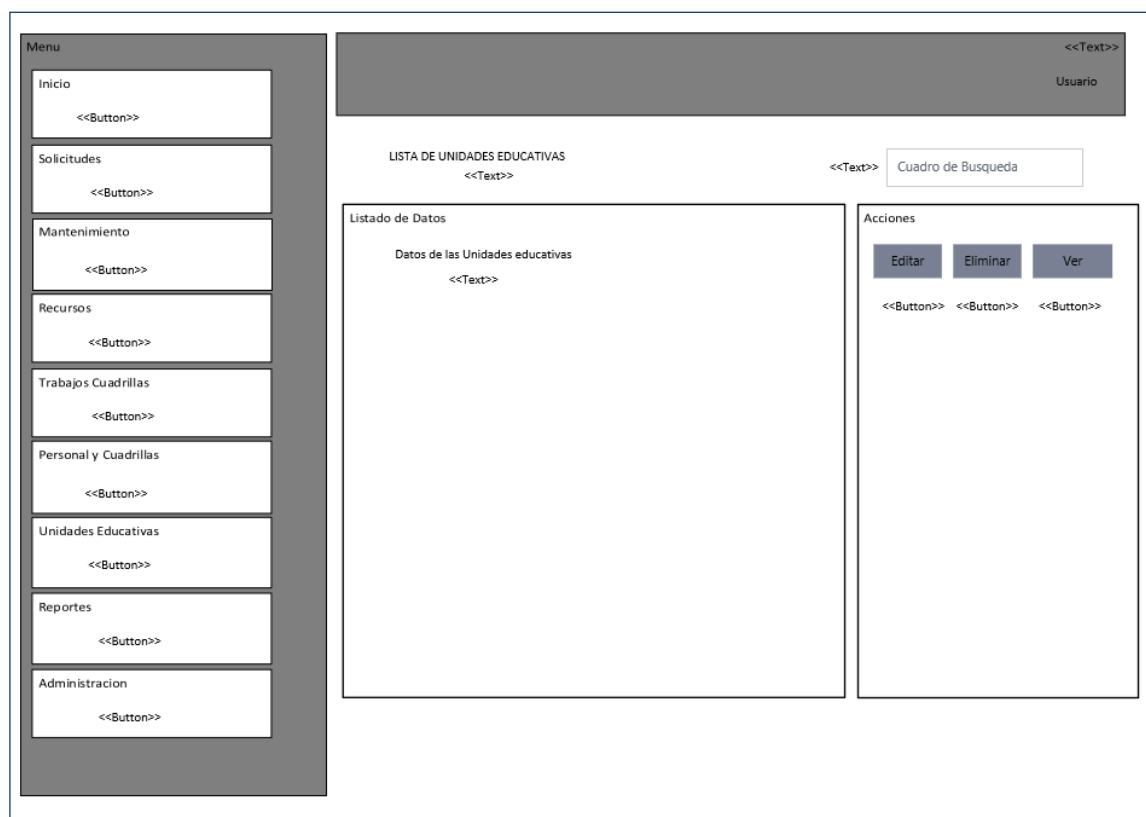


3.4.4.3 Modelo de Presentación Listado de Registros y Acciones

El Modelo nos muestra cómo funcionan los menús de una ejemplificación general de muestra de todas las operaciones que realiza el sistema.

Figura 22.

Modelo de Listado de Registros y Acciones



3.5. IMPLEMENTACION DEL SISTEMA

En esta fase de implementación se presenta el desarrollo de las interfaces del sistema y sus elementos, construidos a partir del diseño del Modelo de Presentación de UWE. A continuación, se ilustran estos elementos en las figuras siguientes:

Figura 23.

Interfaz de Inicio de Sesión

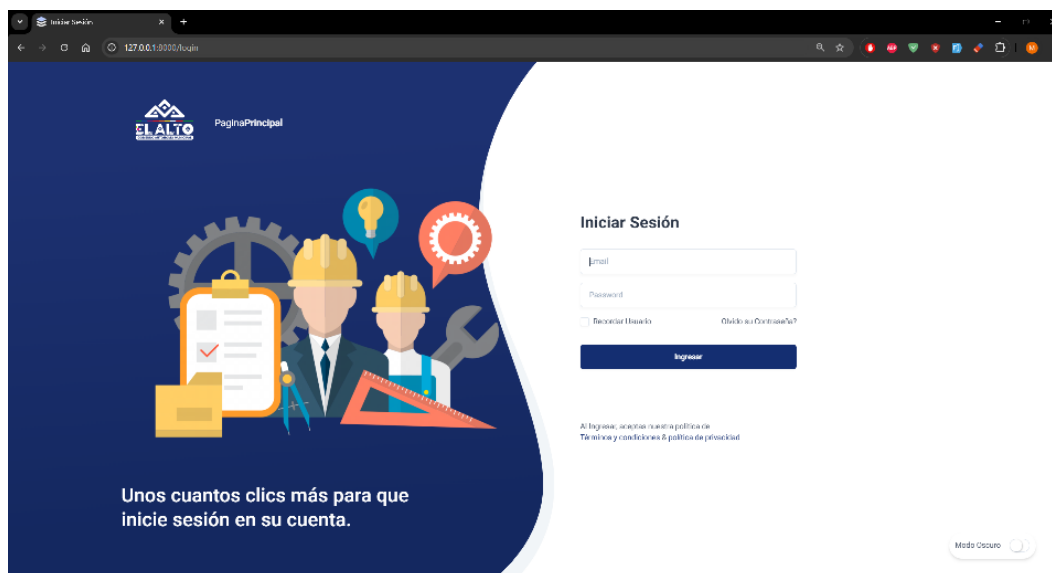


Figura 24.

Inicio Principal

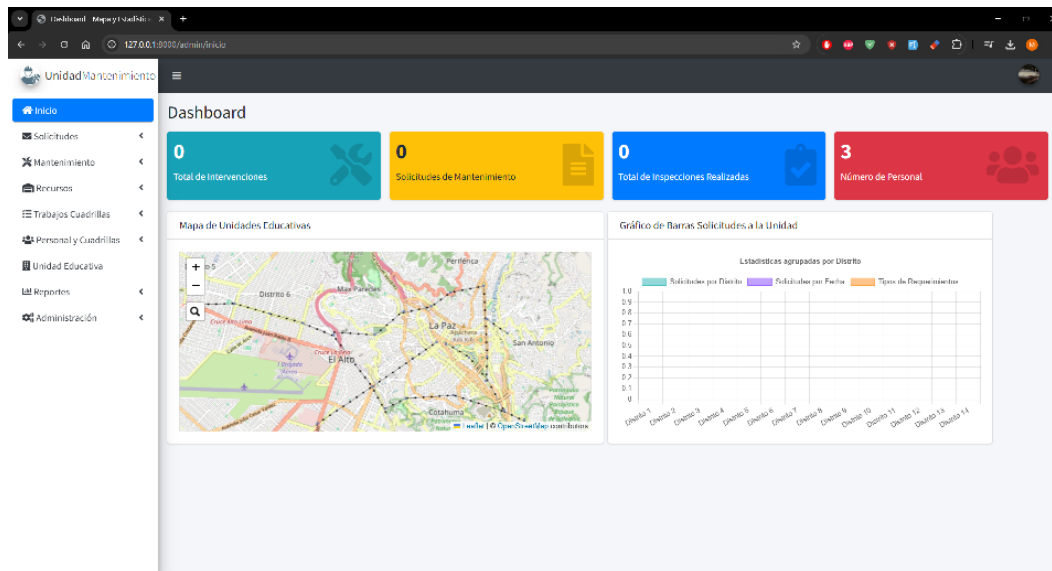


Figura 25.

Hoja de Ruta

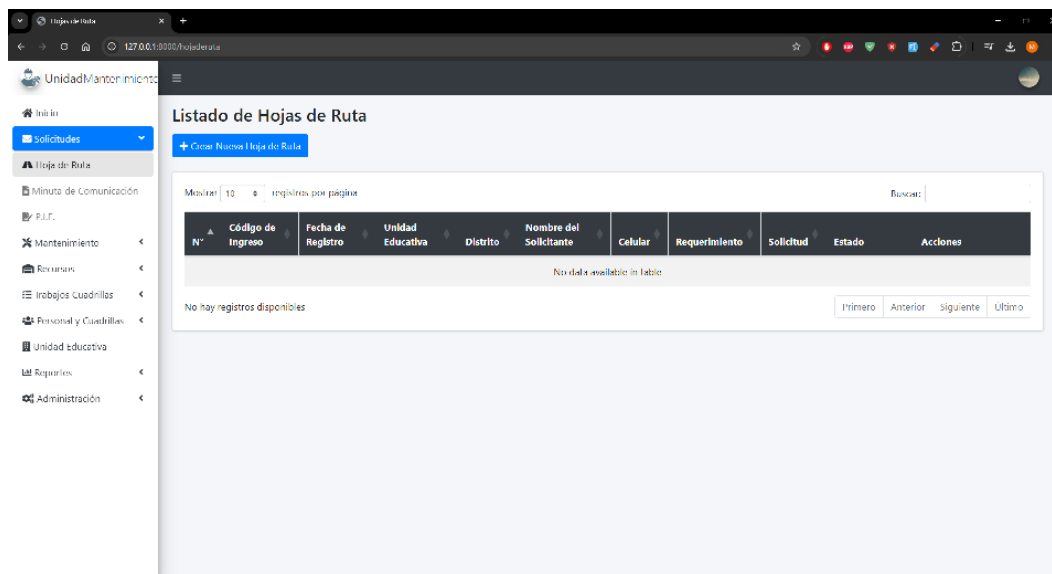


Figura 26.

Inspecciones

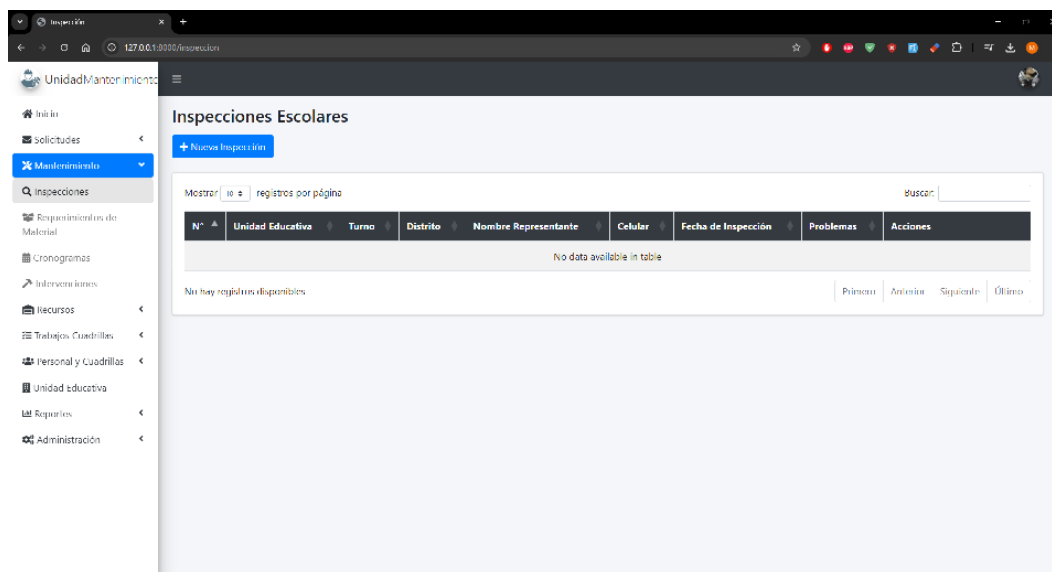


Figura 27.

Requerimientos de Material

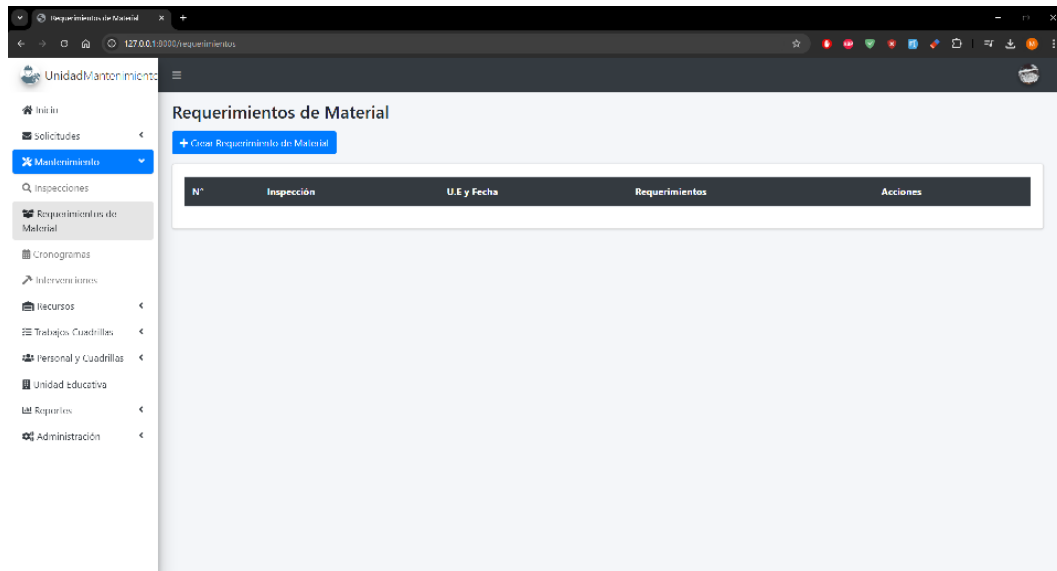


Figura 28.

Cronogramas

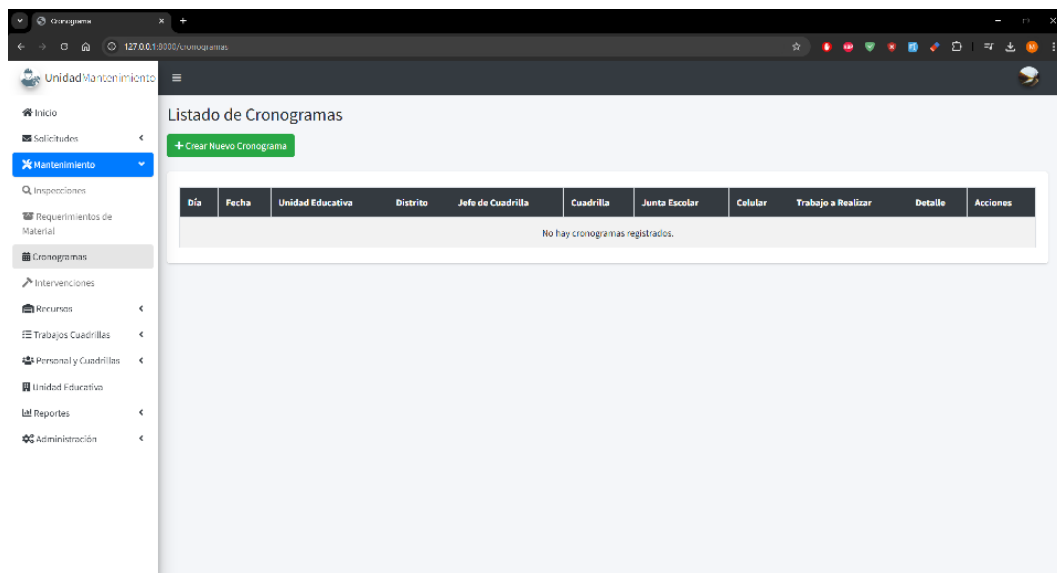
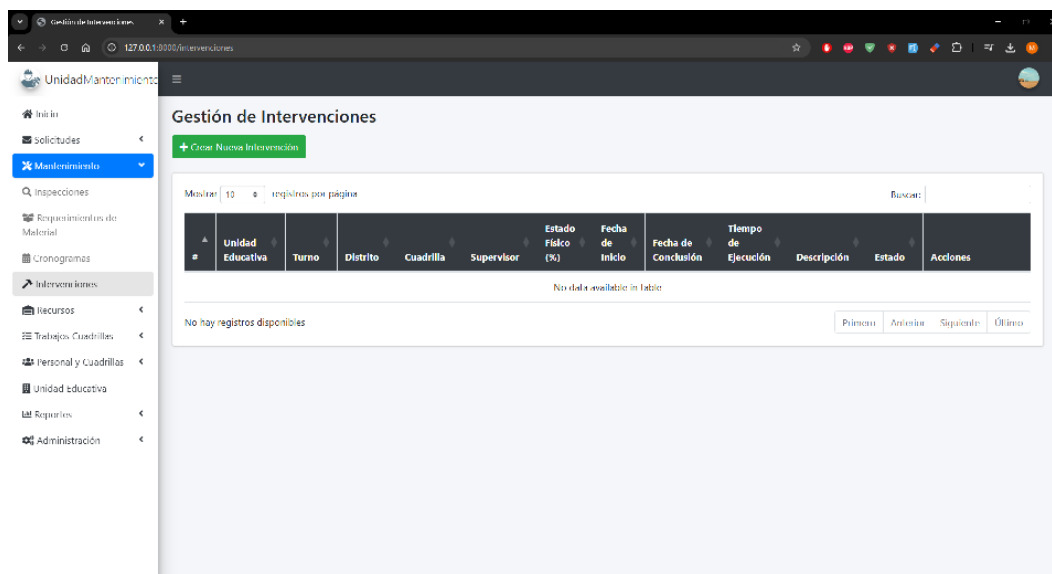


Figura 29.

Intervenciones

The screenshot shows a web browser window displaying a management interface for interventions. The browser's address bar shows the URL `127.0.0.1:3000/intervenciones`. The page title is "Gestión de Intervenciones". On the left, a sidebar menu lists various system components, with "Intervenciones" currently selected. The main content area features a header with a green button labeled "+ Crear Nueva Intervención". Below this is a table with columns for "Unidad Educativa", "Turno", "Distrito", "Cuadrilla", "Supervisor", "Estado Físico (%)", "Fecha de Inicio", "Fecha de Conclusión", "Tiempo de Ejecución", "Descripción", "Estado", and "Acciones". The table is currently empty, displaying the message "No data available in table" and "No hay registros disponibles". Navigation controls at the bottom of the table include "Primero", "Anterior", "Siguiente", and "Último".

Figura 30.

Almacén

Unidad/Mantenimiento

Inicio

Solicitudes

Mantenimiento

Recursos

Almacenes

Trabajos Cuadrillas

Personal y Cuadrillas

Unidad educativa

Repuestos

Administración

Almacén de Materiales

+ Agregar Nuevo Registro

Mostrar: 10 registros por página

Buscar:

#	Unidad Educativa	Turno	Distrito	Descripción del Requerimiento	Material Entregado Por	Material Recibido Por	Fecha de Entrega	Descripción del Material Entregado	Acciones
No data available in table									

No hay registros disponibles

Primero Anterior siguiente Ultimo

Figura 31.

Trabajos Cuadrillas

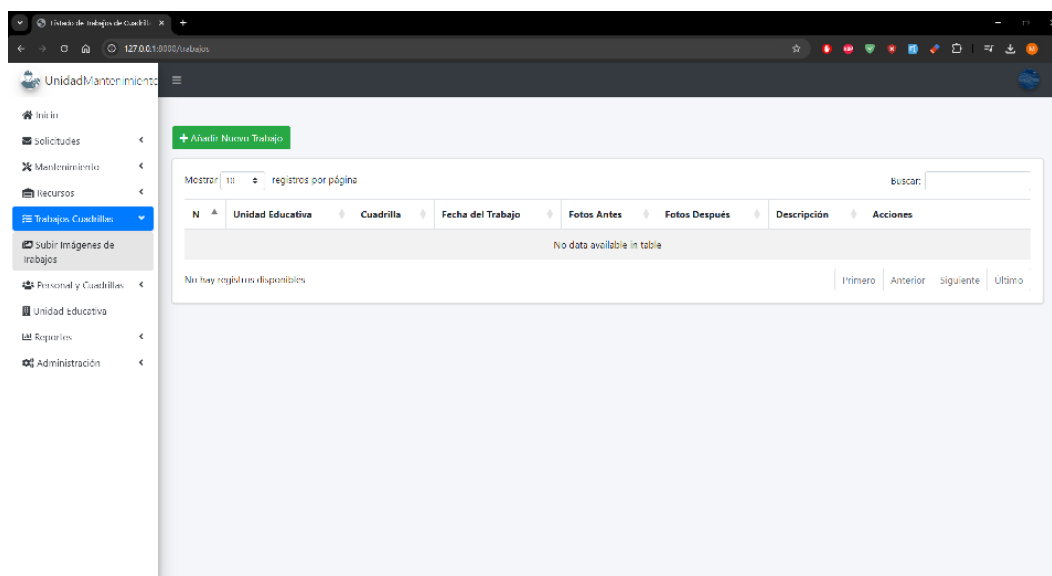


Figura 32.

Listado de Personal de la Unidad

Unidad/Mantenimiento

Inicio

Solicitudes

Mantenimiento

Recursos

Trabajos Cuadrillas

Personal y Cuadrillas

Personal

Cuadrillas

Unidad Educativa

Reportes

Administración

Personal

+ Agregar Personal

Mostrar 10 registros por página

Buscar:

#	Nombre	Apellido Paterno	Apellido Materno	Cargo	Carnet	Celular	Dirección	Acciones
1	Bryan	Quipe	Albino	Inspector de obras	1291567	12515578	Calle falso 123	editar eliminar
2	Jose	Ramirez	Huaytuo	electricista	2136789	11222991	Calle Verdadero 789	editar eliminar
3	Marco Antonio	Yujra	Mendoza	Inspector de obras	2345678	87651921	Av. Real 456	editar eliminar

Mostrando página 1 de 1

Primera Anterior 1 Siguiente Última

Figura 33.

Cuadrillas

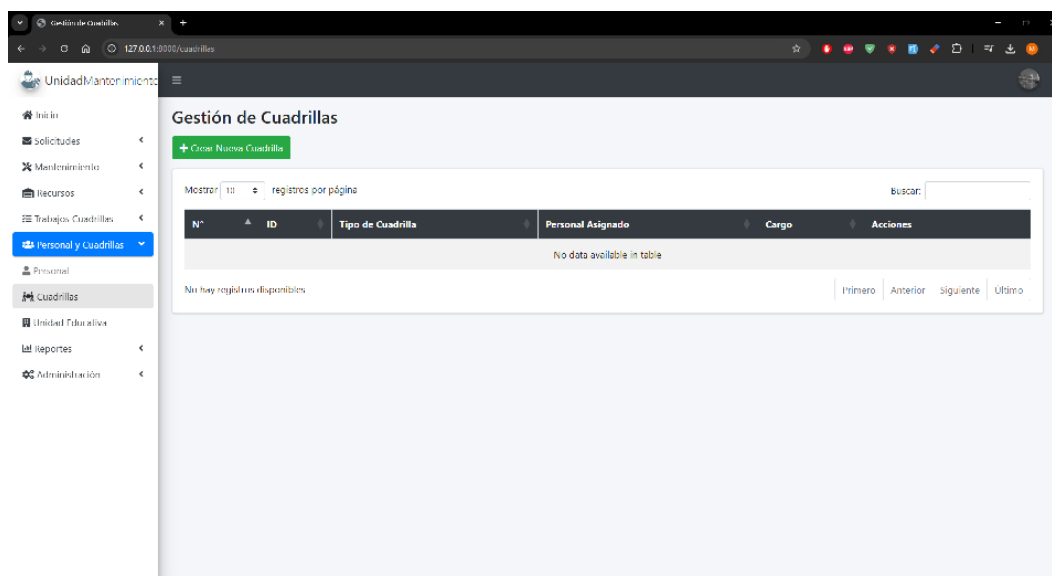


Figura 34.

Unidades Educativas

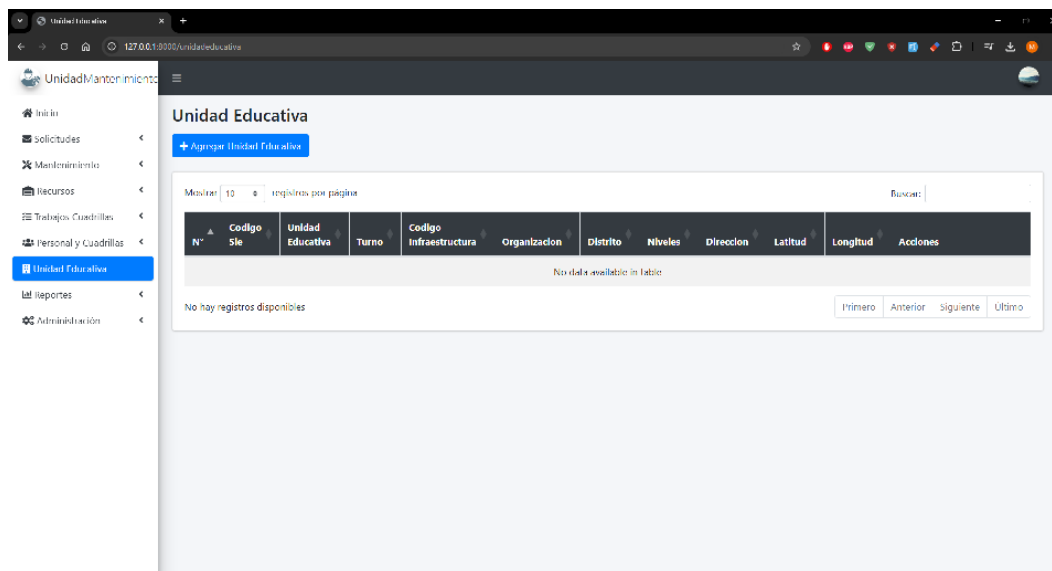


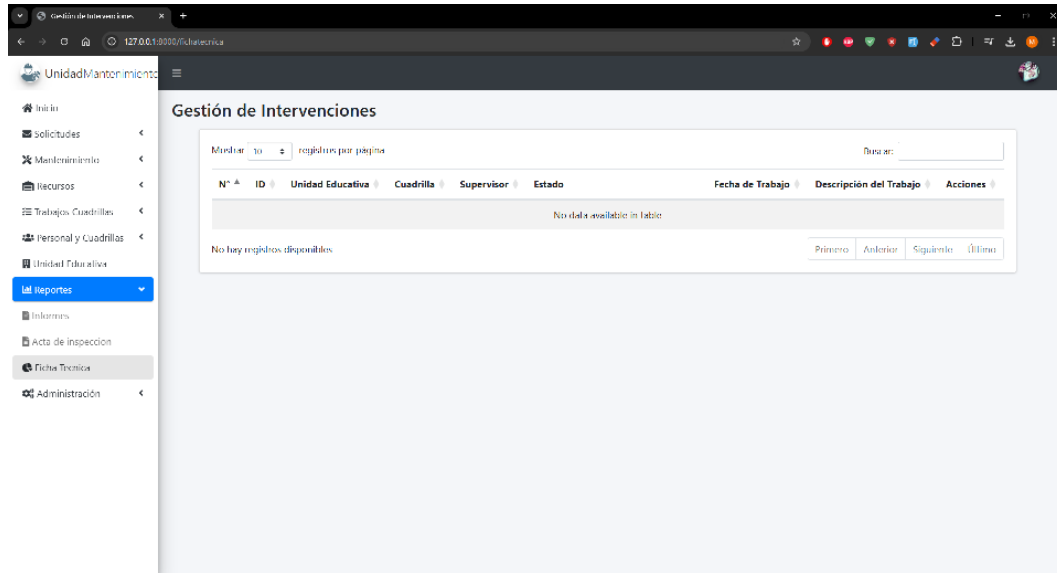
Figura 35.

Acta de Inspección

The screenshot shows a web browser window with the URL `127.0.0.1:3030/Reportes`. The page title is "Acta de Inspección". On the left, there is a sidebar menu with the following items: Inicio, Solicitudes, Mantenimiento, Recursos, Trabajos Cuadrillas, Personal y Cuadrillas, Unidad Educativa, Reportes (highlighted), Informes, Acta de Inspección (highlighted), Ficha Técnica, and Administración. The main content area features a "Crear reporte" button and a table with the following columns: N°, Unidad Educativa, Generado por, Fecha de Inspección, Fecha de Creación, and Acciones. The table is currently empty, displaying the message "No data available in table". Below the table, it states "No hay registros disponibles" and includes navigation buttons: "Primero", "Anterior", "siguiente", and "Último".

Figura 36.

Ficha Técnica



The screenshot shows a web browser window displaying a management interface. The browser's address bar shows the URL `127.0.0.1:8080/FichaTecnica`. The page title is "Gestión de Intervenciones". On the left, a sidebar menu lists various options: Inicio, Solicitudes, Mantenimiento, Recursos, Trabajos Cuadrillas, Personal y Cuadrillas, Unidad Educativa, Reportes (highlighted in blue), Informes, Acta de Inspección, Ficha Técnica (highlighted in grey), and Administración. The main content area features a search bar with "Mostrar" and "registros por página" dropdowns, and a "Buscar:" input field. Below this is a table header with columns: N° A, ID, Unidad Educativa, Cuadrilla, Supervisor, Estado, Fecha de Trabajo, Descripción del Trabajo, and Acciones. The table body is empty, displaying the message "No data available in table". At the bottom of the table area, it says "No hay registros disponibles" and includes navigation buttons: Primero, Anterior, Siguiente, and Último.

CAPITULO IV

INGENIERIA
DE SISTEMAS
UNIVERSIDAD PUBLICA DE EL ALTO



CAPITULO IV

4. INTRODUCCION

A continuación, este capítulo presenta las métricas de calidad, costo y seguridad que respaldan el desarrollo del proyecto de grado en sus distintas etapas.

4.1. METRICAS DE CALIDAD ISO 25000

Este capítulo evalúa la calidad del sistema, un aspecto fundamental en el desarrollo de software, y describe su nivel de seguridad. Se aplicaron métricas de calidad para asegurar un desarrollo adecuado del proyecto, midiendo los parámetros que cumple el sistema. Esto permite especificar de manera organizada las características y atributos del sistema.

4.1.1. Funcionalidad

La funcionalidad de un programa se refiere a su capacidad para cumplir con las especificaciones y objetivos definidos por el usuario. Este atributo esencial del sistema no puede evaluarse de manera directa, sino a través del análisis de sus características y capacidades. Evaluar la funcionalidad implica comprobar que el sistema proporciona las funciones necesarias para satisfacer tanto las necesidades explícitas como las implícitas del usuario, desempeñando de manera adecuada las tareas requeridas bajo condiciones específicas.

Punto de función

El desarrollo de esta técnica de estimación de software, toma en cuenta 5 características:

- **Número de entradas de usuario:** la cantidad de ocasiones en que el usuario ingresa datos en el sistema.

Tabla 9.

Número de entradas de Usuario

	Entradas de Usuario	Numero
1	Administración de usuarios	1
2	Administración de técnicos	1
3	Administración de usuarios	28
	Total	30

- **Número de salidas de usuario:** la cantidad de resultados que el sistema proporciona al usuario, como mensajes, notificaciones, informes, alertas, entre otros.

Tabla 10.

Salidas de Usuario

	Salidas de Usuario	Numero
1	Administración de usuarios	3
2	Administración de técnicos	3
3	Administración de usuarios	20
	Total	26

- **Número de peticiones de usuario:** cantidad de entradas interactivas que generan una respuesta inmediata del software en forma de salidas interactivas.

Tabla 11.

Número de Peticiones del Usuario

	Peticiones del Usuario	Numero
1	Administración de usuarios	3
2	Administración de técnicos	3
3	Administración de usuarios	6
	Total	12

- **Número de archivos:** cantidad de archivos que se consideran, incluyendo grupos lógicos de datos o archivos independientes.
- **Número de interfaces externas:** total de interfaces legibles por la máquina.

Tabla 12.

Parámetros de medición de punto de función

Parámetros de medición	Cuenta
1 Numero de entrada de Usuarios	30
2 Numero de salida de Usuarios	26
3 Numero de Peticiones del usuario	12
4 Numero de Archivos	17
5 Numero de Interfaces externas	0

Una vez realizado el conteo de los parámetros de función en la tabla, se procede al cálculo de los puntos de función sin ajustar, como se muestra en la siguiente tabla.

Tabla 13.

Punto de función sin ajustar

Parámetros de medición	Cuenta	Factor	Total
1 Numero de entrada de Usuarios	30	6	180
2 Numero de salida de Usuarios	26	7	182
3 Numero de Peticiones del usuario	12	6	72
4 Numero de Archivos	17	15	255
5 Numero de Interfaces externas	0	10	0
Total			689

Nota. Punto de función sin ajustar.

La tabla muestra el total de puntos de función sin ajustar. Para obtener los valores

ajustados, se aplican los factores de complejidad que se listan a continuación.

Tabla 14.

Factores de Complejidad

Importancia	20 %	40 %	60 %	80 %	100 %		
Escala	No Influencia	Incidental	Moderado	Medio	Significativo	Esencial	FI
Factor	0	1	2	3	4	5	
¿Requiere el sistema copias de seguridad y recuperación fiable?				X			3
¿Se requiere comunicación de datos?				X			3
¿Existe funciones de procesos distribuidos?				X			3
¿Es critico el rendimiento?		X					1
¿El sistema web será ejecutado en S.O. Actual?						X	5
¿Se requiere una entrada interactiva para el sistema?					X		4
¿Se requiere que el sistema tenga entradas a datos con múltiples ventanas?			X				2
¿Se actualiza los archivos de forma interactiva?					X		4
¿Son complejas las entradas, salidas, los archivos o las peticiones?					X		4
¿Es complejo el procesamiento interno del sistema?					X		4
¿Se ha diseñado el código para ser reutilizado?					X		4
¿Están incluidas en el diseño la conversión y la instalación?				X			3

Importancia	20 %	40 %	60 %	80 %	100 %		
Escala	No Influencia	Incidental	Moderado	Medio	Significativo	Esencial	Fi
Factor	0	1	2	3	4	5	
¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?				X			3
¿Se ha diseñado el sistema para facilitar al usuario el trabajo y ayudarlos a encontrar la información?						X	5
TOTAL							4 8

Para el cálculo del punto de función ajustado se utiliza la siguiente fórmula:

$$PF = \text{Cuenta Total} * (0,65 + 0.01 * \sum fi) \quad (1). \text{ Ecuación Punto de Función}$$

Donde: Cuenta total: es la suma del producto entre el factor de ponderación y los valores de los parámetros. $\sum F i$: representa la sumatoria de los valores de ajuste de complejidad.

$$PF = 689 * (0,65 + 0.01 * 48)$$

$$PF = 689 * 1.13$$

$$PF = 778.57$$

Considerando el máximo ajuste de la complejidad $\sum F i = 70$ calculamos al 100%

el nivel de confianza de la siguiente manera:

$$PF_{\max} = \text{CuentaTotal} * (0,65 * 0,01 * \sum fi) \quad (2). \text{ Ecuación Punto de Función M\u00e1ximo}$$

$$PF_{\max} = \text{CuentaTotal} * (0,65 * 0,01 * \sum fi)$$

$$PF_{\max} = 689 * (0,65 * 0,01 * 70)$$

$$PF_{\max} = 689 * 1,35$$

$$PF_{\max} = 930,15$$

La relaci\u00f3n obtenida entre ambos es la funcionalidad:

$$\text{Funcionalidad} = \frac{PF}{PF_{\max}} \quad (3). \text{ Ecuaci\u00f3n de Funcionalidad}$$

$$\text{Funcionalidad} = \frac{778,57}{930,15}$$

$$\text{Funcionalidad} = 0,837 * 100 = 83,7\%$$

4.1.2 Usabilidad

La usabilidad se refiere al esfuerzo necesario que el usuario debe invertir para utilizar el sistema, basado en su comprensi\u00f3n y la estructura l\u00f3gica del mismo. Esta comprensi\u00f3n del sistema por parte del usuario se eval\u00faa a trav\u00e9s de los siguientes pasos:

Formula:

$$FU = \left[\frac{\sum x_i}{n} * 100 \right] \quad (4). \text{ Ecuaci\u00f3n de Usabilidad}$$

- x_i = Valores de evaluación de cada pregunta
- n = Numero de preguntas *d e p r e g u n t a s* .

Se toma en cuenta la siguiente tabla

Tabla 15.

Parámetros de medición de usabilidad

Escala	Valor
Muy Bueno	5
Bueno	4
Regular	3
Malo	2
Pésimo	1

Tabla 16.

Usabilidad del sistema

Preguntas	Respuestas	Preguntas	Preguntas
¿Puede Utilizar con facilidad el sistema?	9	1	0.9
¿Es crítico el rendimiento?	7	3	0.7
¿Son complicados los procesos que realiza el sistema?	8	2	0.8
¿Es satisfactorio las repuestas que el sistema devuelve?	9	1	0.9
¿El sistema tiene interfaces entendibles?	9	1	0.9
¿El sistema reduce su tiempo de trabajo?	9	1	0.9
¿Es difícil aprender a manejar el sistema?	8	2	0.8
¿El sistema satisface las necesidades que usted requiere?	9	1	0.9
¿Utiliza el sistema con facilidad?	9	1	0.9
¿Considera usted que es una herramienta útil?	10	0	1

TOTAL

8.7

Con los resultados obtenidos del cuestionario obtenemos calculamos la usabilidad:

$$FU = \left[\frac{8.7}{10} * 100 \right]$$

$$FU = [0.87 * 100]$$

$$FU = 87\%$$

4.1.3. Eficiencia

La eficiencia mide si el sistema utiliza los recursos de manera óptima. Para calcular el grado de eficiencia, se utiliza la siguiente escala:

Tabla 17.

Escala de valores de eficiencia

Escala	Valor
Muy Bueno	5
Bueno	4
Regular	3
Malo	2
Pésimo	1

Para valorar la eficiencia del sistema se tiene la siguiente tabla:

Tabla 18.

Valoración de la eficiencia del sistema

Preguntas	Porcentaje
¿La distribución y estilo de la interfaz permite que un usuario introduzca con eficiencia las operaciones información?	5
¿Una secuencia de operaciones (o entrada de datos) puede realizarse con facilidad de movimientos?	4
¿Los datos de salida están presentados de modo que se entienden de inmediato?	5
¿Las operaciones jerárquicas están organizadas de manera que minimizan la navegación del usuario para hacer que alguna se ejecute?	4
¿Procesa y responde adecuadamente cuando realiza alguna consulta o búsqueda?	4
TOTAL	22

Nota. Valoración de la eficiencia del sistema.

Para calcular la eficiencia tenemos la siguiente formula:

$$E = \left[\frac{\sum x_i}{n} * \frac{100}{n} \right] \quad (5). \text{ Ecuación de Eficiencia}$$

Donde:

- $\sum x_i$ = Sumatoria de los valores de eficiencia
- n = Numero de preguntas

Remplazando se obtiene lo siguiente:

$$E = \left[\frac{22}{5} * \frac{100}{5} \right]$$

E=88%

Con esto concluimos que la eficiencia del sistema es del **88%**

4.1.4 Mantenibilidad

Para evaluar la mantenibilidad del sistema se emplea el Índice de Madurez de Software (IMS), que indica la estabilidad de un producto de software en función de los cambios que se presentan en cada versión.

Donde la formula es :

$$\text{IMS} = [\text{Mt} - (\text{Fc} + \text{Fa} + \text{Fe})] / \text{Mt}. \quad (6). \text{ Ecuación de Mantenibilidad}$$

Donde:

Mt: Numero de módulos total de la versiona actual.

Fc: Numero de módulos de la versión que se cambiaron.

Fa: Numero de módulos de la versión Actual que se añadieron.

Fe: Numero de módulos de la versión anterior que se eliminaron en la versión actual.

Por lo tanto, se aplica al sistema y da el siguiente resultado de índice de madures

$$\text{IMS} = [\text{Mt} - (\text{Fc} + \text{Fa} + \text{Fe})] / \text{Mt}$$

$$\text{IMS} = [20 - (0 + 1 + 0)] / 20$$

$$\text{IMS} = 0.95 * 100 = 95\%$$

Por tanto, el sistema empieza a estabilizarse en un 95%

4.1.5 Portabilidad

La portabilidad del sistema fue evaluada considerando la capacidad de trasladarlo a diferentes entornos y equipos con el mínimo esfuerzo. A continuación, se detallan los aspectos clave de esta característica en el sistema:

Automatización mediante Migraciones: El sistema utiliza migraciones para llevar la estructura de la base de datos a nuevos entornos, lo cual permite replicar la configuración de la base de datos de manera automatizada y sin errores manuales. Esto facilita la portabilidad y reduce el tiempo de configuración en nuevos equipos.

Configuración Simplificada: A través del archivo `.env`, el sistema permite gestionar variables de entorno como credenciales de base de datos y configuraciones específicas de cada servidor. Este enfoque centralizado facilita la adaptación del sistema a distintos entornos y mejora la flexibilidad en la configuración.

Compatibilidad de Dependencias: Aunque el sistema funcionó en varios equipos, se requiere asegurar que los entornos cumplan con versiones específicas de PHP y PostgreSQL. Esto implica un paso adicional para verificar la compatibilidad de las dependencias, pero permite un funcionamiento óptimo en sistemas que cumplan estos requisitos.

Documentación de Comandos Necesarios: Para garantizar la instalación exitosa en otros entornos, se documentaron los comandos esenciales, como `php artisan migrate` y `composer install`, necesarios para replicar el sistema en nuevos equipos. Esta guía reduce la necesidad de configuraciones manuales y apoya la portabilidad del sistema.

- **Automatización mediante Migraciones:** 90%
- **Compatibilidad de Entorno:** 85%

- Dependencias Documentadas: 80%
- Guía de Comandos Necesarios: 95%

Calcular el Promedio de Portabilidad:

$$\text{Portabilidad} = \frac{90+85+80+95}{4} \quad (7). \text{ Portabilidad}$$

$$\text{Portabilidad} = \frac{350}{4}$$

$$\text{Portabilidad} = 87.5 \%$$

Resultados: El sistema demostró un alto nivel de portabilidad de un 87.5%, especialmente gracias al uso de migraciones y archivos de configuración que facilitan el traslado entre equipos y entornos. La portabilidad se mejora aún más con documentación específica sobre dependencias y comandos, lo cual asegura una instalación más uniforme y eficiente.

4.1.5. Confiabilidad

Es la probabilidad de que un programa opere sin fallos en un entorno específico durante un tiempo determinado. Se observa el funcionamiento hasta que ocurra un fallo en el instante t , y se calcula la probabilidad de fallo mediante una variable aleatoria continua T . En una función exponencial, la relación es la siguiente:

Probabilidad de hallar una falla: $P(T \leq t) = F(t)$

Probabilidad de no hallar una falla: $P(T > t) = 1 - F(t)$

$\text{Conf}(t) = FC * e^{-\lambda t}$

Donde:

$FC = 0.81$; funcionalidad del sistema.

$\lambda = 1$; tasa de fallos en 8 ejecuciones dentro de un mes.

Hallamos la confiabilidad del sistema:

$$F(t) = FC * e^{-\lambda t}$$

$$F(t) = 0.81 * e^{-1 * 12}$$

$$F(t) = 0.13$$

portabilidad de hallar una falla es de un 13% durante los próximos 12 meses.

$$P(T > t) = 1 - F(t)$$

$$P(T > t) = 1 - 0.13$$

$$P(T > t) = 0.87$$

4.1.7 Resultados

Tabla 19.

Resultados de evaluación de calidad

Características	Resultado
Funcionabilidad	83.7%
Usabilidad	87%
Eficiencia	88%
Mantenibilidad	95%
Portabilidad	87.5%
Confiabilidad	87%
Evaluación Total	88.03%

Nota: Muestra de resultados obtenidos de calidad

4.2 ANALISIS DE COSTOS

Para realizar el presente proyecto es necesario planificar y estimar los costos durante y hasta la finalización del mismo, es preciso estimar el costo total del sistema se tomará en cuenta los siguientes costos:

- **Orgánico:** Proyectos relativamente sencillos, con menos de 5000 líneas de código, que implican procesamiento de datos y uso de bases de datos,

centrados en transacciones y recuperación de información.

- **Semiacoplado:** Proyectos de complejidad y tamaño intermedios, con experiencia variable en su desarrollo y restricciones de nivel medio.
- **Empotrado:** Proyectos de alta complejidad, con poca experiencia previa y en un entorno de gran innovación técnica. Para estimar los costos de este tipo de proyecto, se utilizará el submodelo post-arquitectura.

Para calcular el esfuerzo, necesitamos determinar la variable KLDC (kilo-líneas de código). En este proyecto, se implementarán 27,073 líneas de código. Aplicando las conversiones correspondientes, se obtiene la siguiente tabla:

Tabla 20.

Tabla de Linas de Código

LENGUAJE	FILES	CODE	COMENT	BLANK	TOTAL
PHP	174	5,103	742	1,022	6,867
JavaScript	11	1,380	247	310	1,937
HTML	23	1,395	0	423	1,818
CSS	8	554	38	21	613
Markdown	1	126	0	28	154
JSON	2	41	0	1	42
Total	219	8,599	1,027	1,805	11,431

Nota. Presenta la cantidad de líneas de código fuente del sistema web

LCD=8599

$$KLCD = \frac{LCD}{1000}$$

$$KLCD = \frac{8599}{1000}$$

$$KLCD = 8.599$$

El proyecto de 8,599 líneas de código se encuentra en la categoría Semiacoplado. Este modelo es adecuado para proyectos de complejidad y tamaño intermedios.

Tabla 21.

Coefficientes de a,b,c,d de COCOMO II

Proyecto Software	a	b	c	d
Orgánico	3.2	1.05	2.5	0.38
Semi acoplado	3.0	1.12	2.5	0.35
Empotrado	2.8	1.20	2.5	0.32

Nota. Coeficientes de a,b,c,d de COCOMO II

Dado que el proyecto no supera las 50,000 líneas de código, se utilizarán los coeficientes para proyectos semiacoplados. A continuación, se presentan las ecuaciones correspondientes al coeficiente seleccionado según la cantidad de líneas de código, para calcular el costo del software:

Tabla 22.

Ecuaciones del Modelo COCOMO II

Variable	Ecuación	Tipo/Unidad
Esfuerzo requerido por el proyecto	E	Personas/Mes
Tiempo requerido por el proyecto	$T = c * (E)^d$	Meses
Número de Personas requeridos para el proyecto	$P = ET$	Personas
Costo Total	$CT = NP * T$	\$us

Tabla 23.

Conductores para COCOMO II

Conductores de Coste	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extr. Alto
Fiabilidad requerida del software	0.75	0.88	1.00	1.15	1.40	-
Tamaño de la base de datos	-	0.94	1.00	1.08	1.16	-
Complejidad del producto	0.70	0.84	1.00	1.15	1.30	1.65
Restricciones del tiempo de ejecución	-	-	1.00	1.11	1.30	1.66

Restricciones del almacenamiento principal	-	-	1.00	1.06	1.21	1.56
Volatilidad de la máquina virtual	-	0.87	1.00	1.15	1.30	-
Tiempo de respuesta del ordenador	-	0.87	1.00	1.07	1.15	-
Capacidad del analista	1.46	1.19	1.00	0.86	0.71	-
Experiencia en la aplicación	1.29	1.13	1.00	0.91	0.82	-
Capacidad de los programadores	1.42	1.17	1.00	0.86	0.70	-
Experiencia en S.O. utilizado	1.21	1.10	1.00	0.90	-	-
Experiencia en el lenguaje de programa	1.14	1.07	1.00	0.95	-	-
Prácticas de programación modernas	1.24	1.10	1.00	0.91	0.82	-
Utilización de herramientas software	1.24	1.10	1.00	0.91	0.83	-
Limitaciones de planificación del proyecto	1.23	1.08	1.00	1.04	1.10	-

Nota. Presenta los valores de los conductores de costo para el modelo COCOMO II

FAE

$$= 1.00 \times 1.00 \times 1.15 \times 1.00 \times 1.56 \times 0.87 \times 0.87 \times 0.86 \times 0.82 \times 0.86 \times 0.90 \times 0.95 \\ \times 0.82 \times 0.83 \times 1.00$$

$$\mathbf{FAE = 0.4792}$$

Calculando el Esfuerzo:

$$E = a * (KLDC)^b * FAE \left(\frac{\text{PERSONAS}}{\text{MES}} \right) \quad (8). \text{ Esfuerzo}$$

$$E = 2.4 * (8.599)^{1.05} * 0.4792 \left(\frac{\text{PERSONAS}}{\text{MES}} \right)$$

$$E = 11.012 \left(\frac{\text{PERSONAS}}{\text{MES}} \right) = 11 \left(\frac{\text{PERSONAS}}{\text{MES}} \right)$$

Calculando el Tiempo de desarrollo:

$$T = c * (E)^d * (\text{Meses}) \quad (9). \text{ Tiempo de Desarrollo del Sistema}$$

$$T = 2.5 * (11.01)^{0.38} * (\text{Meses})$$

$$\mathbf{T = 6.2(\text{Meses})}$$

Calculando la Productividad:

$$PR = LCD/E \text{ (LDC/Personas Mes)}$$

$$PR = 8599/11.01 \text{ (LDC/Personas Mes)}$$

$$PR = 781.017 \text{ (LDC/Personas Mes)}$$

Calculando del Personal Promedio:

$$P = E/T \text{ (Personas)}$$

$$P = 11.01/6.2$$

$$P = 1.77 \text{ Equivalente a 2 personas}$$

Cálculo de Costo Personas mes (Salario promedio = 350 \$ o 2450 Bs)

$$\text{Costo Mes} = \text{Persona} * \text{Salario promedio entre programadores}$$

$$\text{Costo Mes} = 1.77 * 350 = 619 \$$$

Cálculo de Costo Total del Proyecto

$$\text{Costo Total} = \text{Costo Mes} * \text{Tiempo}$$

$$\text{Costo Total} = 619 * 6.2 = 3837 \$$$

En conclusión, se requiere un estimado de 2 personas trabajando alrededor de 6.2 Meses con un costo total de 3837 \$us que equivalente a 26.743.89 Bs.

4.3 SEGURIDAD

Uno de los aspectos clave en este proyecto es la implementación de normas de seguridad. Las normas ISO 27001 e ISO 27002 son fundamentales en este contexto: ISO 27001 evalúa y asegura el cumplimiento de estándares de seguridad, promoviendo la mejora continua mediante un conjunto de controles que reducen el riesgo de incidentes de seguridad en el funcionamiento del sistema dentro de la institución. ISO 27002 complementa esto proporcionando directrices prácticas para la implementación de controles de seguridad. Para este proyecto, se consideran los siguientes tipos de seguridad:

4.3.1 Seguridad Lógica

Gestión de Comunicaciones y Operaciones

Se tiene las siguientes medidas:

- Los respaldos (backups) de la base de datos del sistema deberán realizarse semanalmente.
- El personal involucrado y los usuarios deben cambiar la contraseña del sistema periódicamente, al menos una vez cada 20 días o una vez al mes.
- En el caso de administradores del sistema, se recomienda realizar el cambio de contraseña de manera regular.

Encriptación

Se tiene la siguiente medida:

- Se implementa encriptación de alta seguridad para las contraseñas, protegiendo este dato esencial para el acceso al sistema mediante un algoritmo de encriptación robusto.
- Se implementa trazabilidad acción del usuario (inicio de sesión, actualizaciones, eliminaciones, etc.) debe ser registrada.

4.3.2 Seguridad Física

Almacenamiento Seguro de Backups

- Se recomienda almacenar los backups en distintas ubicaciones para mayor seguridad.

- Los backups de la base de datos deben ser protegidos en áreas seguras con acceso restringido únicamente a personal autorizado.

4.4 PRUEBA

4.4.1. Prueba Unitarias

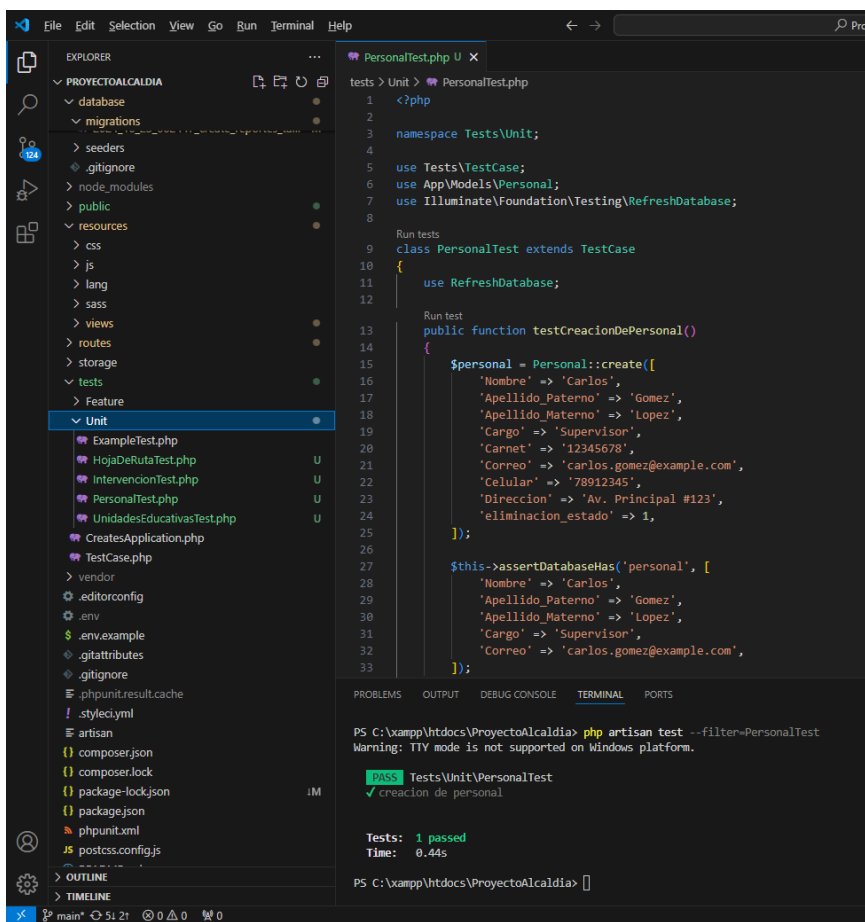
Se focaliza en ejecutar cada módulo (unidad mínima a ser probada, ej = una clase) lo que provee un mejor modo de manejar la integración de las unidades en componentes mayores.

Busca asegurar que el código funciona de acuerdo con las especificaciones y que el módulo lógico es válido.

Se realizo la prueba al modelo de unidades educativas y Personal Con la Herramienta PHPUnit:

Figura 37.

Pruebas Unitarias Personales



The screenshot displays the Visual Studio Code interface with a PHP unit test file, `PersonalTest.php`, open in the editor. The file is located in the `tests > Unit` directory. The code defines a `PersonalTest` class that extends `TestCase` and includes a `testCreacionDePersonal()` method. This method creates a `Personal` object with specific attributes and asserts that the database has the correct data.

```
tests > Unit > PersonalTest.php
1  <?php
2
3  namespace Tests\Unit;
4
5  use Tests\TestCase;
6  use App\Models\Personal;
7  use Illuminate\Foundation\Testing\RefreshDatabase;
8
9  Run tests
10 class PersonalTest extends TestCase
11 {
12     use RefreshDatabase;
13
14     Run test
15     public function testCreacionDePersonal()
16     {
17         $personal = Personal::create([
18             'Nombre' => 'Carlos',
19             'Apellido_Paterno' => 'Gomez',
20             'Apellido_Materno' => 'Lopez',
21             'Cargo' => 'Supervisor',
22             'Carnet' => '12345678',
23             'Correo' => 'carlos.gomez@example.com',
24             'Celular' => '78912345',
25             'Direccion' => 'Av. Principal #123',
26             'eliminacion_estado' => 1,
27         ]);
28
29         $this->assertDatabaseHas('personal', [
30             'Nombre' => 'Carlos',
31             'Apellido_Paterno' => 'Gomez',
32             'Apellido_Materno' => 'Lopez',
33             'Cargo' => 'Supervisor',
34             'Correo' => 'carlos.gomez@example.com',
35         ]);
36     }
37 }
```

The terminal output shows the command `php artisan test --filter=PersonalTest` being executed, resulting in a `PASS` status for the `Tests\Unit\PersonalTest` class and the `creacion de personal` test. The test passed successfully, and the execution time was 0.44s.

```
PS C:\xampp\htdocs\ProyectoAlcaldia> php artisan test --filter=PersonalTest
Warning: TTY mode is not supported on Windows platform.

PASS Tests\Unit\PersonalTest
  ✓ creacion de personal

Tests: 1 passed
Time: 0.44s

PS C:\xampp\htdocs\ProyectoAlcaldia>
```

Las pruebas realizadas a el modelo Personal fueron correctas.

Figura 38.

Pruebas Unitarias Unidades Educativas Parte I

The screenshot shows a code editor with the following PHP code in `UnidadesEducativasTest.php`:

```

1  <?php
2
3  namespace Tests\Unit;
4
5  use Tests\TestCase;
6  use App\Models\UnidadesEducativas;
7  use Illuminate\Foundation\Testing\RefreshDatabase;
8
9  Run tests
10 class UnidadesEducativasTest extends TestCase
11 {
12     use RefreshDatabase;
13
14     Run test
15     public function testRelacionJuntaEscolar()
16     {
17         // Crea manualmente una instancia de UnidadesEducativas sin persistir en la base de datos
18         $unidadEducativa = new UnidadesEducativas();
19
20         // Verifica que la relación juntaEscolar sea una instancia de HasOne
21         $this->assertInstanceOf(\Illuminate\Database\Eloquent\Relations\HasOne::class, $unidadEducativa->juntaEscolar());
22     }
23
24     Run test
25     public function testAtributosAsignables()
26     {
27         // Ajusta los datos según los tipos requeridos en la migración
28         $data = [
29             'Codigo_SIE_UE' => '1234',
30             'Nombre' => 'Unidad Educativa Ejemplo',
31             'Turno' => 'Mañana',
32             'Codigo_Infraestructura' => 123, // Cambia a entero según la migración
33             'Organizacion' => 'Organización Ejemplo',
34             'Distrito' => '1',
35             'Nivel' => 'Primaria'
36         ];
37     }
38 }

```

The terminal output shows the following results:

```

PS C:\xampp\htdocs\ProyectoAlcaldia> php artisan test --filter=UnidadesEducativasTest
Warning: TTY mode is not supported on Windows platform.

PASS Tests\Unit\UnidadesEducativasTest
PASS Tests\Unit\UnidadesEducativasTest
✓ relación junta escolar
✓ relación junta escolar
✓ atributos asignables
✓ casts

Tests: 3 passed

```

Figura 39.

Prueba Unitaria Unidad Educativa Parte II

The image shows a screenshot of an IDE (Visual Studio Code) with a dark theme. The Explorer view on the left shows a project structure for 'PROYECTOALCALDIA' with various folders and files, including a 'tests' directory containing 'Unit' and 'UnidadesEducativasTest.php'. The main editor window displays the code for 'UnidadesEducativasTest.php'. The code includes a test case 'testCasts()' that creates a 'UnidadEducativa' object and asserts its properties. The terminal at the bottom shows the command 'php artisan test --filter=UnidadesEducativasTest' and the output indicating that three tests passed successfully.

```

10  {
23  {
25      $data = [
33          'Direccion' => 'Calle Falsa 123',
34          'Latitud' => -17.1234,
35          'Longitud' => -66.1234,
36          'eliminacion_estado' => 1,
37      ];
38
39      $unidadEducativa = UnidadesEducativas::create($data);
40
41      $this->assertEquals($data['Nombre'], $unidadEducativa->Nombre);
42      $this->assertEquals($data['eliminacion_estado'], $unidadEducativa->eliminacion_estado);
43  }
44
45  Run test
46  public function testCasts()
47  {
48      $unidadEducativa = UnidadesEducativas::create([
49          'Codigo_SIE_UE' => '4321',
50          'Nombre' => 'Otra Unidad Educativa',
51          'Codigo_Infraestructura' => 456,
52          'Latitud' => 17.1234,
53          'Longitud' => 66.1234,
54          'eliminacion_estado' => 1,
55      ]);
56
57      $this->assertIsFloat($unidadEducativa->Latitud);
58      $this->assertIsFloat($unidadEducativa->Longitud);
59      $this->assertEquals(1, $unidadEducativa->eliminacion_estado);
60  }
61  }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\xampp\htdocs\ProyectoAlcaldia> php artisan test --filter=UnidadesEducativasTest
Warning: TTY mode is not supported on Windows platform.

PASS Tests\Unit\UnidadesEducativasTest
PASS Tests\Unit\UnidadesEducativasTest
✓ relacion junta escolar
✓ relacion junta escolar
✓ atributos asignables
✓ casts

Tests: 3 passed
Time: 0.50s

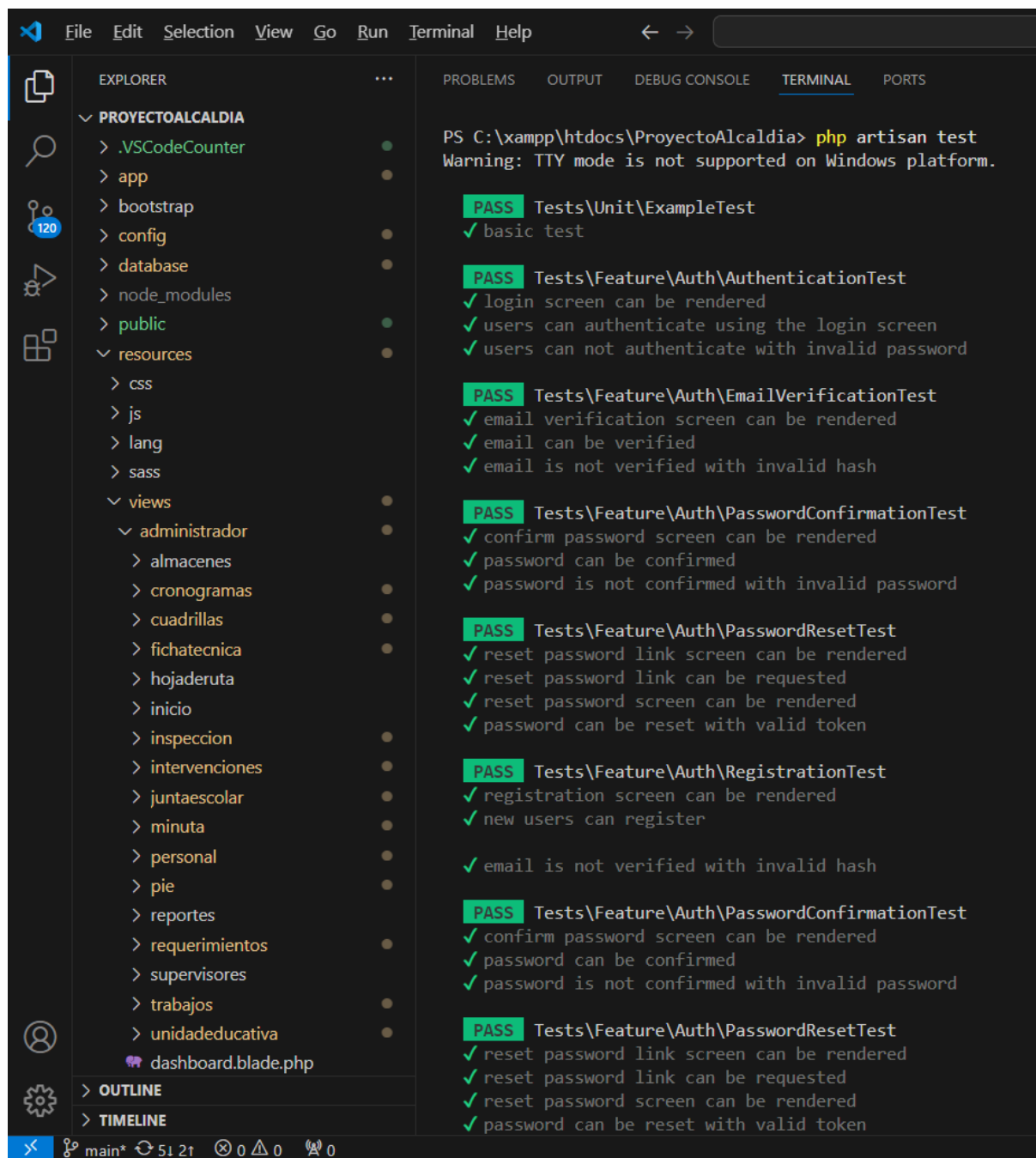
```

Las pruebas realizadas a el modelo Unidades Educativas fueron correctas.

4.4.2. Prueba de Integración

Para realizar un testesteo general atrás des de la herramienta proporcionada por Laravel Además de los comandos pest o phpunit, también puede utilizar el comando test de Artisan para ejecutar sus pruebas. El ejecutor de pruebas de Artizan ofrece informes detallados que facilitan el desarrollo y la depuración ejecutadas en la ruta principal del proyecto como se observa en la siguiente figura:

Figura 40.

Prueba Integración Raíz del Proyecto Parte I

The image shows a screenshot of the Visual Studio Code interface. On the left, the Explorer view displays the project structure for 'PROYECTOALCALDIA'. The file tree includes folders like .VSCodeCounter, app, bootstrap, config, database, node_modules, public, resources (with subfolders css, js, lang, sass, and views), and views (with subfolders administrador, almacenes, cronogramas, cuadrillas, fichatecnica, hojaderuta, inicio, inspeccion, intervenciones, juntaescolar, minuta, personal, pie, reportes, requerimientos, supervisores, trabajos, and unidadeducativa). The active file is 'dashboard.blade.php'.

The Terminal view on the right shows the output of a command: `PS C:\xampp\htdocs\ProyectoAlcaldia> php artisan test`. The output indicates that TTY mode is not supported on the Windows platform. Below this, several test results are shown, all marked as 'PASS' with green checkmarks:

- Tests\Unit\ExampleTest**
 - ✓ basic test
- Tests\Feature\Auth\AuthenticationTest**
 - ✓ login screen can be rendered
 - ✓ users can authenticate using the login screen
 - ✓ users can not authenticate with invalid password
- Tests\Feature\Auth\EmailVerificationTest**
 - ✓ email verification screen can be rendered
 - ✓ email can be verified
 - ✓ email is not verified with invalid hash
- Tests\Feature\Auth>PasswordConfirmationTest**
 - ✓ confirm password screen can be rendered
 - ✓ password can be confirmed
 - ✓ password is not confirmed with invalid password
- Tests\Feature\Auth>PasswordResetTest**
 - ✓ reset password link screen can be rendered
 - ✓ reset password link can be requested
 - ✓ reset password screen can be rendered
 - ✓ password can be reset with valid token
- Tests\Feature\Auth\RegistrationTest**
 - ✓ registration screen can be rendered
 - ✓ new users can register
 - ✓ email is not verified with invalid hash
- Tests\Feature\Auth>PasswordConfirmationTest**
 - ✓ confirm password screen can be rendered
 - ✓ password can be confirmed
 - ✓ password is not confirmed with invalid password
- Tests\Feature\Auth>PasswordResetTest**
 - ✓ reset password link screen can be rendered
 - ✓ reset password link can be requested
 - ✓ reset password screen can be rendered
 - ✓ password can be reset with valid token

Figura 41.

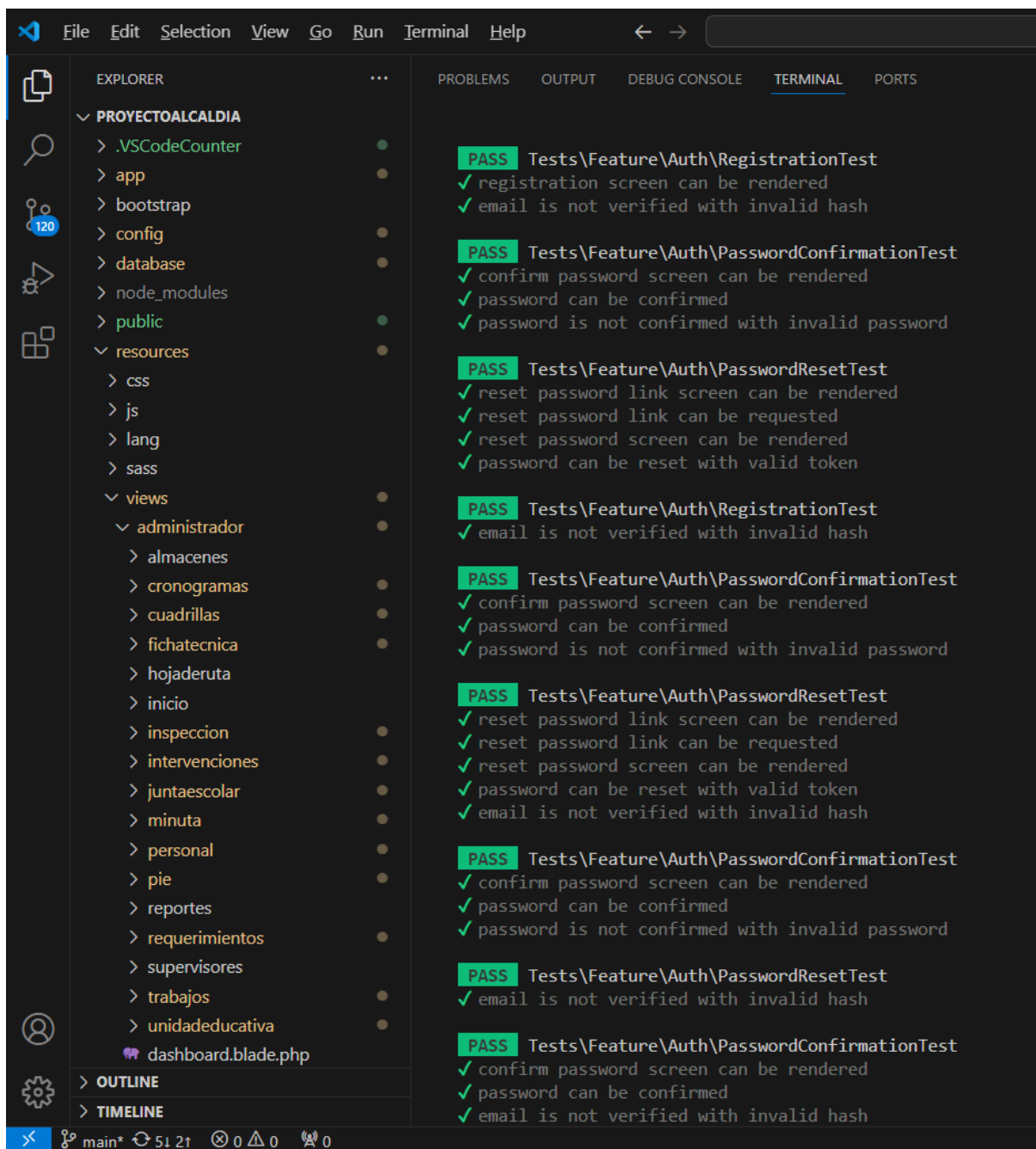
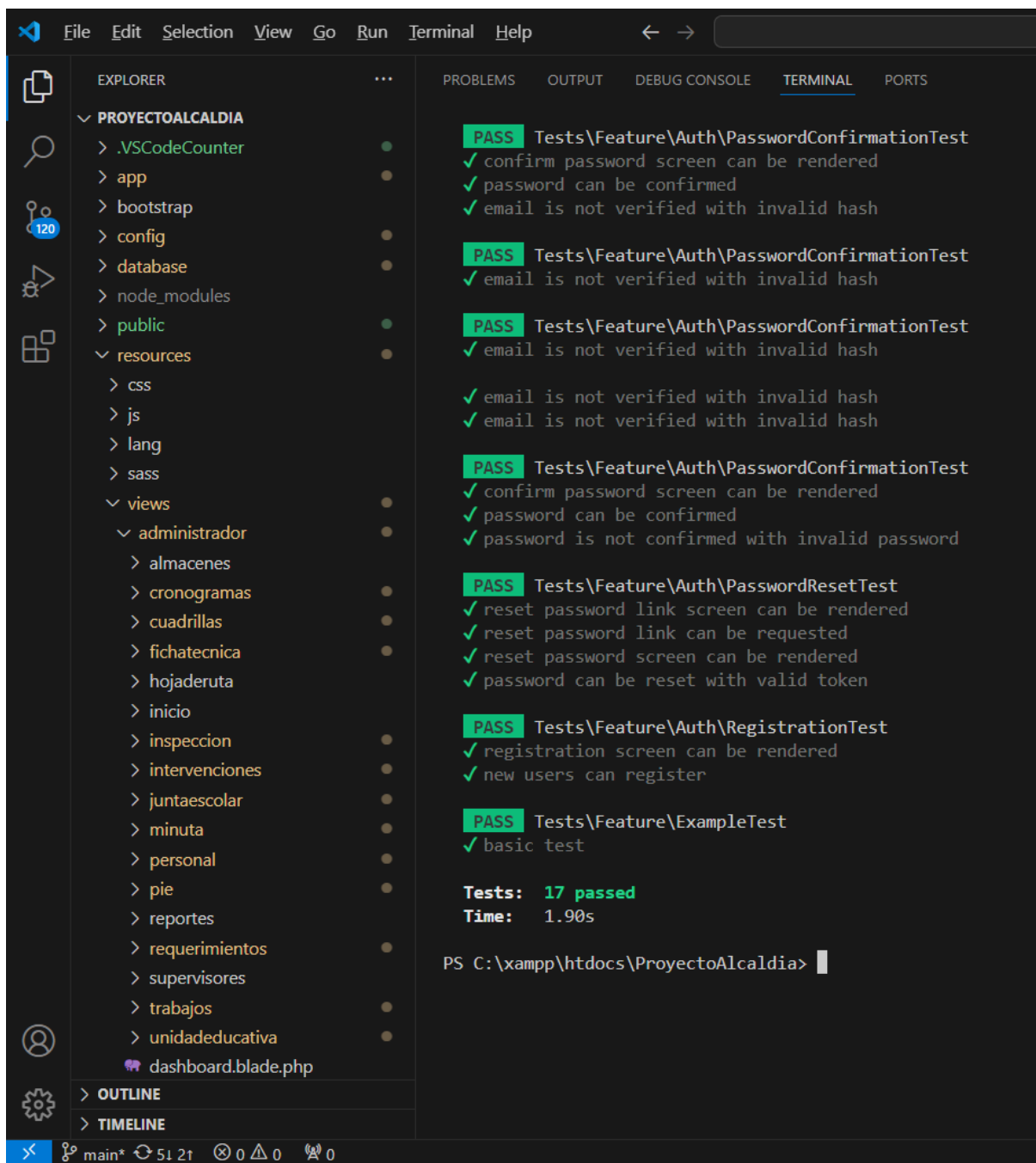
Prueba Integración Raíz del Proyecto Parte II

Figura 42.

Prueba Integración Raíz del Proyecto Parte III

The screenshot displays the Visual Studio Code interface with the Explorer view on the left showing the project structure for 'PROYECTOALCALDIA'. The Terminal view on the right shows the output of a test run, indicating that all tests passed.

```
File Edit Selection View Go Run Terminal Help
```

EXPLORER

- PROYECTOALCALDIA
 - .VSCodeCounter
 - app
 - bootstrap
 - config
 - database
 - node_modules
 - public
 - resources
 - css
 - js
 - lang
 - sass
 - views
 - administrador
 - almacenes
 - cronogramas
 - cuadrillas
 - fichatecnica
 - hojaderuta
 - inicio
 - inspeccion
 - intervenciones
 - juntaescolar
 - minuta
 - personal
 - pie
 - reportes
 - requerimientos
 - supervisores
 - trabajos
 - unidadeducativa
 - dashboard.blade.php
 - OUTLINE
 - TIMELINE

TERMINAL

```
PASS Tests\Feature\Auth\PasswordConfirmationTest
✓ confirm password screen can be rendered
✓ password can be confirmed
✓ email is not verified with invalid hash

PASS Tests\Feature\Auth\PasswordConfirmationTest
✓ email is not verified with invalid hash

PASS Tests\Feature\Auth\PasswordConfirmationTest
✓ email is not verified with invalid hash
✓ email is not verified with invalid hash

PASS Tests\Feature\Auth\PasswordConfirmationTest
✓ confirm password screen can be rendered
✓ password can be confirmed
✓ password is not confirmed with invalid password

PASS Tests\Feature\Auth\PasswordResetTest
✓ reset password link screen can be rendered
✓ reset password link can be requested
✓ reset password screen can be rendered
✓ password can be reset with valid token

PASS Tests\Feature\Auth\RegistrationTest
✓ registration screen can be rendered
✓ new users can register

PASS Tests\Feature\ExampleTest
✓ basic test

Tests: 17 passed
Time: 1.90s

PS C:\xampp\htdocs\ProyectoAlcaldia>
```

Quedando resultados exitosos del testeo.

Prueba de Hoja de ruta comprobando efectiva la relacion.

Figura 43.

Prueba de Integración de Hoja de Ruta Parte I

The screenshot shows an IDE window with the following content:

EXPLORER: Project structure for 'PROYECTOALCALDIA' including folders like 'database', 'migrations', 'seeder', 'public', 'resources', 'tests', and 'Unit'. The file 'HojaDeRutaTest.php' is selected in the 'Unit' folder.

Code Editor: Shows the content of 'HojaDeRutaTest.php':

```

1 <?php
2
3 namespace Tests\Unit;
4
5 use Tests\TestCase;
6 use App\Models\HojaDeRuta;
7 use App\Models\UnidadesEducativas;
8 use App\Models\JuntaEscolar;
9 use Illuminate\Foundation\Testing\RefreshDatabase;
10
11 Run tests
12 class HojaDeRutaTest extends TestCase
13 {
14     use RefreshDatabase;
15
16     Run test
17     public function testRelacionUnidadEducativa()
18     {
19         $unidadEducativa = UnidadesEducativas::create([
20             'Codigo_SIE_UE' => '1234',
21             'Nombre' => 'Unidad Educativa Ejemplo',
22             'Codigo_Infraestructura' => 123,
23             'latitud' => -17.1234.

```

TERMINAL: Shows the execution output:

```

Time: 0.52s

PS C:\xampp\htdocs\ProyectoAlcaldia> php artisan test --filter=HojaDeRutaTest
Warning: TTY mode is not supported on Windows platform.

PASS Tests\Unit\HojaDeRutaTest
✓ relacion unidad educativa
✓ relacion junta escolar
✓ relacion unidad educativa
✓ relacion junta escolar
✓ valores predeterminados
✓ relacion unidad educativa
✓ relacion junta escolar
✓ valores predeterminados
✓ relacion unidad educativa
✓ relacion junta escolar
✓ relacion unidad educativa
✓ relacion unidad educativa
✓ relacion junta escolar
✓ valores predeterminados

Tests: 3 passed
Time: 0.50s

```

Figura 44.

Prueba de Integración de Hoja de Ruta Parte II

The screenshot shows an IDE with the following components:

- EXPLORER:** A file tree on the left showing a project structure with folders like 'database', 'migrations', 'seeder', 'resources', 'tests', and 'vendor'. The 'tests' folder is expanded to show 'Unit' tests, with 'HojaDeRutaTest.php' selected.
- Code Editor:** The main area displays the PHP code for 'HojaDeRutaTest.php'. It includes a 'testValoresPredeterminados()' function that creates instances of 'UnidadEducativa' and 'JuntaEscolar', and then creates a 'HojaDeRuta' instance. Assertions are used to verify the 'Solicitud' and 'Estado' attributes.
- Terminal:** At the bottom, a terminal window shows the output of running the test. It displays a 'Warning: TTY mode is not supported on Windows platform.' followed by a 'PASS' status and a list of successful assertions:
 - ✓ relacion unidad educativa
 - ✓ relacion junta escolar
 - ✓ relacion unidad educativa
 - ✓ relacion junta escolar
 - ✓ valores predeterminados

Teniendo respuesta efectiva después de haber realizado el testeo.

Prueba de intervención para el modelo Intervención.

Figura 45.

Prueba de Integración Intervenciones

The screenshot shows an IDE with the following components:

- EXPLORER:** Shows the project structure for 'PROYECTOALCALDIA', including folders like 'app', 'Models', 'database', and 'tests'. The 'tests' folder is expanded to show 'Unit' > 'IntervencionTest.php'.
- Code Editor:** Displays the content of 'IntervencionTest.php':


```

1  <?php
2
3  namespace Tests\Unit;
4
5  use Tests\TestCase;
6  use App\Models\Intervencion;
7  use App\Models\UnidadesEducativas;
8  use App\Models\TrabajoCuadrilla;
9  use Illuminate\Foundation\Testing\RefreshDatabase;
10
11 Run tests
12 class IntervencionTest extends TestCase
13 {
14     use RefreshDatabase;
15
16     Run test
17     public function testRelacionUnidadEducativa()
18     {
19         $unidadEducativa = UnidadesEducativas::create([
20             'Codigo_SIE_UE' => '5678',
21             'Nombre' => 'Unidad Educativa Ejemplo',
22             'Codigo_Infraestructura' => 456,
23             'Latitud' => -17.9876,
24             'Longitud' => -66.4321,
25             'eliminacion_estado' => 1,
26         ]);
27
28         $intervencion = Intervencion::create([

```
- TERMINAL:** Shows the execution of the test command and its results:

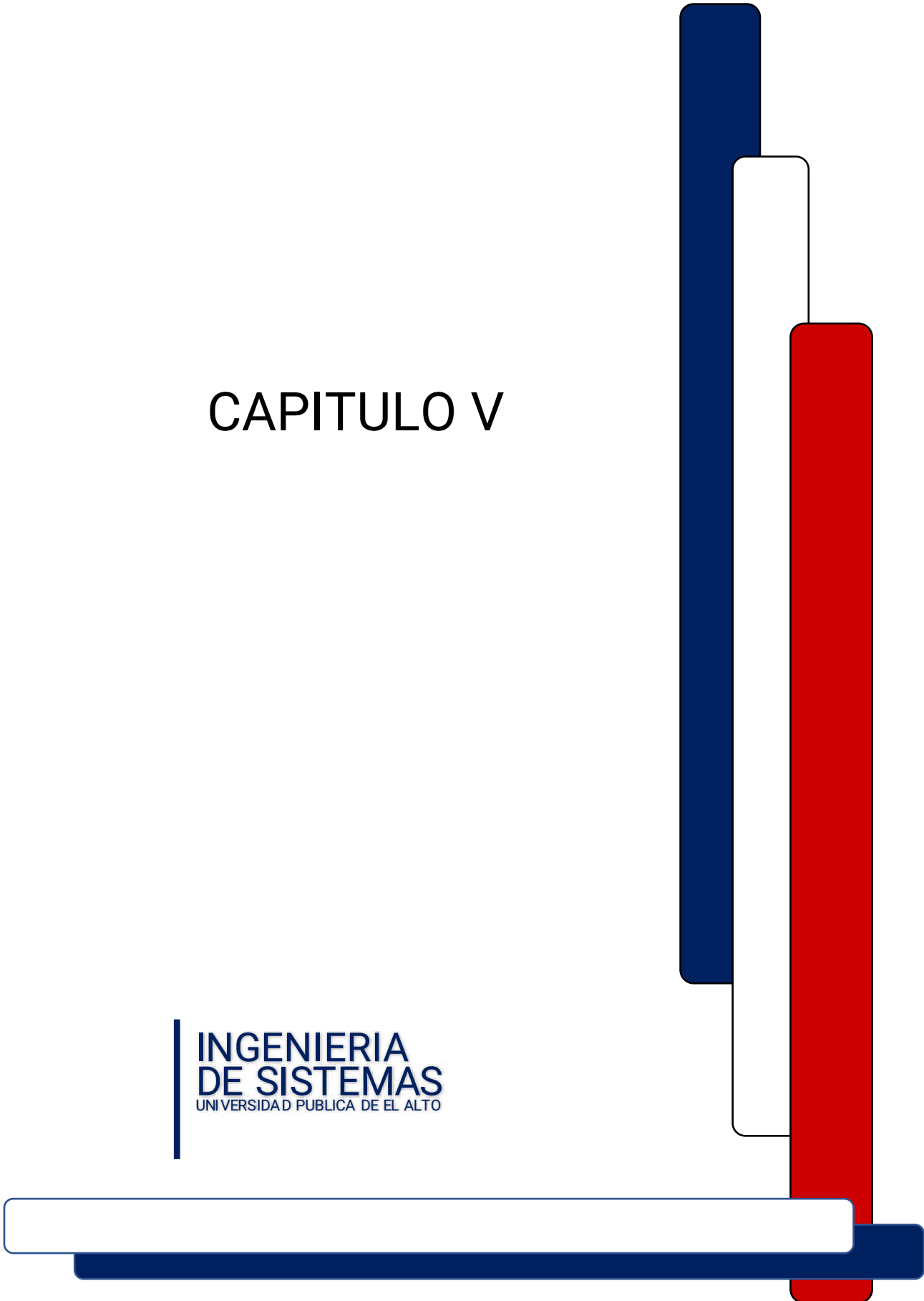

```

Time: 0.51s
PS C:\xampp\htdocs\ProyectoAlcaldia> php artisan test --filter=IntervencionTest
Warning: TTY mode is not supported on Windows platform.
PASS Tests\Unit\IntervencionTest
✓ relacion unidad educativa
✓ relacion trabajo cuadrilla
✓ atributos asignables
Tests: 3 passed
Time: 0.51s
PS C:\xampp\htdocs\ProyectoAlcaldia>

```


CAPITULO V

INGENIERIA
DE SISTEMAS
UNIVERSIDAD PUBLICA DE EL ALTO



CAPITULO V

5.CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

El objetivo general del proyecto se ha cumplido satisfactoria mente con el desarrollo de la plataforma web para administración y control de solicitudes de la unidad de mantenimiento utilizando las herramientas de diseño y desarrollo propuestos para la finalización del proyecto.

- Se realizó un análisis detallado de los procesos, lo cual permitió diseñar los módulos del sistema de forma organizada, adaptados a cada proceso.
- Se desarrolló una base de datos confiable y segura para almacenar toda la información relevante para el funcionamiento de las solicitudes y el proceso de trabajo.
- Se desarrollo módulos que registran el proceso de trabajo de la unidad de mantenimiento, desde la solicitud hasta la conclusión del trabajo.
- Se logró una mejora significativa en los tiempos operativos al sustituir los procesos manuales por un sistema digital que permite acceder a la información de manera oportuna y efectiva.
- El sistema permite generar informes de inspecciones realizadas de acuerdo a visitas técnicas, además de informe de conclusión de trabajo de las unidades educativas.

5.2. RECOMENDACIONES

A partir del presente proyecto, se proponen las siguientes recomendaciones para mejorar el sistema:

- Capacitar a los nuevos administradores para que puedan operar el sistema con mayor eficacia y mantener un mejor control de los proyectos gestionados.
- Realizar el mantenimiento del software en intervalos de tiempo establecidos,

siguiendo las normas preestablecidas.

- Cambiar las contraseñas de forma regular para proteger el sistema y evitar accesos no autorizados.
- Antes de ampliar o crear nuevos módulos, revisar la documentación para tomar decisiones informadas, ya que el sistema contiene elementos reutilizables que pueden facilitar el desarrollo de nuevas funcionalidades.

Tabla 24.

Tabla comparativa

Antes del Sistema	Con el Sistema
Respuestas a solicitudes eran tardías y a veces sin respuesta.	Las solicitudes se reciben en tiempo real, consecuentemente las respuestas son oportunas y cuentan con diferentes estados (pendiente, en proceso, completado),
Información dispersa de registros de trabajos realizados de manera desorganizada, sin seguimiento del trabajo efectuado.	Seguimiento del trabajo desde la inspección hasta la culminación del mismo, respaldo de que se llevo a cabo el trabajo por medio de un reporte detallado de la intervención realizada en el momento que se lo requiera.
Acceder a las solicitudes eran mediante libros de registro, generando demoras en seguimiento de las solicitudes o en la búsqueda de los mismos.	El acceso de las solicitudes ya no demora como antes buscaban hoja por hoja el registro, ahora con solo el código o fecha de la solicitud, el acceso a la información es inmediata.

Antes del Sistema	Con el Sistema
Los registros de inspecciones antes eran registrados, en Excel o en libros de actas tardando en el proceso manual de tener un registro que detalle sobre cuanto material usaron si la unidad educativa brindo el material	El proceso de inspección ahora es registrado en el sistema, con los requerimientos de material, generando el acta de inspección digitalizada para su fácil compartimiento con los responsables de la Unidades educativas, así mismo en base a la inspección, se obtiene de manera sincronizada con el almacén si la unidad requiere de material de la unidad de mantenimiento en tiempo real.
Antes el personal de cuadrillas la mano de obra, adjuntaba el trabajo realizado en un grupo de WhatsApp donde en el lapso de tiempo adjuntaban fotos de trabajos realizados de manera general, donde en ocasiones por el tiempo estos datos se perdían al no ser descargados.	Ahora las cuadrillas pueden adjuntar detalles y imágenes sobre el trabajo que realizaron en las unidades educativas, mismas imágenes que ahora se guardan de forma segura, así mismo el administrador tendrá la facilidad de acceder a ellas en base a las intervenciones que va registrar como concluidas para luego generar el documento de respaldo del trabajo concluido.
Búsqueda de las unidades educativas en Google maps ya que no tenían dirección o ubicación de la unidad educativa de donde tenían que intervenir, por lo que algunas unidades educativas eran difícilmente ubicables para el personal de la unidad.	Ahora las unidades educativas están centralizadas en un mapa donde se puede buscar por el nombre de la unidad educativa y obtener la dirección de la misma sin complicación en tiempo real.

REFERENCIAS BIBLIOGRAFICAS

Aleman Ballares, F. Y. (2016). *DESARROLLO DE UN SISTEMA WEB PARA LA GESTION DE*

*INCIDENCIAS INFORMATICAS EN EL AREA DE INFRAESTRUCUTRA Y SOPORTE
TECNICO DE LA EMPRESA EDPYME RAIZ S.A. LIMA-PERU.*

BARBARA BRON FONCESA, K. M. (2019). *Sistema informatico para la gestion de reportes de incidencias de mantenimiento en la Facultad de Ciencias y Tecnologia Computacionales.* La Habana-Cuba.

Bertalanffy, L. v. (1968). *Teoria General de los Sistemas.* Nueva York: George Braziller.

Boehm, B. W. (1976). *Software Engineering Economics.* Prentice-Hall.

Buitrago, D. R. (2021). *DESARROLLO DE SOFTWARE PARA LA ATENCION DE SINIESTROS POR PARTE DE ENTES ESTATALES EN LA CIUDAD DE PEREIRA.* PEREIRA-COLOMBIA.

Chile, G. d. (2004). *Guia Para Desarrollo Web.* Santiago de Chile.

CLAUDIA, C. C. (2020). *SISTEMA DE INFORMACION WEB PARA EL CONTROL Y SEGUIMIENTO ACADEMICO.* El Alto-Bolivia.

GAHUAPAZA, H. C. (2020). *SISTEMA WEB DE GESTION Y CONTROL DE ALMACENES .* El Alto-Bolivia.

James A. Senn. (1978). *Análisis y diseño de sistemas de información.* Nueva York: MacGraw-Hill.

Laudon, K. C. (2012). *istemas de información gerencial.* Mexico: Pearson Educacion.

Ifonso Barca Lozano, S. A.-C.-B. (2008). *Motivación y aprendizaje en el alumnado de educación secundaria y rendimiento académico: un análisis desde la diversidad e inclusión educativ.*

Luigui, C. P. (2018). *SISTEMA WEB PARA LA GESTION DE INCIDENCIAS.* LIMA-PERU.

Montilva, J. (1999). *Desarrollo de sistemas de informacion.* Universidad de los Andes.

Natividad, B. B. (2017). *DISEÑO DE SISTEMA DE GESTION DE INCIDENCIAS INFORMATICAS EN DSE INGENIERA SAC LIMA- PERU.* LIMA-PERU.

Pandiani, W. (2005). *nternational Organization for Standardization.* Geneva: Switzerland: ISO.

PAREJA, J. S. (2014). *SISTEMA WEB DE REGISTRO Y CONTROL DE UNIDADES EDUCATIVAS*

ANEXOS

**INGENIERIA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



GOBIERNO AUTÓNOMO MUNICIPAL DE LOS RÍOS

PLAN DE INSPECCIÓN TÉCNICA Y REQUERIMIENTO DE MATERIALES

UNIDAD EJECUTORA: **PROYECTO DE OBRAS DE RECONSTRUCCIÓN DEL PUERTO DE LA VILLA DE LOS RÍOS** IDENTIFICACION: **7**

FECHA DE EMISIÓN: **15/07/2024** FECHA DE EJECUCIÓN: **07/07/2024** VALOR ESTIMADO: **142.700**

TELÉFONO: **73241503**

DESCRIPCIÓN Y OBSERVACIÓN

Se realizó un levantamiento de Actos, que no cumple con los requisitos de la normativa técnica de la zona.

Se requiere de los siguientes materiales:

Actos de los Actos (Piso)

No cumple con Norma de Construcción

Se requiere de los siguientes materiales:



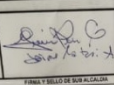
Actos de los Actos (Piso)

REQUERIMIENTO DE MATERIAL

DESCRIPCIÓN	CANTIDAD	MATERIAL		AUTORIZACIÓN
		UNIDAD	VALOR ESTIMADO	
Actos	20	ml	400	Actos de los Actos (Piso)
Actos de los Actos	10	ml	5	Actos de los Actos (Piso)
Actos de los Actos	16	ml	5	Actos de los Actos (Piso)
Actos de los Actos	3	ml		
Actos de los Actos	6	ml		
Actos de los Actos	40	ml		
Actos de los Actos	2	ml	1	Actos de los Actos (Piso)
Actos de los Actos	2	ml	1	Actos de los Actos (Piso)
Actos de los Actos	20	ml	5	Actos de los Actos (Piso)

LA NOTACIÓN TÉCNICA ESTA SUJETA A EVALUACIÓN TÉCNICA Y DISPONIBILIDAD EN ALMACENES

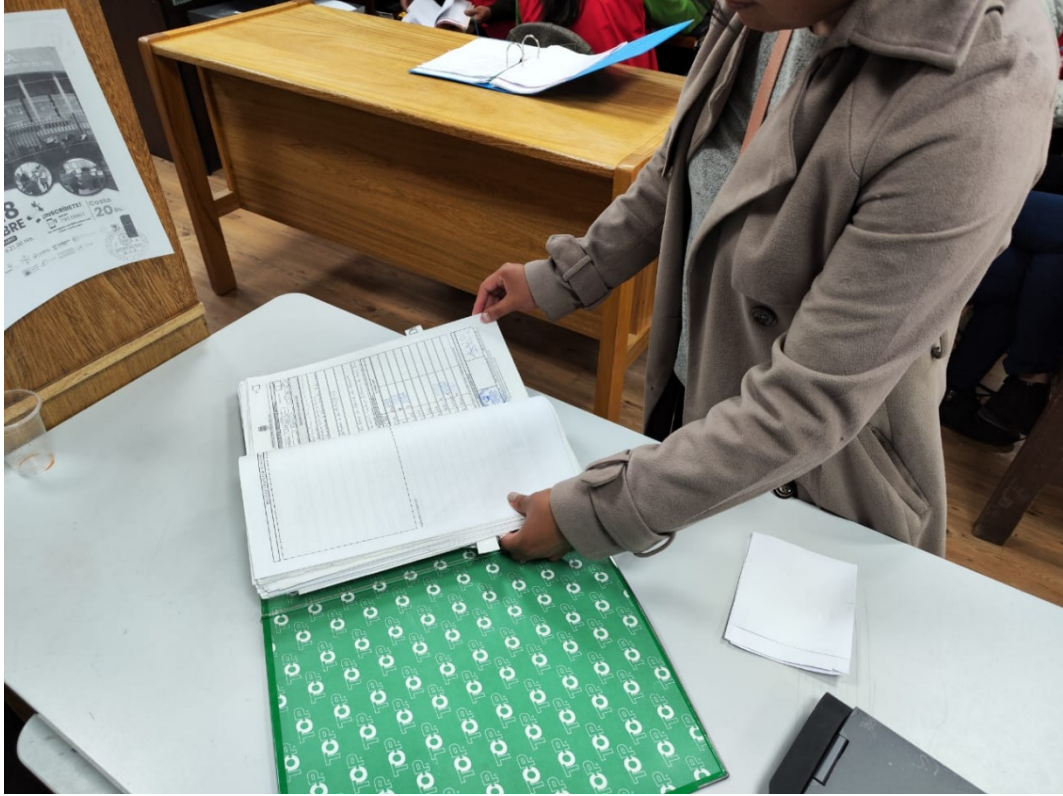
FIRMAS DE INSPECCIÓN

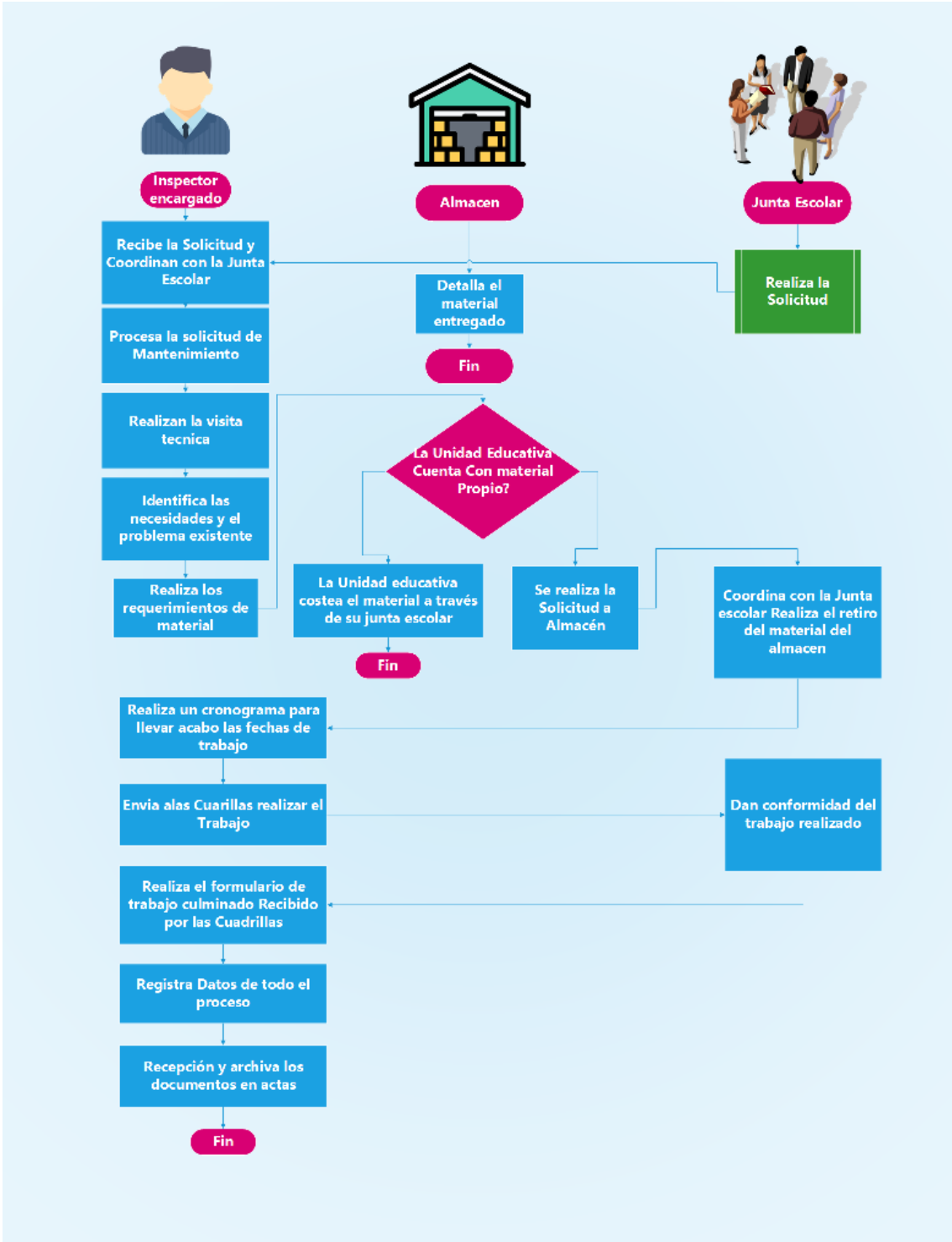
PROYECTO DE OBRAS DE RECONSTRUCCIÓN DEL PUERTO DE LA VILLA DE LOS RÍOS FECHA Y SELLO DE TÉCNICO INSPECTOR FECHA Y SELLO DE SUB ALCAIDE

Avenida Constitución, No. 9601 - Montecristo (Cuenca), entre calles 1 y 2 - Tarata y calle Manuel Velez Guzmán
 Cuenca - P.O. Box 1000, 6ª planta, 6ª planta de la estación de bomberos.









Inspector encargado



Almacen



Junta Escolar

Recibe la Solicitud y Coordinan con la Junta Escolar

Procesa la solicitud de Mantenimiento

Realizan la visita técnica

Identifica las necesidades y el problema existente

Realiza los requerimientos de material

Detalla el material entregado

Fin

Realiza la Solicitud

La Unidad Educativa Cuenta Con material Propio?

La Unidad educativa costea el material a través de su junta escolar

Fin

Se realiza la Solicitud a Almacén

Coordina con la Junta escolar Realiza el retiro del material del almacen

Realiza un cronograma para llevar acabo las fechas de trabajo

Envia alas Cuanillas realizar el Trabajo

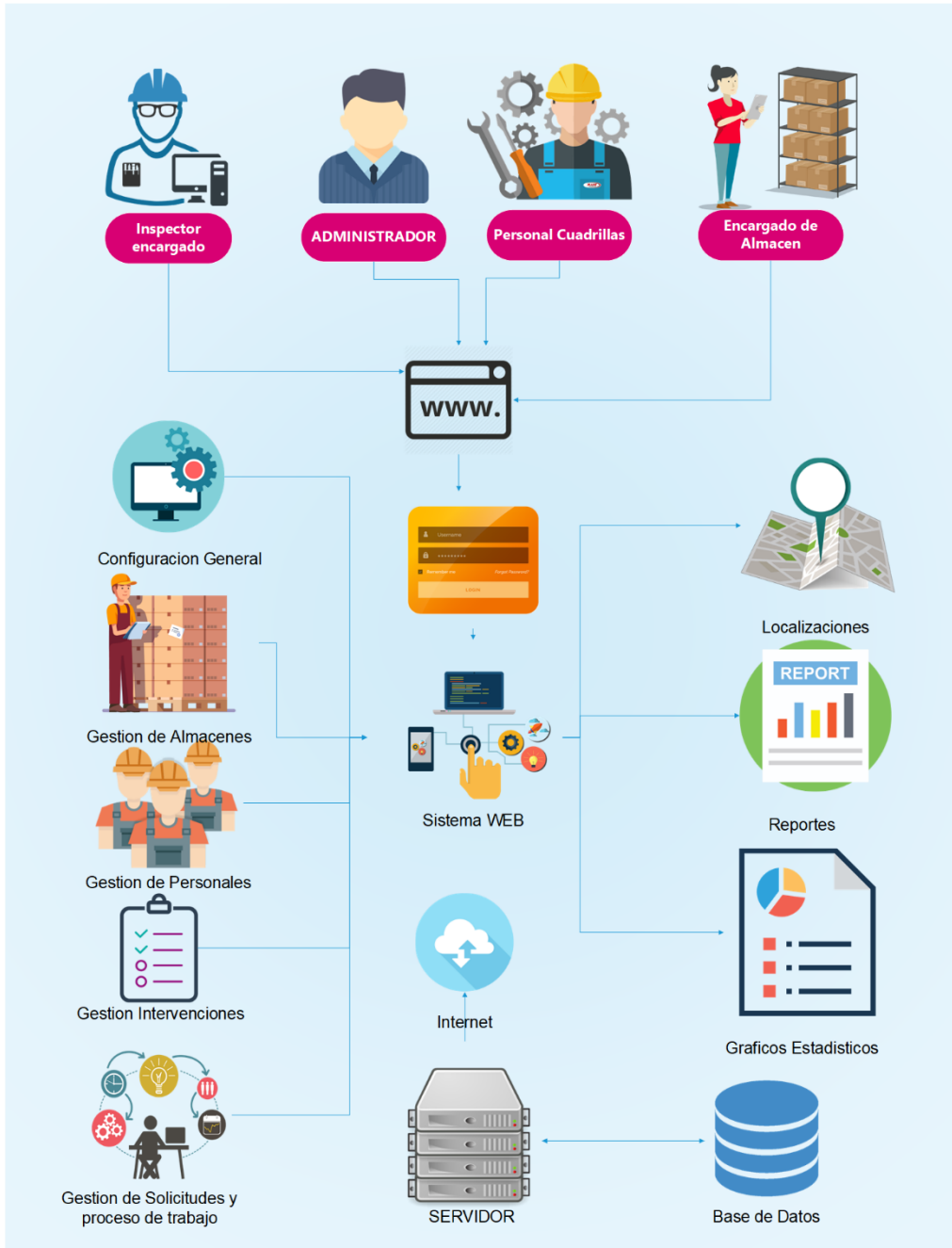
Dan conformidad del trabajo realizado

Realiza el formulario de trabajo culminado Recibido por las Cuadrillas

Registra Datos de todo el proceso

Recepción y archiva los documentos en actas

Fin





MANUAL DE USUARIO

I. INTRODUCCION

El presente manual de administración tiene como objetivo servir como una guía práctica para el uso adecuado del sistema web de gestión y control. Este sistema está diseñado para optimizar el registro, seguimiento y gestión de las solicitudes relacionadas con problemas de infraestructura en las unidades educativas, mejorando así los procesos de trabajo de la Unidad de Mantenimiento del Gobierno Autónomo Municipal de El Alto.

II. OBJETIVO DEL SISTEMA

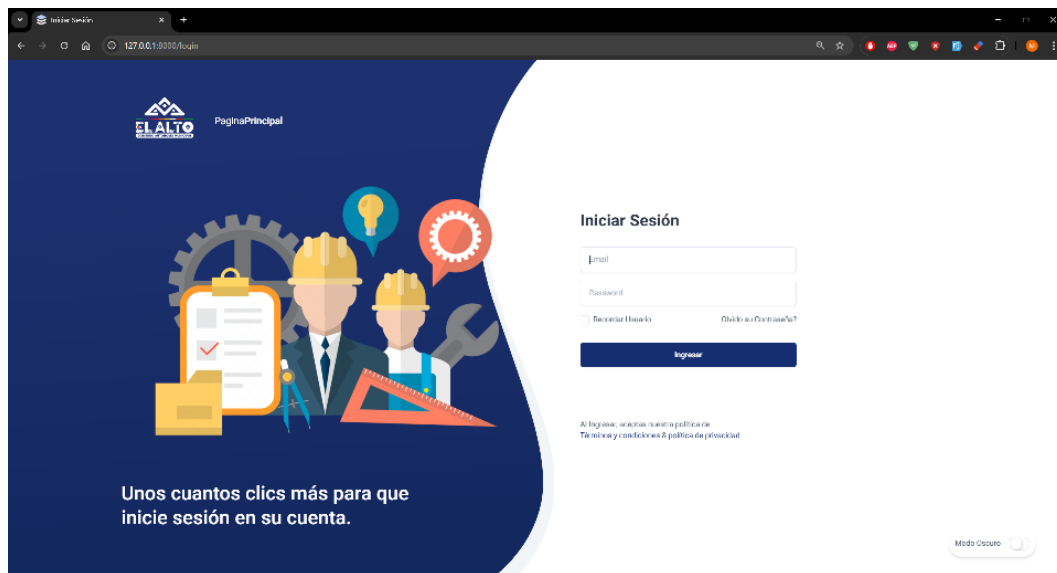
El objetivo de este manual es proporcionar al usuario final una guía clara y completa para la adecuada administración del sistema "SISTEMA DE INFORMACION WEB PARA EL REGISTRO Y CONTROL DE EMERGENCIAS EN INFRAESTRUCTURAS DE LAS UNIDADES EDUCATIVAS EN LA CIUDAD DE EL ALTO". Donde el usuario podrá registrar el proceso de trabajo desde la solicitud de inicio, hasta la conclusión del trabajo termina la intervención.

III. GUIA DEL SISTEMA

Interfaz Inicio de Sesión

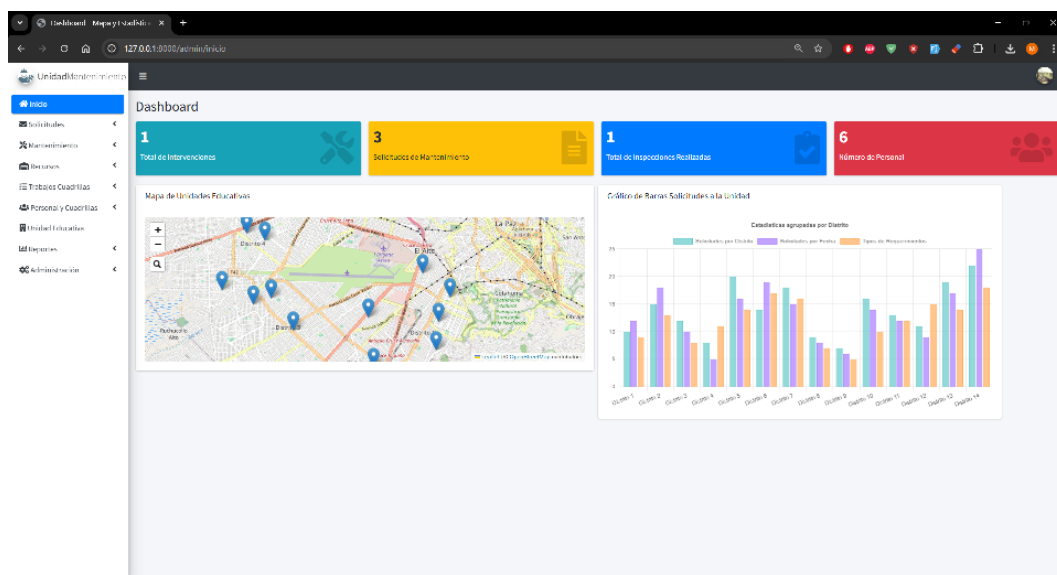
Para ingresar a sistema primero se deberá de iniciar sesión. Donde ingresara el Correo y

la Contraseña asignada.



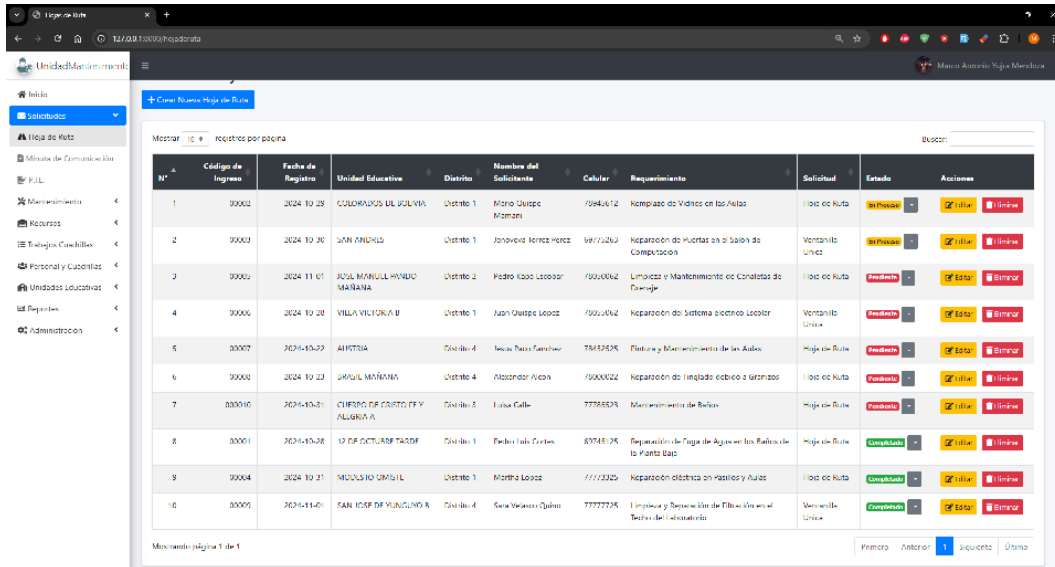
Página Principal

Una vez que el usuario ingresa el sistema le permitirá acceder a la página principal, desde donde podrá navegar por las distintas opciones del menú.



Módulo de Solicitudes de las Unidades Educativas

Se listan las solicitudes realizadas por las unidades educativas especificando el Requerimiento de la incidencia que puedan tener.



N°	Código de Ingreso	Fecha de Registro	Unidad Educativa	Distrito	Nombre del Solicitante	Celular	Requerimiento	Solicitud	Estado	Acciones
1	00002	2024-10-29	COLUMBIOS DE SULLVA	Distrito 1	Rafael Guispe Mamani	7984614	Reparación de Vidrios en las Habitaciones	Hoja de Ruta	Revisado	Verificar, Imprimir, Eliminar
2	00003	2024-10-30	SAN ANDRÉS	Distrito 1	Johanna Torres Perez	98772263	Reparación de Puertas en el Salón de Computación	Verificación Única	Revisado	Verificar, Imprimir, Eliminar
3	00005	2024-11-01	JOSÉ MANUEL HUAYU MARAÑA	Distrito 2	Pedro Kato Lopez	99000062	Limpieza y Mantenimiento de Conchales de Pasaje	Hoja de Ruta	Revisado	Verificar, Imprimir, Eliminar
4	00006	2024-10-20	VILLA VICTORIA B	Distrito 1	Juan Guispe Lopez	99000062	Reparación del Sistema eléctrico Local	Verificación Única	Revisado	Verificar, Imprimir, Eliminar
5	00007	2024-10-29	ALFROSA	Distrito 4	Isela Pazo Sanchez	78452574	Plata y Mantenimiento de las Aulas	Hoja de Ruta	Revisado	Verificar, Imprimir, Eliminar
6	00008	2024-10-23	URBIL MARAÑA	Distrito 4	Alexander Alcon	99000062	Reparación de Limpieza exterior a Limpieza	Hoja de Ruta	Revisado	Verificar, Imprimir, Eliminar
7	00010	2024-10-31	CUERPO DE CASTO FE Y ALBAKOLA	Distrito 3	Luz Calle	77786528	Mantenimiento de Beñas	Hoja de Ruta	Revisado	Verificar, Imprimir, Eliminar
8	00001	2024-10-28	12 DE OCTUBRE TARDIF	Distrito 1	Pedro Luis Cortes	83748124	Reparación de Fuga de Agua en las Habitaciones y Plantas de la Planta Baja	Hoja de Ruta	Completado	Verificar, Imprimir, Eliminar
9	00004	2024-10-21	MOLLOS OMBIL	Distrito 1	Martha Lopez	77773320	Reparación eléctrica en Habitaciones y Aulas	Hoja de Ruta	Completado	Verificar, Imprimir, Eliminar
10	00009	2024-11-01	SAN JOSE DE MUNGUNDO B	Distrito 4	Sara Velasco Quiro	77777774	Limpieza y Conservación de Filtros en el Beñas del Laboratorio	Verificación Única	Completado	Verificar, Imprimir, Eliminar

Para agregar nuevas solicitudes se debe llenar el siguiente formulario donde los campos serán validados además de tener una guía de cómo debe llenarse cada campo en caso de dudas a continuación se muestra el formulario para agregar nuevas solicitudes:

✕
Crear Hoja de Ruta

Código de Ingreso ⓘ

Seleccione Unidad Educativa ⓘ

Distrito

Nombre del Solicitante ⓘ

Cargo ⓘ

Celular ⓘ

Requerimiento ⓘ

Fecha de Registro ⓘ

Solicitud ⓘ

Módulo de Mantenimiento Sub módulo de Inspecciones

Previamente después de recibir la Solicitud de la Unidad Educativa Se tiene el registro de la Inspección realizada a la unidad educativa encontrando los problemas y requerimientos de material que se pueda necesitar para el trabajo que luego serán impresos por el inspector.

+ Nueva Inspección

N°	Unidad Educativa	Turno	Distrito	Nombre Representante	Celular	Fecha de Inspección	Problemas	Acciones
1	COLORADOS DE BOLIVIA	Mañana	Distrito 1	Mario Quispe Mamaní	78945612	2024-11-04	Infraestructura: Aulas, Trabajo: Instalacion Otros, Descripción: Reemplazo de Vidrios a las aulas, cambio de ventanas, Observación: N/A	Ver Actualizar Eliminar
2	12 DE OCTUBRE TARDE	Tarde	Distrito 1	Pedro Luis Cortes	69745125	2024-11-01	Infraestructura: Baños, Trabajo: Instalacion Sanitaria, Descripción: Reparación Tanque de agua, Observación: Ninguna	Ver Actualizar Eliminar
3	SAN ANDRES	Mañana	Distrito 1	Jehoveva Torrez Perez	69775263	2024-10-28	Infraestructura: Aulas, Trabajo: Instalacion Otros, Descripción: Centrado de Puertas, Observación: Ninguno	Ver Actualizar Eliminar
4	MODESTO OMISTE	Tarde	Distrito 1	Martha Lopez	77773325	2024-10-29	Infraestructura: Aulas, Trabajo: Instalacion Electrica, Descripción: Requiere reinstalación de cableado electrico, Observación: Ninguno	Ver Actualizar Eliminar
5	SAN JOSE DE YUNGUYO B	Tarde	Distrito 4	Sara Velasco Quino	77777725	2024-10-29	Infraestructura: Otros, Trabajo: Instalacion Otros, Descripción: Mantenimiento y reparación de los techos del laboratorio, Observación: Ninguno	Ver Actualizar Eliminar

Mostrando página 1 de 1 Primero Anterior 1 Siguiente Último

Para agregar una nueva inspección se deberá llenar el siguiente formulario donde el

usuario accederá y seleccionara la unidad educativa y se llenaran campos para facilitar la información de la unidad educativa, además de contemplar que puedan existir solicitudes de más de una unidad educativa se tiene la opción de seleccionar la hoja de ruta correspondiente donde se llenaran los campos y la información para coordinar con representante de la unidad educativa que realizo la solicitud, además del tipo de requerimiento que llego en la fecha de la solicitud, se registrara la inspección en fecha tomando en cuenta los problemas detectados en la visita técnica. Contemplando que pueda existir más de un problema se agregó un botón para añadir más de un problema

Selección Unidad Educativa ⓘ	Turno	Distrito
12 DE OCTUBRE TARDE - Turno: Tarde	Tarde	Distrito 1
Selección Hoja de Ruta ⓘ		
00001 - 2024-11-03		
Nombre del Representante	Celular	
Pedro Luis Cortes	77772345	
Fecha de Registro	Requerimiento	
2024-11-03	Se necesita Reinstalación Eléctrica	
Solicitud	Estado	
Hoja de Ruta	Pendiente	
Inspector ⓘ	Fecha de Inspección ⓘ	
Marco Antonio Yujra Mendoza	19/11/2024	
Problemas Detectados:		
Tipo de Infraestructura	Tipo de Trabajo	
Aulas	Instalación Eléctrica	
Descripción del Problema		
Cables dañados por sobre voltaje		
Observaciones (Opcional)		
Ninguna		
Agregar otro problema		
		Guardar Cerrar

Módulo de Mantenimiento

Sub módulo de Requerimientos de Material realizada la inspección, aquí basado en la inspección se realiza en requerimiento del material, para hacer la solicitud de material a la unidad educativa o en caso de que no cuenten con material, la unidad tiene un almacén que registra estas solicitudes de las cuales se verán más adelante.

N°	ID	U.E y Inspección	Descripción	Cantidad	Unidad de Medida	Autorización	Fuente Material	Acciones
1	1	COLORADOS DE BOLIVIA - 2024-11-04	se necesita ventanas nuevas	5.00	Vidrios 1.5*50	Fernando Apaza	Propio	Ver Editar Eliminar
2	2	12 DE OCTUBRE TARDE - 2024-11-01	Parche sellador	2.00	Parche	Fernando Apaza	Propio	Ver Editar Eliminar
3	3	SAN ANDRES - 2024-10-28	Visagras con Tornillos	3.00	Unidades	Fernando Apaza	Propio	Ver Editar Eliminar
4	4	MODESTO OMISTE - 2024-10-29	Rollos de Cable Eléctrico de 14 mm	2.00	Rollos	Fernando Apaza	Almacen	Ver Editar Eliminar
5	5	SAN JOSE DE YUNGUYO B - 2024-10-29	Requiere calaminas	3.00	Hojas	Julian apaza	Almacen	Ver Editar Eliminar

Se registrara en el siguiente formulario asociado la unidad educativa y la inspección que se tenga en curso.

Nuevo Requerimiento de Material

Buscar Unidad Educativa
12 DE OCTUBRE TARDE - Tl

Buscar Inspección
Fecha: 2024-11-19

Descripción del Material
Cable Eléctrico

Cantidad
1

Unidad de Medida
ROLLO

Autorizado por
Juan Torrez

Fuente de Material
Material Propio
Selecione una fuente
Material Propio
Material del Almacén

Agregar otro requerimiento

Guardar Requerimientos **Cancelar**

Módulo de Recursos Sub módulo de Almacén

Previamente después de Realizar la inspección y tener el requerimiento de materiales se tiene la gestión de alemanes, donde previamente la unidad educativa tuvo que especificar si contaba con material propio o dependiente del almacén de la Unidad de Mantenimiento de Infraestructuras la solicitud de material se realiza con la junta Escolar y el Encargado de Almacén hace el despacho de material para la Unidad Educativa previa autorización y dependiente de la inspección del Inspector Encargado.

#	Unidad Educativa	Turno	Distrito	Descripción del Requerimiento	Material Entregado Por	Material Recibido Por	Fecha de Entrega	Descripción del Material Entregado	Acciones
1	SAN JOSE DE YUNGUYO B	Tarde	Distrito 4	Requiere calaminas	Juan Perez	Lucia Mendez	2024-11-04	Fueron entregadas 3 hojas de c...	Editar Eliminar

Para agregar un nuevo registro de material saliente del almacén se llega el formulario especificando los siguientes detalles:

Agregar Nuevo Registro
✕

Unidad Educativa

Turno

Distrito

Fecha de Inspección

Descripción del Requerimiento

Material

Cantidad

Material Entregado Por

Material Recibido Por

Fecha de Entrega

Descripción del Material Entregado

Módulo de Mantenimiento Sub módulo de Cronogramas

Previamente después de recibir la Solicitud de la Unidad Educativa se realiza el cronograma asignando al personal de trabajo de mano de obra para el trabajo que luego serán impresos por el inspector.

N°	Día	Fecha	Unidad Educativa	Distrito	Jefe de Cuadrilla	Cuadrilla	Junta Escolar	Celular	Trabajo a Realizar	Detalle	Acciones
1	Lunes	2024-11-01	12 DE OCTUBRE TARDE	Distrito 1	Federico Saazo Lantco	Electrico	Pedro Luis Cortes	69745125	Reparacion del tanque de agua	Ninguno	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
2	Miércoles	2024-11-06	COLORADOS DE BOLIVIA	Distrito 1	Fernando Ali Mendoza	Albañileria	Mario Quispe Mamani	78945612	Reemplazo de Vidrios	Ninguno	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
3	Lunes	2024-11-06	SAN JOSE DE YUNGUYO B	Distrito 4	Estrella Palacio Dueñas	Tingladista	Sara Velasco Quino	77777725	Reparación de Techos	Ninguna	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
4	Martes	2024-10-29	MODESTO OMISTE	Distrito 1	Estrella Palacio Dueñas	Tingladista	Martha Lopez	77773325	Reparación y mantenimiento de...	Ninguno	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
5	Lunes	2024-10-28	SAN ANDRES	Distrito 1	Aquiles Torres Perez	Albañileria	Jenoveva Torrez Perez	69775263	Reparación de la puerta del la...	Ninguno	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Para coordinar asignar a las cuadrillas correspondientes al día de trabajo en cuestión y lugar donde deben realizar la visita técnica se tiene el siguiente formulario donde se designan fechas para que las cuadrillas intervengan a la unidad educativa.

✕
Crear Nuevo Cronograma

Selección Unidad Educativa

Turno

Distrito

Selección Hoja de Ruta

Nombre del Representante

Celular

Fecha de Registro

Requerimiento

Tipo de Cuadrilla

Asignar Personal

Fecha

Trabajo a Realizar

Detalle

Cerrar
Crear Cronograma

Módulo de Trabajo Cuadrillas Sub módulo de Subir Imágenes de Trabajos

Una vez las cuadrillas encargadas de realizar el trabajo solicitado describen el trabajo realizado y adjuntan fotografías para el informe de Ficha Técnica.

N°	Unidad Educativa	Cuadrilla	Fecha del Trabajo	Fotos Antes	Fotos Después	Descripción	Acciones
1	12 DE OCTUBRE TARDE	Plomería	2024-11-04			Reparación del tanque de agua	Editar Eliminar Ver
2	MODESTO OMISTE	Eléctrico	2024-11-05			Reinstalación del cableado eléctrico	Editar Eliminar Ver
3	SAN JOSE DE YUNGUYO B	Tingladistas	2024-10-30			Reparación de techo de laboratorio	Editar Eliminar Ver

Mostrando página 1 de 1

[Primero](#)
[Anterior](#)
[1](#)
[Siguiente](#)
[Último](#)

para agregar un nuevo trabajo de intervención por parte de las cuadrillas se tiene el siguiente formulario, donde el personal de cuadrillas podrá detallar la intervención realizada y adjuntar fotografías del trabajo realizado.

Crear Nuevo Trabajo

Seleccione Unidad Educativa
12 DE OCTUBRE TARDE - Turno: Tarde

Turno
Tarde

Distrito
Distrito 1

Cuadrilla
Canaletería

Fecha del Trabajo
19/11/2024

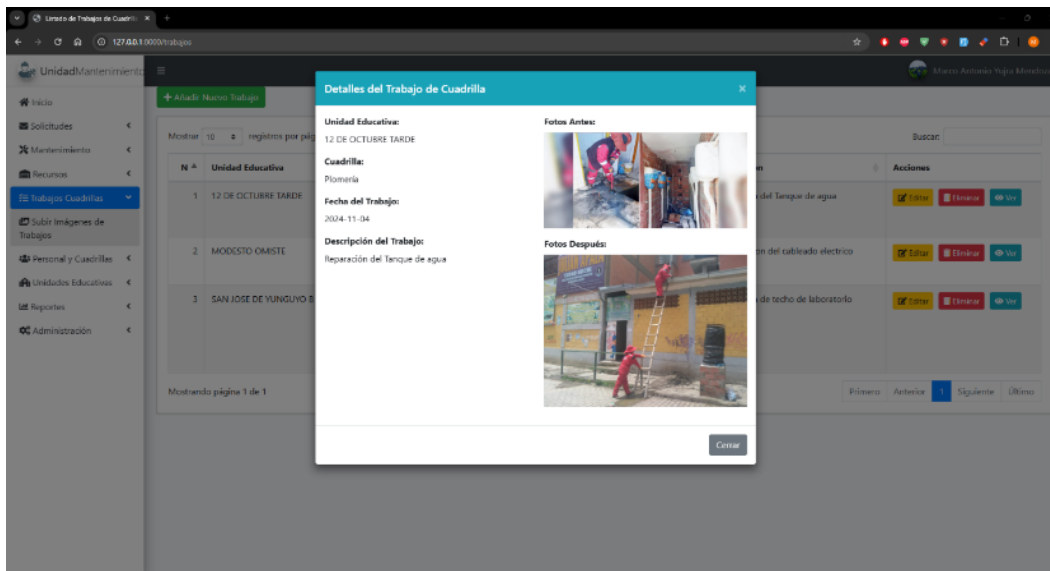
Fotos Antes
Seleccionar archivo 296670373_187380743661037_7271244755423281657_n.jpg

Fotos Después
Seleccionar archivo 295068951_185872960478482_6098524200947300737_n.jpg

Descripción del Trabajo
Reinstalación Eléctrica

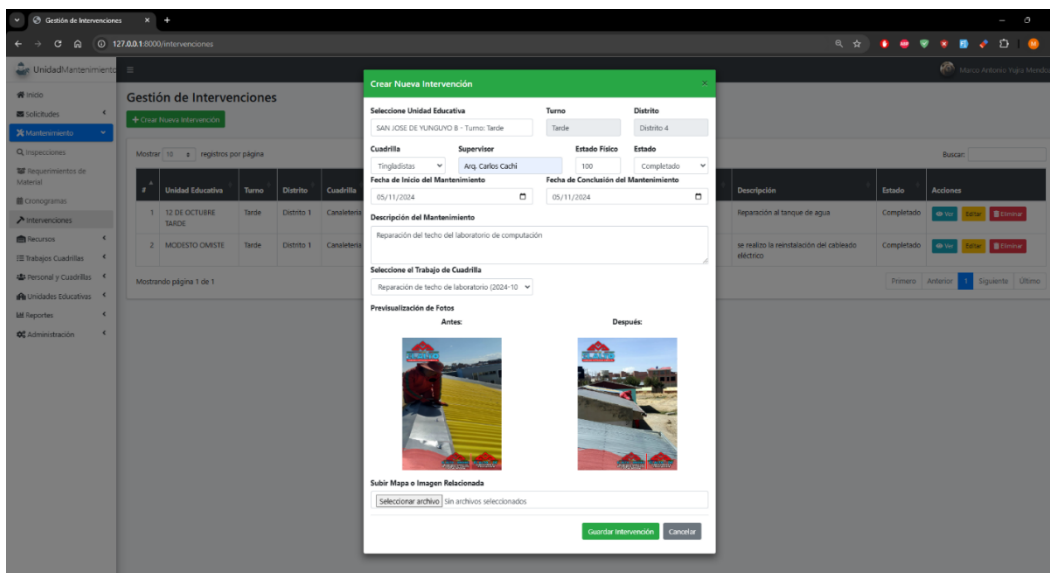
Cerrar Guardar

El personal encargado puede previsualizar las fotografías que adjunto además de los detalles del trabajo realizado



Módulo de Mantenimiento Sub módulo de Intervención

Una vez realizado el trabajo por parte del personal cuadrillas a la Unidad Educativa se realiza el registro de intervención asignando el trabajo del personal luego serán impresos por el inspector.



Se tiene el trabajo listado y registrado en el sub modal Intervención

A #	Unidad Educativa	Turno	Distrito	Cuadrilla	Supervisor	Estado Físico (%)	Fecha de Inicio	Fecha de Conclusión	Tiempo de Ejecución	Descripción	Estado	Acciones
1	12 DE OCTUBRE TARDE	Tarde	Distrito 1	Canaletaria	Arq. Carlos Cachi	100%	2024-11-01	2024-11-05	4 días	Reparación al tanque de agua	Completado	Ver Editar Eliminar
2	MODESTO OMISTE	Tarde	Distrito 1	Canaletaria	Arq. Carlos Cachi	100%	2024-10-30	2024-11-01	2 días	se realizo la reinstalación del cableado eléctrico	Completado	Ver Editar Eliminar
3	SAN JOSE DE YUNGUYO B	Tarde	Distrito 4	Tingladistas	Arq. Carlos Cachi	100%	2024-11-05	2024-11-05	0 días	Reparación del techo del laboratorio de computación	Completado	Ver Editar Eliminar

Módulo de Personal y Cuadrillas Sub módulo de Personal

Se realiza la gestión de todo el personal de la unidad, así como inspectores como personal de mano de obra.

Personal

+ Agregar Personal

Mostrar 10 registros por página

#	Nombre	Apellido Paterno	Apellido Materno	Cargo	Carnet	Correo	Celular	Dirección	Acciones
1	Amalia	Cafas	Vaca	plomero	39913796	Amalia@gmail.com	61643428	Avenida Murillo #1567	Editar Eliminar
2	Narciso	Valenzuela	Kuno	plomero	1232456780	Narciso@gmail.com	69550030	Calle La Paz #136	Editar Eliminar
3	Juana	Perez	Gomez	secretaria	64587410	juana@gmail.com	77771155	Urbanizacion Los Andes	Editar Eliminar
4	Juan	Algel	Royo	soldador	46307633	juan@gmail.com	76314410	Celda «Avenida Siempre Viva #3853	Editar Eliminar
5	Raul	Garcia	Marquez	albañil	69178225	Raul@gmail.com	77775025	Urbanización Mercedario	Editar Eliminar
6	Fernando	Pardo	Nogales	electricista	76011571	Fernando@gmail.com	77585536	Calle Mercado P2497	Editar Eliminar
7	Joan	Patiño	Royo	electricista	37237329	Joan@gmail.com	63250267	Avenida Buenos Aires #5318	Editar Eliminar
8	Pedro	Leon	Perez	electricista	64525410	pedro@gmail.com	78451212	Av. Puerta	Editar Eliminar
9	Ernesto	Cruz	Larico	albañil	2536146	Ernesto@gmail.com	77746505	Av. Puerta	Editar Eliminar
10	Maria	Carillo	Quispe	albañil	61117898	maria@gmail.com	77777775	Camino Laja	Editar Eliminar

Mostrando página 1 de 2

Primero Anterior 1 2 Siguiente Último

Para agregar a mas personal se tiene el siguiente formulario donde se registran los datos del personal de la unidad.

Añadir Nuevo Personal ✕

Nombre	Carnet
<input type="text" value="Nombre"/>	<input type="text" value="Número de Carnet"/>
Apellido Paterno	Celular
<input type="text" value="Ingrese su Apellido Paterno"/>	<input type="text" value="Número de Celular"/>
Apellido Materno	Dirección
<input type="text" value="Ingrese su Apellido Materno"/>	<input type="text" value="Dirección"/>
Cargo	
<input type="text" value="Albañil"/>	

Módulo de Personal y Cuadrillas Sub módulo de Cuadrillas

Se realiza la gestión del personal de mano de obra donde se le asigna las cuadrillas:

Gestión de Cuadrillas

+ Crear Nueva Cuadrilla

Mostrar: 10 registros por página

N° A	ID	Tipo de Cuadrilla	Personal Asignado	Cargo	Acciones
1	1	Canaletaria	Juan Algel Rojo	soldador	Editar Eliminar
2	2	Plomería	Amalia Cañas Vaca	plomero	Editar Eliminar
3	3	Electrico	Fernando Pardo Nogales	electricista	Editar Eliminar
4	4	Electrico	Federico Suazo Lanico	electricista	Editar Eliminar
5	5	Tingladista	Estrella Palacio Dueñas	soldador	Editar Eliminar
6	6	Plomería	Narciso Valenzuela Kuno	plomero	Editar Eliminar
7	7	Albañilería	Raul Garcia Marquez	albañil	Editar Eliminar
8	8	Albañilería	Fernando Ali Mendoza	albañil	Editar Eliminar
9	9	Albañilería	Aquiles Torres Perez	albañil	Editar Eliminar
10	10	Albañilería	Maria Carrillo Quispe	albañil	Editar Eliminar

Mostrando página 1 de 1

Primero Anterior 1 Siguiente Ultimo

Para registrar se tiene el siguiente formulario donde en base al cargo que tenga el personal se cargara los datos para darle más facilidad al usuario.

Registrar Cuadrilla

Tipo de Cuadrilla

Tingladista

Asignar Personal (opcional)

-- Seleccione Personal --

Alberto Quispe - soldador

Guardar Cuadrilla Cerrar

Electrico Pedro Ali Quinteros

Módulo de Unidades Educativas Sub módulo de Unidad Educativa

Se realiza la gestión todas las Unidades educativas de la Ciudad de El Alto.

N°	ID	Código S.I.E.	Unidad Educativa	Turno	Código Infraestructura	Organización	Distrito	Niveles	Dirección	Latitud	Longitud	Acciones
1	1	4073001	11 DE OCTUBRE (15-03)	Tarde	4073001	FEDEPAF	Distrito 1	Secundaria	ZONA SAN BENITO (11) WINGRE (15) N.UM. (15)	-16.5209473	-68.1829711	[Ver] [Editar]
2	4	4073010	OCELOSACOS DE BOLIVIA	Mañana	4073007	COOIDE	Distrito 1	Inicial-Primaria	PLAN 07 CALLE DANIEL BRACAMONTE S/N	-16.5209073	-68.1844338	[Ver] [Editar]
3	5	4073031	SAN ANTONIO	Mañana	4073073	FEDEPAF	Distrito 1	Inicial-Primaria-Secundaria	CALLE 16 AV. (15) N.UM. (15)	-16.5209473	-68.1829711	[Ver] [Editar]
4	3	4073004	INDUSTRIAL OESTE	Tarde	4073003	FEDEPAF	Distrito 1	Inicial-Primaria	PLAN DE CALLES 3 Y 4	-16.5209073	-68.1844338	[Ver] [Editar]
5	3	4073031	ELA MANUEL MORAÑO MARABIA	Mañana	4073073	FEDEPAF	Distrito 2	Inicial-Primaria	ZONA SAN BENITO (11) WINGRE (15) N.UM. (15)	-16.5209473	-68.1829711	[Ver] [Editar]
6	7	4073031	13 DE ABRIL (15-03)	Tarde	4073073	FEDEPAF	Distrito 2	Secundaria	AV. SAN BENITO (11) WINGRE (15) N.UM. (15)	-16.5209473	-68.1829711	[Ver] [Editar]
7	5	4073003	VILLA VICTORIA B	Tarde	4073004	FEDEPAF	Distrito 1	Secundaria	GUAYO A. OTERO Y ACIDOS ARBUENAS N.UM. (15)	-16.5209073	-68.1844338	[Ver] [Editar]
8	9	4073009	13 DE ABRIL (15-03)	Tarde	4073073	FEDEPAF	Distrito 2	Inicial-Primaria	ZONA SAN BENITO (11) WINGRE (15) N.UM. (15)	-16.5209473	-68.1829711	[Ver] [Editar]
9	10	4073004	LUIS ESPINAL DE COLLAPAN	Mañana	4073003	COOIDE	Distrito 3	Inicial-Primaria	CALLE 9 CINCA EN COEM 03 79 COLLAPAN	-16.5233053	-68.2233053	[Ver] [Editar]
10	11	4073031	ZONA SAN BENITO (15-03)	Tarde	4073073	FEDEPAF	Distrito 1	Secundaria	ZONA SAN BENITO (11) WINGRE (15) N.UM. (15)	-16.5209473	-68.1829711	[Ver] [Editar]

Donde se tiene los siguientes campos y formulario para agregar una nueva unidad educativa:

Agregar Unidad Educativa

Código S.I.E. !

 ✓

Distrito !

 ✓

Nombre de la Unidad Educativa !

 ✓

Niveles !

 ✓

Turno !

 ✓

Dirección !

 ✓

Código Infraestructura !

 ✓

Latitud !

 ✓

Organización !

 ✓

Longitud !

 ✓

Módulo de Reportes Sub módulo de Acta de Inspección

Se realiza la generación del documento de inspección y requerimiento de material ya realizados de la unidad educativa.

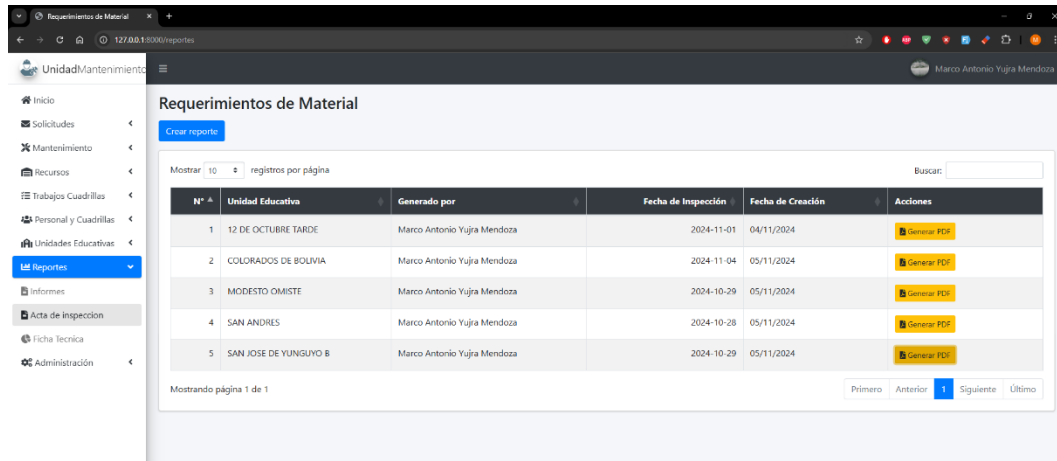
The screenshot shows a web application interface for 'Requerimientos de Material'. On the left is a navigation menu with options like 'Inicio', 'Solicitudes', 'Mantenimiento', 'Recursos', 'Trabajos Cuadrillas', 'Personal y Cuadrillas', 'Unidades Educativas', 'Reportes', 'Informes', 'Acta de Inspección', 'Ficha Técnica', and 'Administración'. The main content area is titled 'Requerimientos de Material' and features a 'Crear reporte' button. Below this is a table with columns: 'N°', 'Unidad Educativa', 'Generado por', 'Fecha de Inspección', 'Fecha de Creación', and 'Acciones'. The table contains four rows of data. At the bottom, there are pagination controls showing 'Mostrando página 1 de 1' and buttons for 'Primero', 'Anterior', 'Siguiente', and 'Último'.

N°	Unidad Educativa	Generado por	Fecha de Inspección	Fecha de Creación	Acciones
1	12 DE OCTUBRE TARDE	Marco Antonio Yujra Mendoza	2024-11-01	04/11/2024	Generar PDF
2	COLORADOS DE BOLIVIA	Marco Antonio Yujra Mendoza	2024-11-04	05/11/2024	Generar PDF
3	MODESTO OMISTE	Marco Antonio Yujra Mendoza	2024-10-29	05/11/2024	Generar PDF
4	SAN ANDRES	Marco Antonio Yujra Mendoza	2024-10-28	05/11/2024	Generar PDF

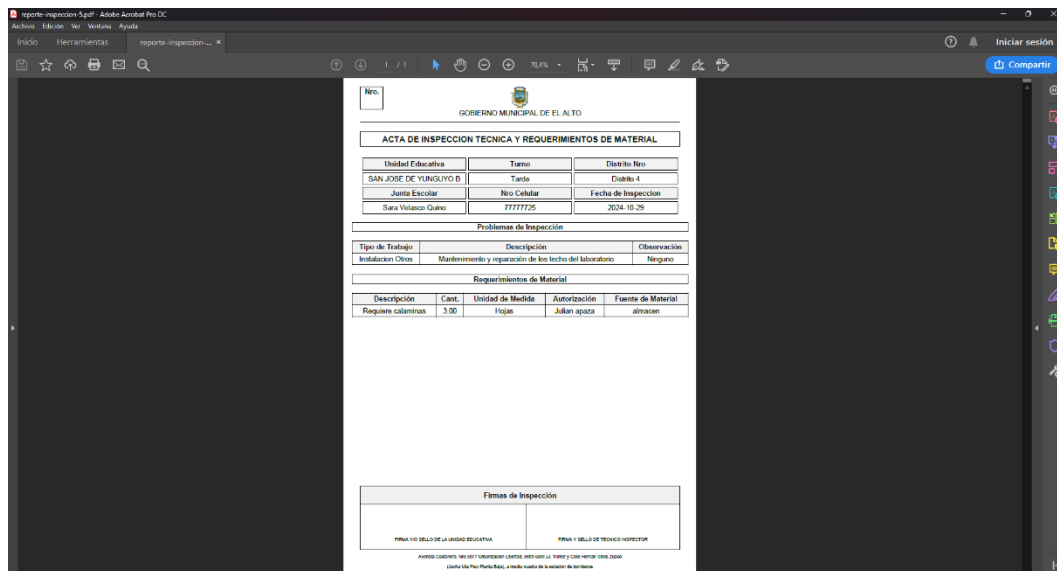
Se genera la inspección y el Requerimiento del Material Ingresando el nombre de la Unidad Educativa y seleccionado la fecha de inspección como aparece en la figura:

This screenshot shows the same web application interface as the first image, but with a modal dialog box titled 'Crear Reporte' open in the center. The dialog has two sections: 'Seleccione Unidad Educativa' with a dropdown menu showing 'SAN JOSE DE YUNGUYO B' and 'Crear' button, and 'Seleccione Inspección' with a date picker showing '2024-10-29' and a 'Generar Reporte' button. The background content is dimmed.

Previamente después se lista el reporte y el inspector deberá de generar el reporte a través del botón Generar Reporte

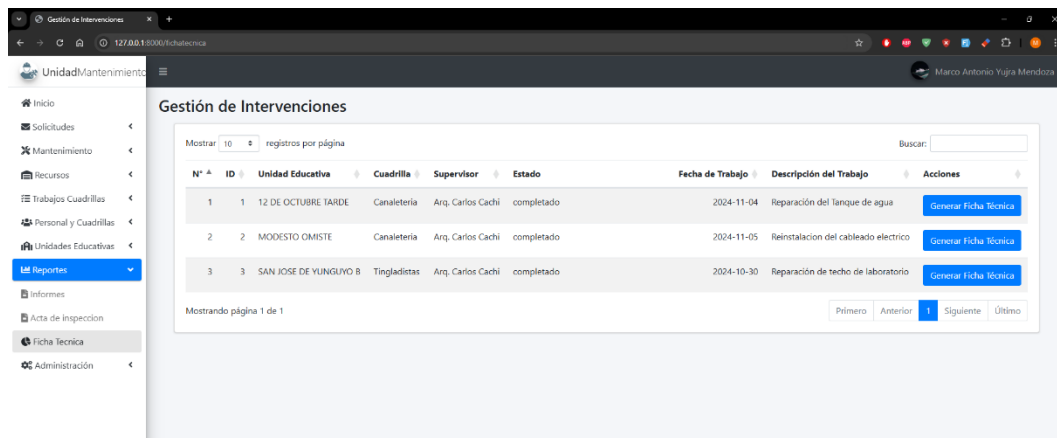


El reporte es generado en PDF con los problemas encontrados y requerimientos de material.



Módulo de Reportes Sub módulo de Acta de Ficha Técnica

Se genera la intervención previamente registrada por las cuadrillas, adjuntando el trabajo realizado para crear un reporte visual y detallado de la actividad.



Mostrar 10 registros por página

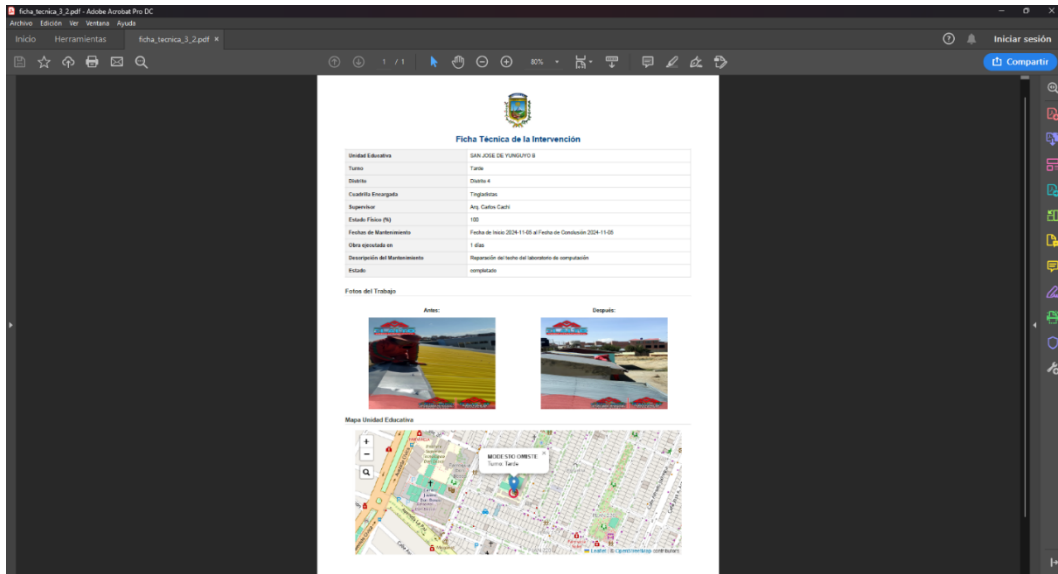
Buscar:

N°	ID	Unidad Educativa	Cuadrilla	Supervisor	Estado	Fecha de Trabajo	Descripción del Trabajo	Acciones
1	1	12 DE OCTUBRE TARDE	Canaletaria	Arq. Carlos Cachi	completado	2024-11-04	Reparación del tanque de agua	Generar Ficha Técnica
2	2	MODESTO OMISTE	Canaletaria	Arq. Carlos Cachi	completado	2024-11-05	Reinstalación del cableado eléctrico	Generar Ficha Técnica
3	3	SAN JOSE DE YUNGLUVO B	Tingladietas	Arq. Carlos Cachi	completado	2024-10-30	Reparación de techo de laboratorio	Generar Ficha Técnica

Mostrando página 1 de 1

Primero Anterior 1 Siguiente Último

Se genera el documento en PDF de la intervención realizada mostrándose de la siguiente manera:



Teniendo la conclusión del trabajo detallado para el respaldo de la unidad.