

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



PROYECTO DE GRADO
SISTEMA WEB PARA EL CONTROL DE CONDUCTORES Y
VEHÍCULOS DEL TRANSPORTE ESCOLAR
CASO: DIVISIÓN DE SERVICIOS PÚBLICOS DEPENDIENTES DE
LA DIRECCIÓN DEPARTAMENTAL DE TRÁNSITO, TRANSPORTE
Y SEGURIDAD VIAL SANTA CRUZ

PARA OPTAR EL GRADODE LICENCIATURA EN INFORMÁTICA
MENCIÓN INGENIERÍA DE SISTEMAS INFORMÁTICOS

POR: ROSARIO VERONICA QUISPE ALANOCA

TUTOR : M.SC. ALDO VALDEZ ALVARADO

LA PAZ – BOLIVIA

2022



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES, PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRÍCTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) Visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) Copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) Copiar textualmente parte(s) de su contenido mencionado la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARÁ EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

Dedicatoria

El presente proyecto de grado lo dedico a mi familia...

AGRADECIMIENTO

A Dios por su infinita gracia para con mi familia y amigos.

A mi Padre Victoriano Quispe por ser mi amigo en quien puedo contar siempre con su apoyo.

A mi madre Inocencia Alanoca de Quispe por su paciencia, cariño y apoyo incondicional en cada etapa de mi vida, mi más hermosa motivación para con mis objetivos.

A mis hermanos Angel, Julieta, Andrea, Nieves, Rolando, por su apoyo moral, enseñanza y ejemplo de superación.

A mi Tutor Metodológico M.Sc. Aldo Valdez Alvarado por la enseñanza, tiempo y dedicación brindada hacia mi persona en cada una de las materias que me tocó cursar con su persona.

A mi asesor M.Sc. Aldo Valdez Alvarado por la buena labor de dirigir y trabajar en equipo para con el proyecto y demás materias que me tocó compartir con su persona.

A la Dirección Departamental de Tránsito Santa Cruz. Sgto. Ángel Franklin Quispe quienes brindaron su apoyo incondicional para con el proyecto.

A mis amigos y compañeros de estudio por los buenos momentos compartidos dentro de mi formación profesional.

ÍNDICE GENERAL

1. CAPÍTULO I MARCO REFERENCIAL	1
1.1 INTRODUCCIÓN	1
1.2 ANTECEDENTES	2
1.2.1 ANTECEDENTES INSTITUCIONALES	2
1.2.2 ANTECEDENTES DE PROYECTOS SIMILARES	4
1.3 PLANTEAMIENTO DEL PROBLEMA	5
1.3.1 PROBLEMA CENTRAL	5
1.3.2 PROBLEMAS SECUNDARIOS	5
1.4 DEFINICIÓN DE OBJETIVOS	6
1.4.1 OBJETIVO GENERAL	6
1.4.2 OBJETIVOS ESPECÍFICOS	6
1.5 JUSTIFICACIÓN	7
1.5.1 JUSTIFICACIÓN ECONÓMICA	7
1.5.2 JUSTIFICACIÓN SOCIAL	7
1.5.3 JUSTIFICACIÓN TECNOLÓGICA	7
1.6 ALCANCES Y LÍMITES	8
1.6.1 ALCANCES	8
1.6.2 LÍMITES	8
1.7 APORTES	9
1.7.1 PRÁCTICO	9
1.7.2 TEÓRICO	9
1.8 METODOLOGÍA	9
1.8.1 METODOLOGÍA CIENTÍFICA	9
1.8.2 METODOLOGÍA DE DESARROLLO	10
2. CAPÍTULO II MARCO TEÓRICO	10
2.1 INTRODUCCIÓN	10
2.2 MARCO INSTITUCIONAL	11
2.2.1 ASPECTOS GENERALES	11
2.3 INGENIERÍA DE SOFTWARE	12

2.3.1	EL PROCESO DEL SOFTWARE.....	12
2.3.2	MODELADO	13
2.4	METODOLOGÍA DE DESARROLLO ÁGIL SCRUM	13
2.4.1	COMPONENTES DEL SCRUM.....	13
2.4.2	ELEMENTOS DEL SCRUM.....	15
2.4.3	DESARROLLO DE LAS FASES DE UN PROYECTO EN SCRUM.....	18
2.4.4	PRUEBAS DE SOFTWARE	19
2.5	MODELO DE DESARROLLO UWE BASADO EN UML.....	21
2.5.1	MODELO DE REQUISITOS.....	21
2.5.2	MODELO DE CONTENIDO	22
2.5.3	MODELO DE NAVEGACIÓN.....	23
2.5.4	MODELO DE PRESENTACIÓN	24
2.5.5	MODELO DE PROCESO.....	25
2.6	PATRON ARQUITECTONICO MVC	26
2.7	HERRAMIENTAS TECNOLÓGICAS.....	27
2.7.1	BACKEND.....	28
2.7.2	FRONTEND.....	28
3.	CAPITULO III MARCO APLICATIVO	29
3.1	INTRODUCCIÓN	29
3.2	PRE-GAME	30
3.2.1	PLANEAMIENTO Y ANÁLISIS.....	30
3.2.2	ASIGNACIÓN DE ROLES.....	31
3.2.3	HISTORIAS DE USUARIO.....	32
3.2.4	PRODUCT BACKLOG	36
3.2.5	DIAGRAMA DE CASOS DE USO.....	38
3.2.6	DIAGRAMA DE DATOS	38
3.3	GAME.....	41
3.3.1	SPRINT 1.....	41
3.3.2	SPRINT 2.....	45
3.3.3	SPRINT 3.....	49

3.4	POST GAME	54
3.4.1	INTERFAZ DE USUARIO	54
3.4.2	PRUEBAS DE STRESS	58
4.	CAPITULO IV CALIDAD Y SEGURIDAD DE SOFTWARE	63
4.1	INTRODUCCIÓN	63
4.2	NORMA ISO/IEC 25010	63
4.3	GQM	65
4.4	APLICACIÓN DE LA NORMA ISO/IEC 25010 Y GQM	65
4.4.1	USABILIDAD.....	66
4.4.2	SEGURIDAD.....	73
4.4.3	PORTABILIDAD	79
4.4.4	COMPATIBILIDAD	81
4.4.5	MANTENIBILIDAD.....	83
4.4.6	FIABILIDAD	85
4.4.7	EFICIENCIA DE DESEMPEÑO	88
4.4.8	ADECUACIÓN FUNCIONAL.....	90
5.	CAPITULO V ESTIMACIÓN DE COSTO Y BENEFICIO	93
5.1	INTRODUCCIÓN	93
5.2	COCOMO II.....	93
5.2.1	ESTIMACIÓN DEL ESFUERZO	94
5.2.2	ESTIMACIÓN DEL CRONOGRAMA	95
5.2.3	MÉTRICAS DEL SOFTWARE	95
5.3	APLICACIÓN DE COCOMO II	100
5.3.1	ESTIMACIÓN DEL ESFUERZO	100
5.3.2	COSTO DE SOFTWARE	104
5.4	INDICADORES	105
5.4.1	VALOR ACTUAL NETO – VAN.....	105
5.4.2	TASA INTERNA DE RETORNO – TIR	106
5.4.3	COSTO BENEFICIO - C/B	106
6.	CAPITULO VI CONCLUSIONES Y RECOMENDACIONES.....	107

6.1 CONCLUSIONES	107
6.2 RECOMENDACIONES.....	108
BIBLIOGRAFÍA.....	108
ANEXOS.....	112

ÍNDICE DE FIGURAS

Figura 1.1 Organigrama de la Dirección Departamental de Tránsito, Transporte y Seguridad Vial del departamento de Santa Cruz	3
Figura 2.1 Roles, artefactos y eventos principales del SCRUM.....	14
Figura 2.2 Las reuniones habituales del SCRUM.....	15
Figura 2.3 Tablero de SCRUM.....	17
Figura 2.4 Prueba de software	20
Figura 2.5 Caja Negra	21
Fuente: Elaboración propia	21
Figura 2.6 Diagrama de Casos de Uso.....	22
Figura 2.7 Diagrama de Clases	22
Figura 2.8 Diagrama de Navegación	23
Figura 2.9 Diagrama de Navegación con menú	24
Figura 2.10 Estereotipos de Presentación.....	24
Figura 2.11 Diagrama de Presentación	25
Figura 2.12 Diagrama de Proceso	26
Figura 2.13 Fundamentos del patrón Modelo Vista Controlador.....	27
Figura 3.1. Trabajando UWE con SCRUM	29
Figura 3.2. Desarrollo de SCRUM y UWE	30
Figura 3.3. Diagrama de Casos de Uso.....	38
Figura 3.4 Modelo Entidad Relación	39
Figura 3.5 Modelo Relacional.....	40
Figura 3.6 Modelo de Requerimientos.....	42
Figura 3.7 Modelo de Contenido.....	43
Figura 3.8 Modelo de Navegación	43
Figura 3.9 Modelo de Presentación para el Login.....	44
Figura 3.10 Modelo de Presentación – Menú Principal.....	44
Figura 3.11 Modelo de Requerimientos.....	47
Figura 3.12 Modelo de Contenido.....	47
Figura 3.13 Modelo de Navegación	48

Figura 3.14 Modelo de presentación.....	48
Figura 3.15 Modelo de Requerimientos.....	51
Figura 3.16 Modelo de Contenido.....	52
Figura 3.17 Modelo de Navegación.....	52
Figura 3.18 Modelo de Presentación.....	53
Figura 3.19 Formulario de inicio de sesión.....	54
Figura 3.20 Menú Principal.....	55
Figura 3.21 Formulario de Registro de Conductor.....	55
Figura 3.22 Formulario de Registro de Vehículo.....	56
Figura 3.23 Lista de Conductores.....	56
Figura 3.24 Tarjeta TICE generada.....	57
Figura 3.25 Reporte de Conductores.....	58
Figura 3.26 Reportes estadísticos.....	58
Figura 3.27. Datos transferidos, memoria del sistema y carga de la CPU.....	59
Figura 3.28. Solicitud de apertura y transferencia de datos.....	60
Figura 3.29 Peticiones del usuario.....	61
Figura 3.30 Capacidad de carga del sistema.....	62
Figura 1 Reunión de Seguimiento con solicitante.....	112
Figura 2 Acceso al Sistema.....	112
Figura 3 Registro de Conductor.....	113
Figura 4 Registro de Vehículo.....	113
Figura 5 Tarjeta TICE.....	114
Figura 6 Lista de Conductores.....	114
Figura 7 Estadísticas.....	115
Figura 8 Reportes.....	115

INDICE DE TABLAS

Tabla 2.1 Ejemplo de Pila de Producto.....	16
Tabla 2.2 Ejemplo de historia de usuario.....	16
Tabla 3.1 Tipos de usuario.....	31
Tabla 3.2 Roles del SCRUM	32
Tabla 3.3 Gestión de usuario	32
Tabla 3.4 Registro de conductores, vehículos y establecimiento educativo.....	33
Tabla 3.5 CRUD de los registros de conductores y vehículos	33
Tabla 3.6 Digitalizar los requisitos de afiliación	34
Tabla 3.7 Tarjeta de identificación del conductor escolar	34
Tabla 3.8 Registro de una nueva gestión	34
Tabla 3.9 Reporte de los conductores y las tarjetas tice.....	35
Tabla 3.10 Buscar y mostrar a los conductores	35
Tabla 3.11 Registro de antecedentes	36
Tabla 3.13 Datos estadísticos.....	36
Tabla 3.14 <i>Product Backlog</i>	37
Tabla 3.15 Sprint 1 Módulo de gestión de usuario	41
Tabla 3.16 Planificación del Sprint 1	42
Tabla 3.17 Pruebas y resultados obtenidos del Sprint 1	45
Tabla 3.18 Modulo de Gestión de los conductores y vehículos.....	45
Tabla 3.19 Planificación del Sprint 2.....	46
Tabla 3.20 Pruebas y resultados obtenidos del Sprint 2	49
Tabla 3.21 Módulo tarjeta de identificación	49
Tabla 3.22 Planificación del Sprint 3.....	51
Tabla 3.23 Pruebas y resultados obtenidos del Sprint 3	54
Tabla 4.1 Características y subcaracterísticas definidas en ISO/IEC 25010.....	65
Tabla 4.2 Número de usuarios encuestados.....	66
Tabla 4.3 Criterios de decisión.....	66
Tabla 4.4 Preguntas y respuestas de usabilidad.....	69
Tabla 4.5 Criterios de evaluación para la usabilidad.....	72

Tabla 4.6 Criterios de decisión para usabilidad	73
Tabla 4.7 Preguntas y respuestas para seguridad	76
Tabla 4.8 Criterios de evaluación para la seguridad	78
Tabla 4.9 Criterios de decisión para la seguridad	78
Tabla 4.10 Generación de preguntas y respuestas para portabilidad	80
Tabla 4.11 Criterios de evaluación para portabilidad	80
Tabla 4.12 Criterios de decisión para portabilidad	81
Tabla 4.13 Generación de preguntas y respuestas para la compatibilidad	82
Tabla 4.14 Criterios de evaluación para la compatibilidad	83
Tabla 4.15 Criterios de decisión para la compatibilidad	83
Tabla 4.16 Generación de preguntas y respuestas para la mantenibilidad	84
Tabla 4.17 Criterios de evaluación para la mantenibilidad	85
Tabla 4.18 Criterios de decisión para la mantenibilidad	85
Tabla 4.19 Generación de preguntas y respuestas para la fiabilidad	87
Tabla 4.20 Criterios de evaluación para la fiabilidad	87
Tabla 4.21 Criterios de decisión para la fiabilidad	88
Tabla 4.22 Generación de preguntas y respuestas para la eficiencia de desempeño	89
Tabla 4.23 Criterios de evaluación para la eficiencia de desempeño	90
Tabla 4.24 Criterios de decisión para la eficiencia de desempeño	90
Tabla 4.25 Generación de preguntas y respuestas para la adecuación funcional	91
Tabla 4.26 Criterios de evaluación para la adecuación funcional	92
Tabla 4.27 Criterios de decisión para la adecuación funcional	92
Tabla 5.1 Productividad para el modelo Composición de aplicación	95
Tabla 5.2 Esquema de clasificación de puntos objetos	96
Tabla 5.3 Peso de un punto objeto	96
Tabla 5.4 Determinación del peso	98
Tabla 5.5 Peso del factor de complejidad	98
Tabla 5.6 Conversión de UFP a SLOC	99
Tabla 5.7 Ecuaciones del modelo básico de COCOMO	100
Tabla 5.8 FactorLCD	101

Tabla 5.9 UFP.....	102
Tabla 5.10 Factor de complejidad.....	102
Tabla 5.11 Costo total del proyecto	104

1. CAPÍTULO I MARCO REFERENCIAL

1.1 INTRODUCCIÓN

La principal misión que protege y persigue a través de la CPE es la vida humana, y la Policía Boliviana como Fuerza Pública es que asume esta gran misión velando por este bien jurídico como es la vida y mucho más si tomamos en cuenta nuestra niñez y juventud en edad escolar es la parte más sensible y vulnerable de nuestra población. Según datos estadísticos emitidos por el observatorio nacional de seguridad ciudadana, sobre los delitos vinculados al uso del servicio de transporte automotor público terrestre, llegando a generar una sensación de inseguridad a los usuarios de este servicio, es por ello la necesidad de introducir políticas públicas que permitan reducir la comisión de este tipo de hechos delictivos El Servicio de Transporte Escolar puede ser público o privado, los estudiantes serán transportados desde el establecimiento educativo hasta sus hogares, en algunos casos se establecen líneas especiales en los puntos de parada preestablecidos todo esto en un cierto tiempo.

Las actividades escolares generan una gran preocupación a los padres de familia en la cuestión de transporte, debido a los complicados horarios de trabajo, por ello algunos padres optan por contratar un transporte escolar es por esta situación que la Dirección Departamental de Tránsito, Transporte y Seguridad Vial, requiere contar con un control y supervisión sobre los diferentes medios del transporte público que se prestan en nuestra ciudad, siendo uno de ellos el transporte escolar, el cual amerita una preocupación especial, por lo tanto en cumplimiento a las normativas vigentes se desea precautelar la seguridad a los niños y jóvenes así como a los mismos conductores. (Quispe, Rocha, Monasterio y Pereira, 2020).

Por tal motivo mediante la implementación de la metodología ágil SCRUM y el lenguaje unificado de modelado UML que nos permite visualizar, construir y documentar, el presente proyecto de grado propone implementar un sistema web para el control de conductores y vehículos del transporte escolar que ayude a automatizar los registros, reportes, búsqueda manual, generación de la Tarjeta de Identificación del Conductor Escolar (T.I.C.E.), mostrar datos estadísticos y administrar el sistema.

1.2 ANTECEDENTES

1.2.1 ANTECEDENTES INSTITUCIONALES

Según Quispe, Rocha, Monasterio y Pereira (2020) El transporte escolar es un servicio público de transporte para los alumnos de las diferentes unidades educativas, sean públicos o privados, de buses, góndolas, minibuses, taxis y radios taxis, debiendo estar equipados con los medios necesarios y seguros para poder prestar este servicio.

A través de diferentes medios de comunicación, denuncias, se han tomados conocimientos de diferentes casos de inseguridad, conducción peligrosa porque los choferes consumen bebidas alcohólicas en los motorizados, fallas mecánicas, por lo cual es que la Policía Boliviana a través de la Dirección Departamental de Tránsito, Transporte y Seguridad Vial por medio de la División de Servicios Públicos, tiene la misión y responsabilidad de ejercer un control y supervisión sobre este tipo de servicio público, que es el transporte escolar.

Este servicio público llega para coadyuvar o colabora la responsabilidad de los padres de familia, ya que se toma más difícil todavía debido a los complicados horarios de trabajo de muchos papás y mamás.

Los padres de familia deberán tener un especial cuidado en la elección de este tipo de servicio público, ya que colocan en manos ajenas la seguridad, resguardo y cuidado de sus hijos al momento de traslado a sus centros educacionales. La seguridad es un aspecto muy importante, y más aún cuando se trata de niños y jóvenes que son los usuarios que utilizan este tipo de servicio, en el transporte escolar, la seguridad en las vías deberá estar garantizada, ya que los escolares contarán en todo momento con un profesional de conducción de los escolares si fueran necesario.

Con carácter general la prestación del servicio de transporte escolar durante una gestión, será necesario que los interesados soliciten la autorización correspondiente en la dirección Departamental de tránsito, transporte y seguridad vial a través de la división servicios públicos.

Esta división no cuenta con un sistema de control para los conductores y vehículos, registran

la información en hojas de cálculo, realiza búsqueda manual, se tiene extravío de documentos, no se tiene datos actualizados u otros.

Organigrama de la Dirección Departamental de Tránsito, Transporte y Seguridad Vial del departamento de Santa Cruz

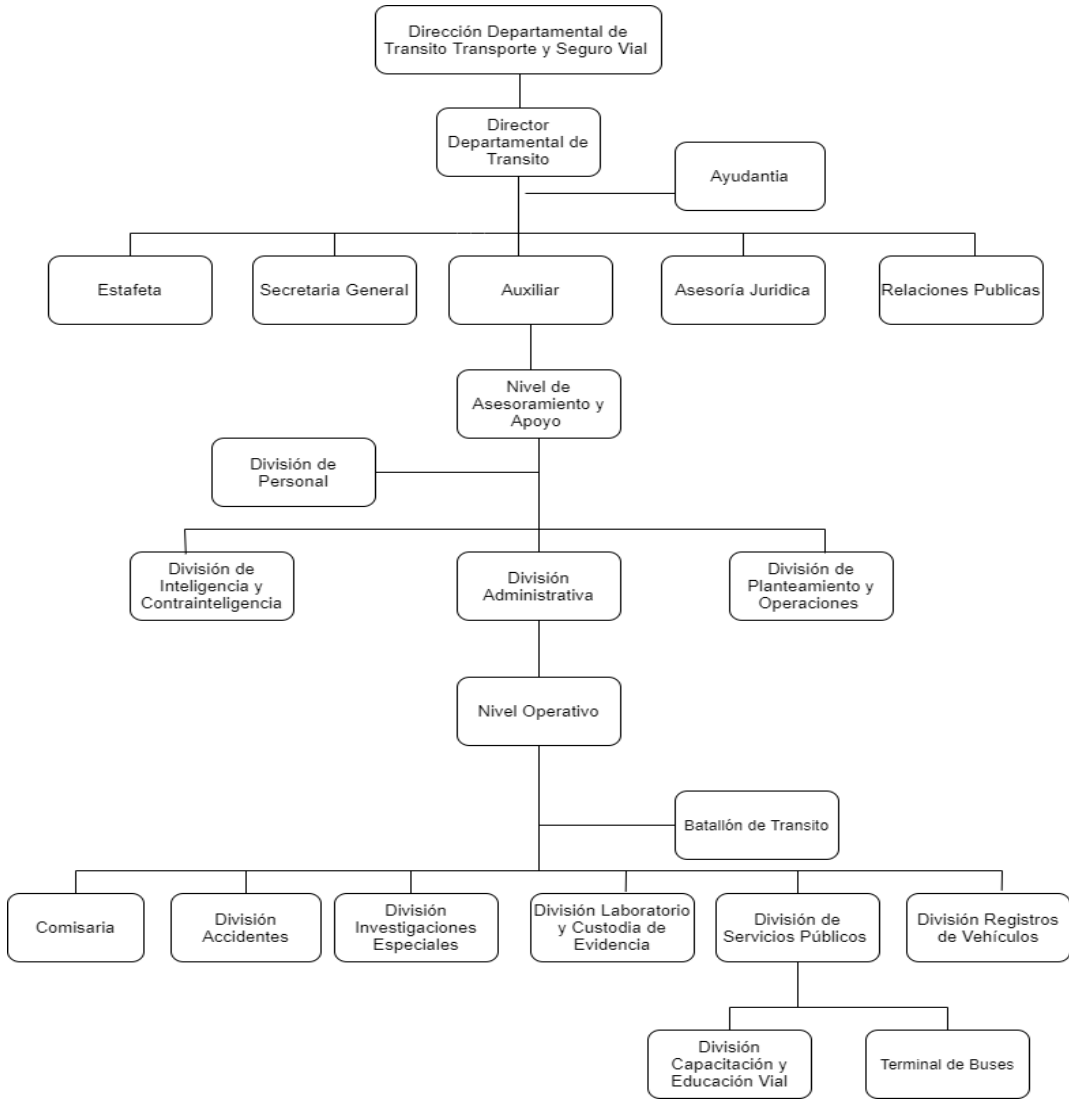


Figura 1.1 Organigrama de la Dirección Departamental de Tránsito, Transporte y Seguridad Vial del departamento de Santa Cruz
Fuente: Elaboración propia

Actualmente no existe un sistema o plataforma de gestión académica dentro la unidad, esto

conlleva a realizar procesos manuales, demora en la búsqueda de información, largas filas de alumnos con diferentes consultas, demora en la actualización de las notas, falta de coordinación en la planificación y elaboración de los cronogramas.

1.2.2 ANTECEDENTES DE PROYECTOS SIMILARES

En la Universidad Peruana Austral del Cusco se tiene un proyecto para generar matrículas, para nuestro proyecto vamos a generar una tarjeta TICE con características similares, pero datos diferentes.

- Sistema de Matrícula y Consulta de Notas para la Universidad Peruana Austral del Cusco (SMCN-UPAC), Kenny Ángel Merino Covarrubias, 2016, sistema de generación de matrículas para la universidad, utilizó la metodología de Proceso Unificado de Desarrollo de Software (PUDS). Herramientas de desarrollo PHP, MySQL y Apache.

En la Carrera de Informática de la Facultad de Ciencias Puras y Naturales de la Universidad Mayor de San Andrés se puede evidenciar varios proyectos similares, pero con características diferentes dependiendo de la institución los cuales son:

- Sistema Municipal de infracciones de movilidad y transporte caso: Gobierno Autónomo Municipal de La Paz, Autor: Oscar Ronny Conde Saico, 2017. Control del Servicio Público de Transporte los cuales son realizados por los miembros de la Guardia Municipal de imponiendo sanciones y aplicando las medidas precautorias, se utiliza la metodología ágil SCRUM, los *framework* AngularJS y NodeJS, como gestores de base de datos utilizamos PostgreSQL y MongoDB
- Sistema de Administración Municipal de la Movilidad (SAMM) del Gobierno Autónomo Municipal de La Paz caso: Gobierno Autónomo Municipal de La Paz - GAMLP, Autor: Mariela Condori Castañeta, 2016, el presente proyecto administra las rutas, sindicatos y choferes. aplicando sanciones si así lo amerita se utilizó la metodología Ágil AUP (*Agile Unified Process*), fue desarrollado en AngularJS como lenguaje de programación y como gestor de base de datos POSTGRESQL Basada en la calidad estándar ISO 9126

1.3 PLANTEAMIENTO DEL PROBLEMA

La Policía Boliviana como fuerza pública e institución fundamental del Estado es quien asume la principal misión que protege y persigue a través de la C.P.E. es la vida humana. La División de Servicio Público dependiente de la Dirección Departamental de Tránsito, transporte y Seguridad Vial Santa Cruz, no cuenta con un sistema de control sobre los diferentes medios del servicio público que se prestan en nuestra ciudad, siendo uno de ellos el transporte escolar, el cual amerita una preocupación especial, por lo tanto, en cumplimiento a las normativas legales lograr brindar la seguridad a los niños y jóvenes, así como a los mismos conductores. Según las normativas de la policía, todo conductor y vehículo que presta el servicio de transporte escolar debe registrarse en la división de servicio público cumpliendo los requisitos y obtener una tarjeta de identificación.

La sección de empadronamiento realiza el procedimiento de registro (afiliar) a los conductores y vehículos, extendiendo una tarjeta de identificación, todo este proceso se lo realiza en hojas de cálculo, formularios manuales. En la división de servicios públicos se realizan informes, reportes, datos estadísticos haciendo una búsqueda manual de los archivos. En casos de cualquier conducta o infracción por parte del conductor la policía realiza la sanción correspondiente anotando los datos del conductor en su libreta ya que no cuenta con un sistema este sector. (Quispe, Rocha, Monasterio y Pereira, comunicación personal, abril de 2020).

1.3.1 PROBLEMA CENTRAL

¿Cómo gestionar la información de conductores, vehículos tanto del servicio público y particular que ejercen de transporte escolar en la ciudad de Santa Cruz?

1.3.2 PROBLEMAS SECUNDARIOS

- La sección de empadronamiento no tiene el registro de conductores y vehículos del transporte escolar, ocasionando una falsificación de datos
- No se tiene la implementación de la Tarjeta de Identificación del Conductor Escolar (T.I.C.E.), ocasionando una demora en los operativos policiales

- En casos de cualquier conducta, que el chofer consuma bebidas alcohólicas dentro el vehículo, o que cometa infracciones durante su recorrido la policía o ciudadanía no tiene un sistema disponible para verificar los antecedentes de estos conductores dentro del servicio de transporte escolar.
- Existen muchos conductores que realizan la circulación de sus vehículos y servicio aun estando estos con sanciones o no registrados de manera legal.
- La elaboración de informes, reportes, datos estadísticos u otros requerimientos de la institución se las realiza de manera manual, ocasionando pérdida de tiempo y demora en la búsqueda de datos.

1.4 DEFINICIÓN DE OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar un sistema web para gestionar la información de los conductores y vehículos tanto del servicio público y particular que ejercen de transporte escolar en la ciudad de Santa Cruz.

1.4.2 OBJETIVOS ESPECÍFICOS

- Registrar a los conductores y vehículos que prestan el servicio de transporte escolar
- Generar una Tarjeta de Identificación del Conductor Escolar (T.I.C.E.) para los conductores y para los operativos que realiza la policía boliviana
- Agilizar la verificación de antecedentes de los conductores.
- Verificar el estado de los conductores si se encuentran activos ya que la renovación de la tarjeta deben realizarlo anualmente.
- Elaborar informes, reportes, datos estadísticos y otros de la institución de manera sistemática con opciones de filtro y búsqueda además de su descarga y posterior impresión.

1.5 JUSTIFICACIÓN

1.5.1 JUSTIFICACIÓN ECONÓMICA

El sistema está desarrollado con herramientas de software libre, reemplaza los procesos manuales a un sistema automatizado, agiliza el tiempo de los trámites además que la institución cuenta con equipos computacionales necesarios, cableado de red, acceso a internet para la implementación del proyecto.

De esta manera se garantiza el buen funcionamiento y desarrollo del software teniendo en cuenta todos los puntos básicos fundamentados que se seguirá para implementar en la institución.

1.5.2 JUSTIFICACIÓN SOCIAL

Con la implementación del presente proyecto se beneficia la unidad de servicio público dependiente de la Direcciones Departamental de Tránsito, Transporte y Seguridad Vial centralizando la información, búsqueda rápida, los conductores que prestan el servicio escolar teniendo una identificación, la ciudadanía al agilizar los trámites de denuncia.

Registrando, validando y generando la tarjeta de identificación automática, haciendo seguimiento a las denuncias de algún conductor por parte de los padres de familia. Se benefician los conductores agilizando los trámites de filiación. También se benefician los padres de familia al poder verificar con el código en el sistema, que el conductor está registrado en la unidad de servicios públicos.

1.5.3 JUSTIFICACIÓN TECNOLÓGICA

La unidad de servicio público dependiente de la Direcciones Departamental de Tránsito, Transporte y Seguridad Vial cuenta con equipos computacionales, cableado de red y acceso a internet que es fundamental para la implementación del sistema.

Para el desarrollo se utilizará herramientas como *TypeScript* como lenguaje de programación, angular como marco de trabajo web para aplicaciones y sitios web de tipo SPA, Bootstrap y angular material como marco progresivo para construir interfaces de usuario, apache para el

servidor y MySQL para el gestor de base de datos.

1.6 ALCANCES Y LÍMITES

1.6.1 ALCANCES

Para el presente proyecto se contempla los siguientes alcances de acuerdo a los módulos mencionados:

Módulo de registros a los conductores y vehículos

- Registrar a todos los propietarios que prestan el servicio de transporte escolar
- Registrar a los choferes, puede que un propietario tenga varios choferes
- Registrar a los vehículos.
- Verificar los requerimientos.

Módulo tarjeta de identificación del conductor escolar (T.I.C.E.)

- Generar una tarjeta de Identificación con los datos del conductor y vehículo
- Generar un código de tarjeta, un código QR
- Registrar los antecedentes del conductor

Módulo de reportes, consultas, búsqueda y datos estadísticos

- Reportes estadísticos sobre los sindicatos, conductores y vehículos
- Listar a los conductores y vehículos, establecimiento educativo, sindicato
- Buscar al conductor por su carnet, código de conductor o código de vehículo.

Módulo de administración de usuarios

- Registro de los usuarios que intervienen con el sistema
- Asignación de roles.

1.6.2 LÍMITES

- El sistema no está relacionado con el área contable de la unidad
- El sistema solo cubre para los usuarios de la unidad, policías de otras unidades no tienen acceso.
- Es un sistema web y no contempla una aplicación Android.

1.7 APORTES

1.7.1 PRÁCTICO

El presente proyecto tiene como aportes realizar un sistema web de control al sector del transporte escolar con las siguientes características: Adaptable a todo tipo de dispositivo, diseño fácil de comprender para el usuario, privacidad del usuario y seguridad de la información, además de otros aspectos como búsqueda de la información inmediata, centralizar y sistematizar la información, validación de los datos en los registros, acceso inmediato al sistema, registro y verificación de antecedentes.

1.7.2 TEÓRICO

Para el desarrollo del proyecto se utilizará la metodología ágil SCRUM que es un marco de trabajo que logra acabar proyectos en corto tiempo, es iterativo incremental, estructura el desarrollo en ciclos de trabajo llamados *sprint*, para el modelado del sistema web se utilizará UWE que cubre todo el ciclo de vida del software basado en UML que un lenguaje de modelado para la planificación del sistema, además se utilizará la arquitectura de software MVC (Modelo Vista Controlador) que separa la lógica de negocio donde se maneja la base de datos de la interfaz de usuario.

1.8 METODOLOGÍA

1.8.1 METODOLOGÍA CIENTÍFICA

La investigación científica es un proceso metódico que sigue una metodología y sistemático sigue una serie de pasos dirigido a la solución de problemas o preguntas científicas, mediante la producción de nuevos conocimientos, los cuales constituyen la solución o respuesta a tales interrogantes (Arias, 2012).

Vamos a utilizar la investigación de tipo científica aplicada, que resuelve problemas prácticos, además de la exploratoria que estudia el objeto a nivel superficial y descriptiva para establecer su estructura.

1.8.2 METODOLOGÍA DE DESARROLLO

El presente trabajo se regirá bajo la metodología ágil SCRUM, por su simplicidad, rapidez, flexibilidad, procesos de producción en corto tiempo lo que la unidad requiere.

Para el modelado del sistema UWE ingeniería web basado en UML, nos permite visualizar (plano) las funciones, los procesos, el lenguaje de programación, la estructura de base de datos y demás aspectos del sistema

Para la arquitectura del software MVC (Modelo Vista Controlador), reutiliza código, nos proporciona una mejor arquitectura de desarrollo, separa la lógica del negocio con la representación gráfica.

Las herramientas de desarrollo que se utilizarán: Apache para el servidor, MySQL como gestor de base de datos, NodeJS y Express en el lado del servidor, Angular 13 como *framework* de JS para trabajar en el *frontend*, Bootstrap y Angular Material en el manejo de estilos y Typescript como lenguaje de desarrollo web.

2. CAPÍTULO II MARCO TEÓRICO

2.1 INTRODUCCIÓN

Para el presente capítulo se hace referencia a los conceptos y fundamentos teóricos que sustentan el desarrollo del presente proyecto, la unidad de Servicios Públicos tiene la finalidad del control y supervisión policial, mediante los servicios de la Dirección Departamental de Tránsito Transporte y Seguridad Vial, procediendo al registro e identificación plena del conductor y vehículo, al mismo tiempo inculcar a los señores

conductores a obedecer y respetar las normas y señales de tránsito. Según entrevistas con los Solicitantes.

La ingeniería de software comprende todos los aspectos de la producción, desarrollo y mantenimiento del producto, el proceso es un conjunto de actividades las cuales son: comunicación, planeación, modelado, construcción, acciones y tareas que se deben realizar y para guiar las actividades se está implementando la metodología ágil SCRUM que nos permite llevar a cabo proyectos de desarrollo, adaptables a diferentes cambios, es un marco de trabajo iterativo incremental, para el modelado que es uno de los puntos importantes en las actividades se implementa el modelo de desarrollo UWE basado en UML, que se utiliza para aplicaciones web.

La arquitectura de software nos ayuda a organizar los componentes del sistema, y que mejor implementar el patrón MVC que está encargado de separar la lógica de negocio de la interfaz de usuario, a estas dos áreas en el mundo de desarrollo web la conocemos como *backend* y *frontend*.

2.2 MARCO INSTITUCIONAL

2.2.1 ASPECTOS GENERALES

El transporte escolar es un servicio público para los alumnos de diferentes unidades educativas, sean públicos o privados, se trata de buses pequeños o medianos, debiendo estar equipados con los medios necesarios. A través de medios de comunicación, denuncias, se han tomado conocimientos de diferentes casos de abuso sexual, conducción peligrosa, consumo de bebidas alcohólicas en los motorizados, fallas mecánicas generando inseguridad, por lo tanto, la unidad tiene la misión y responsabilidad de ejercer un control y supervisión sobre este tipo de servicios públicos, que es el transporte escolar (Quispe, Rocha, Monasterio y Pereira, 2018)

Bajo la normativa legal de la Constitución Política del Estado en su artículo 251, la policía boliviana tiene la misión específica de la defensa de la sociedad y la conservación del servicio público y el cumplimiento de las leyes en todo el territorio nacional.

Ley Nro. 3988 código y reglamento de tránsito, artículo 17 ningún vehículo que no este en buenas condiciones podrá circular en las vías públicas. Ley Nro. 548 código niño niña y adolescente en su artículo 2 garantiza el ejercicio pleno y efectivo de sus derechos, para su desarrollo integral, entre otros artículos es necesario resaltar para respaldar el presente proyecto.

Según resolución ministerial Nro. 011/2012 establece que las personas que tenga interés en incorporarse al servicio público deberán realizar la inscripción oficial y obligatoria del registro del transporte escolar a través de la tarjeta de identificación del conductor escolar (T.I.C.E.) con fines de identificación, responsabilidad y estadística (Quispe, Rocha, Monasterio y Pereira, 2018)

2.3 INGENIERÍA DE SOFTWARE

Según, (Sommerville, 2005) nos da la siguiente definición: “La ingeniería de software es una disciplina de ingeniería que comprende todos los aspectos de la producción de software. ¿Cuál es la diferencia entre ingeniería de software y ciencia de la computación? la ciencia de la computación comprende la teoría y los fundamentos; la ingeniería de software comprende las formas prácticas para desarrollar y entregar un software útil.” (p. 5).

Ingeniería de software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software, los objetivos de la ingeniería de software son mejorar el diseño, mayor calidad, eficiencia, exactitud, y una mejor organización de trabajo en equipo (Pressman, 2010)

2.3.1 EL PROCESO DEL SOFTWARE

El proceso de software según (Pressman, 2010) es un conjunto de actividades, las cuales son: comunicación, planeación, modelado, construcción y despliegue, acciones y tareas que se deben realizar. En la actualidad se utiliza el desarrollo ágil. La Agilidad describe un proceso de software moderno, todos son ágiles, un equipo ágil es diestro y capaz de responder de manera apropiada a los cambios. La metodología es un marco de trabajo, utilizado para planificar, estructurar y controlar el proceso de desarrollo del sistema.

2.3.2 MODELADO

Todas las actividades mencionadas se deben modelar, empezando por la comprensión de los requerimientos, escenarios y análisis. Diseño del software Abstracción, Arquitectura, Patrones. Diseños nivel de componentes. Diseño de la interfaz de usuario, Calidad del Software, Estrategias de prueba de software. (Pressman, 2010)

2.4 METODOLOGÍA DE DESARROLLO ÁGIL SCRUM

El desarrollo ágil es una metodología de gestión de proyectos adaptativa, que permite llevar a cabo proyectos de desarrollo, adaptables a diferentes cambios evolucionando. “SCRUM es un marco de trabajo iterativo e incremental para el desarrollo de proyectos, productos y aplicaciones. Estructura el desarrollo en ciclos de trabajo llamados Sprint son de duración fija, terminan en una fecha específica, aunque no se haya terminado el trabajo, y nunca se alargan. Se limitan en tiempo. Al comienzo de cada Sprint, un equipo multifuncional selecciona los elementos al final del sprint. Durante el sprint no se pueden cambiar los elementos elegidos” (Deener, Benefield, Lerman, & Bodde, 2009, p. 5)

La mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor, los responsables del negocio y los desarrolladores trabajan juntos de forma cotidiana durante todo el proyecto. Ver Figura 2.1

2.4.1 COMPONENTES DEL SCRUM

Es una serie de reglas que define roles y artefactos necesarios para los procesos en intervalos de tiempo, en estos intervalos se cumplen los siguientes objetivos: Reunión de planificación de entrega, planificación del Sprint, reunión diaria, revisión y retrospectiva. (Palacios, 2015)

2.4.1.1 LAS REUNIONES

Planificación del Sprint, se definirá un documento en el que se reflejaran los requisitos del sistema por prioridades, la planificación del sprint 0 en las que se deducirá cuales vas a ser los objetivos y el trabajo que hay que realizar.

Seguimiento del Sprint, en esta fase se realiza reuniones diarias, cada persona actualiza en

la pila del sprint el tiempo pendiente de sus tareas y también se actualiza el gráfico

Revisión del Sprint, análisis y revisión del incremento generado, se presenta los resultados finales. (Palacios y Ruata, 2009). Ver figura 2.2

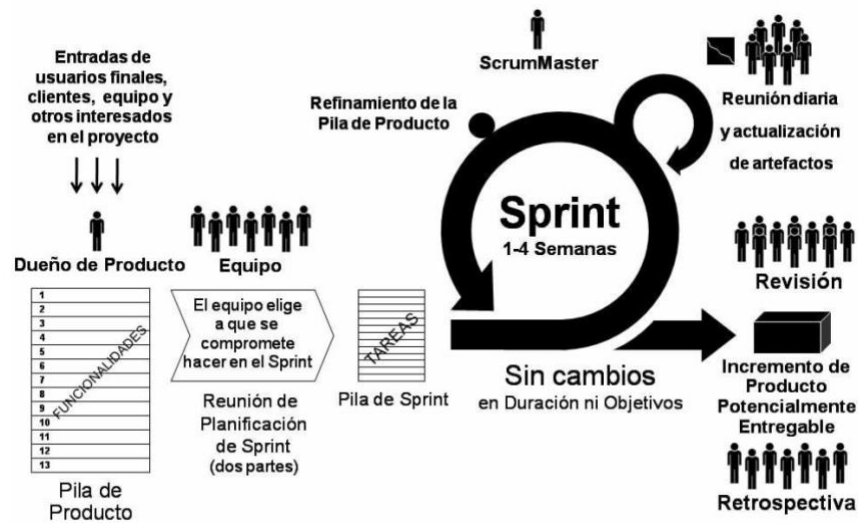


Figura 2.1 Roles, artefactos y eventos principales del SCRUM

Fuente: (Deener, Benefield, Lerman, & Bodde, 2009)

2.4.1.2 LOS ROLES

Los principales roles de SCRUM según Kniberg y Skarin (2010) son:

SCRUM Master: Es el responsable del cumplimiento de las reglas del marco.

Product Owner: Se asegura de que el equipo SCRUM trabaje de forma adecuada desde la perspectiva del negocio.

Equipo Team: realiza la entrega del producto generado en cada sprint.

Tenemos como roles secundarios:

Usuarios: Es el destinatario final del producto.

Stakeholders: Son las personas a las que el proyecto les producirá un beneficio, participan durante las revisiones del sprint.

Managers: Es quien toma las decisiones finales participando en la selección de los objetivos y requisitos (Palacios y Ruata, 2009).

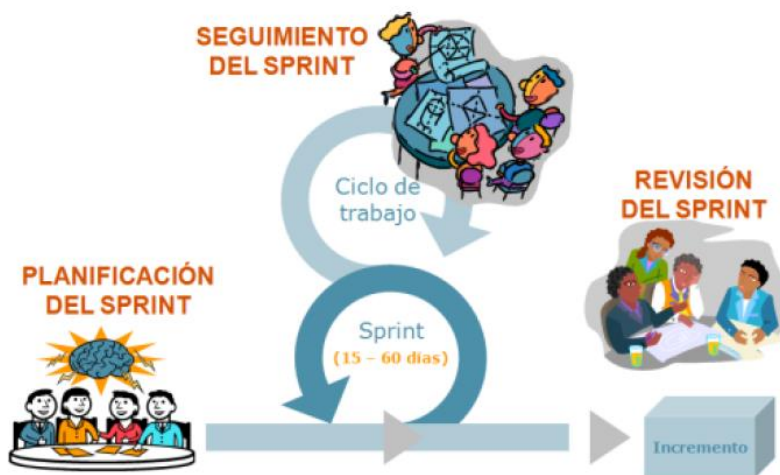


Figura 2.2 Las reuniones habituales del SCRUM

Fuente: (Palacios y Ruata, 2009).

2.4.2 ELEMENTOS DEL SCRUM

2.4.2.1 PRODUCT BACKLOG

Es la lista de necesidades del cliente, donde se almacena todas las funcionalidades o requisitos en forma de lista priorizada. Estos requisitos serán los que tendrá el producto, la lista será gestionada y creada por el cliente con la ayuda del SCRUM Master, quien indicará el coste estimado para completar un requisito. Como se observa en la tabla 2.1

Las principales características son:

- Contendrá los objetivos del producto, se suele usar para expresarlas las historias de usuario
- En cada objetivo. Se indicará el valor que se le dará el cliente y el coste estimado; de esta manera se realizará la lista, priorizando por valor y coste
- En la lista se tendrá que indicar las posibles iteraciones y los *releases* que se han indicado al cliente

Id	Prioridad	Descripción	Est.
1	Muy alta	Plataforma tecnológica	30
2	Muy Alta	Interfaz de usuario	40
3	Muy Alta	Un usuario se registra en el sistema	40
4	Alta	El operador define el flujo y textos de un expediente	60
5	Alta	xxx	999

Tabla 2.1 Ejemplo de Pila de Producto
Fuente: (Palacios, 2015)

Historias de Usuario son las descripciones de las funcionalidades que va a tener el software. Sera el resultado de la colaboración entre el cliente y el equipo, e irán evolucionando en toda la vida del proyecto.

Se componen de tres fases denominadas las 3 C:

- *Card*: será una breve descripción escrita que servirá como recordatorio
- *Conversation*: es una conversación que servirá para asegurarse de que se ha entendido bien todo
- *Confirmation*: test funcionales para fijar detalles que sean relevantes e indicar cuál va ser el límite

Id: 1 Historia de Usuario	
Descripción: Como jefe de la Unidad de servicios públicos quiero registrar	
Usuario: jefe de la Unidad de División	
Prioridad: Alta	Estimación: 2

Tabla 2.2 Ejemplo de historia de usuario
Fuente: Elaboración propia

Descripción sintetiza la conversación con el cliente tomando en cuenta como, quiero y para

Usuario es el cliente con el que se realiza la entrevista y se recaba los requerimientos

Prioridad en la implementación, para tal efecto tendremos 3 tipos: alta, media y baja

Estimación es la evaluación del coste de implementación en unidades de desarrollo estas unidades representaran el tiempo teórico (desarrollo/hombre) que se haya estimado al

comienzo del proyecto. (Palacios, 2015).

2.4.2.2 SPRINT BACKLOG

Es la lista que descompone las funcionalidades de la pila de producto en las tareas necesarias para construir un incremento: una parte completa y operativa del producto.

El sprint backlog es una lista ordenada por prioridades para el cliente, puede haber dependencias entre una tarea y otra, todas las tareas tienen que tener un coste de entre 4 – 16 horas.

Generalmente esta lista se plasma en una pizarra que contempla:

- La lista de tareas.
- La persona responsable de cada tarea.
- El estado en la que se encuentra y el tiempo que queda por terminar.

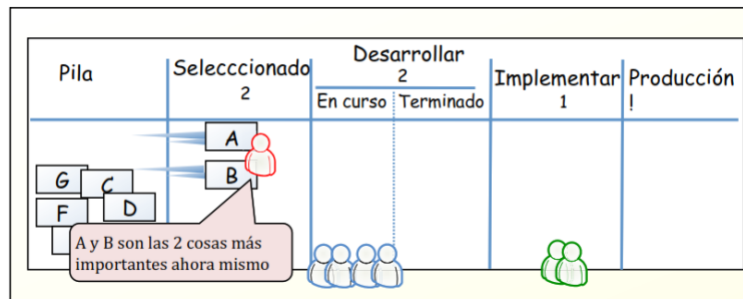


Figura 2.3 Tablero de SCRUM
Fuente: (Kniberg y Skarin, 2010)

En la figura 2.3 se puede observar el proceso visual del desarrollo, creando un flujo suave a través del sistema y minimizar el tiempo de entrega. Cada columna representa un estado de trabajo, la primera columna representa el *product backlog*, la segunda es el *sprint backlog*, la tercera columna representa el desarrollo de las tareas por el equipo, la cuarta columna se implementa y realiza las pruebas correspondientes y por último esta la quinta columna donde la tarea pasa a producción. El incremento es una tarea terminada en condiciones de ser entregada al cliente final (Trigas, s.f)

2.4.3 DESARROLLO DE LAS FASES DE UN PROYECTO EN SCRUM

El flujo de trabajo inicia con las reuniones del cliente y el *product owner*, registra y prioriza los requisitos desde el punto de vista del cliente. Empieza con una visión inicial del producto y crece y evoluciona durante el desarrollo del producto. Los requisitos suelen denominarse historias de usuario. El equipo de desarrollo realiza el sprint backlog que es la lista de tareas que se deben realizar durante un sprint para lograr el incremento previsto. Sprint es el periodo o intervalo de tiempo definido por el equipo. Al final de un sprint, el equipo realiza dos eventos: la revisión del *sprint* y la retrospectiva del *sprint*. Después de cada *sprint*, se lleva a cabo una retrospectiva del sprint, en la cual todos los miembros del equipo dejan sus impresiones sobre el sprint recién superado. El propósito de la retrospectiva es realizar una mejora continua del proceso. (Bahit, 2012).

2.4.3.1 PRE-GAME

Es la preparación del proyecto en la que se comprende el caso de negocio con la finalidad de tomar decisiones, las tareas a realizar son:

- Definir el Proyecto: se debe indicar de forma clara el propósito del proyecto, identificar las necesidades del producto y del cliente
- Definir el *Backlog*: Se comienza la creación del *Backlog* del producto para que el *sprint* siguiente contenga elementos de la lista suficiente para comenzar a trabajar. Esta lista de elementos lo realiza el *Product Owner* que tendrá como responsabilidad priorizar las funcionalidades.
- Definir los entregables: una vez que se tiene el *Backlog* con las funcionalidades, es necesario establecer criterios para hacer pequeñas entregas del producto y así obtener su valor

2.4.3.2 GAME

En esta etapa se realizarán las siguientes tareas:

- Planificar un *sprint*: se realiza una reunión entre el *Product Owner*, el SCRUM master y

el equipo de desarrollo, con el objetivo de seleccionar la lista *backlog* del producto las funcionalidades sobre las que se va a trabajar.

- Desarrollo del *sprint*: el equipo de desarrollo trabaja para conseguir un incremento del producto, entre 2 a 4 semanas. Durante este periodo se realiza 3 reuniones:
 - Reunión de Planificación: el equipo realiza consultas, se auto gestionara y puede eliminar un sprint según su dificultad o si es viable
 - Reunión Diaria: comparten información relativa al desarrollo se responde a las siguientes preguntas ¿Qué has hecho de nuevo con respecto a la última reunión diaria?, ¿Qué será lo siguiente a realizar? y ¿Qué problemas hay para realizarlo?
 - Reunión de Revisión del sprint: el equipo de desarrollo presenta el producto entregable y presenta las funcionalidades

2.4.3.3 POST GAME

Es el cierre, en esta fase se realizan las actividades de fin de proyecto, documentación necesaria, testear o pruebas de estrés rendimiento y marketing para nuevo software. (Palacios y Ruata, 2009)

2.4.4 PRUEBAS DE SOFTWARE

Una estrategia de prueba de *software* debe ser suficientemente flexible para promover un uso personalizado de la prueba, puede verse como la espiral que se ilustra en la figura 2.4 define el papel de software y conduce el análisis de los requerimientos, donde se establecen los criterios del dominio, función, comportamiento, desempeño, restricciones y validación de información. Al avanzar hacia adentro a lo largo de la espiral se llega al diseño y finalmente a la codificación.

La **prueba de unidad** comienza en el vértice de la espiral y se concentra en cada unidad (por ejemplo, componente, clase, objeto) del software como se implementó en el código fuente. La prueba avanza al moverse hacia afuera a lo largo de la espiral, hacia la **prueba de integración**, donde el enfoque se centra en el diseño y la construcción de la arquitectura del software. Al dar otra vuelta hacia afuera de la espiral, se encuentra la **prueba de validación**,

donde los requerimientos establecidos como parte de su modelado se valida confrontándose con el software que se construyó. Finalmente se llega a la **prueba del sistema**, donde el software y otros elementos del sistema se prueban como un todo.

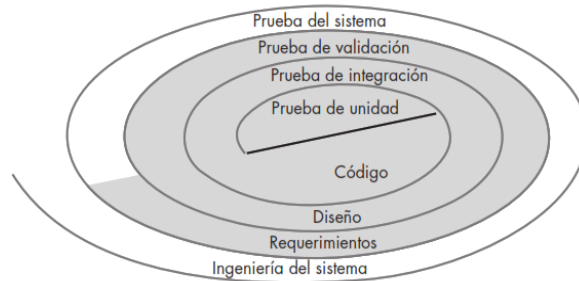


Figura 2.4 Prueba de software
Fuente: (Pressman, 2010)

2.4.4.1 PRUEBAS DE CAJA NEGRA

La prueba de caja negra se basa en la comprobación de la funcionalidad esperada en base a su interfaz (parámetros de E/S) del componente de *software*, disponemos de un módulo, comprobamos su funcionalidad en base a los parámetros de entrada y resultados obtenidos con el objetivo de encontrar errores en las estructuras de datos, su inicialización y finalización como se observa en la figura 2.5.

Se implementará la técnica de partición equivalente que consiste en derivar casos de prueba mediante la división del dominio de entrada en clases de equivalencia, evaluando su comportamiento para un valor cualquiera, una clase de equivalencia representa al conjunto de estados validos o inválidos para todas las condiciones de entrada que satisfagan dicha clase. (Pressman, 2010)

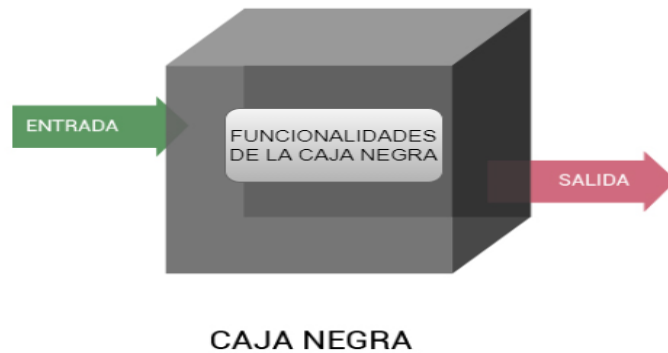


Figura 2.5 Caja Negra
Fuente: Elaboración propia

2.5 MODELO DE DESARROLLO UWE BASADO EN UML

UWE es un método de ingeniería web orientado a objetos basado en UML que se utiliza para aplicaciones web. El Lenguaje Unificado de Modelado (*Unified Modeling Language, UML*) es un lenguaje estándar para escribir planos de software. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucre una gran cantidad de software. (Rumbaugh, Jacobson, y Booch, 2007)

En esta sección se aplica principios y patrones para crear diseños, seguir un conjunto de actividades comunes en el análisis basado en el proceso unificado, crea los diagramas más utilizados con frecuencia en la notación UML. (Larman, 2003)

La extensión UWE consiste en agregar los siguientes modelos o paquetes que refleja la separación de preocupaciones de las aplicaciones web.

2.5.1 MODELO DE REQUISITOS

Está comprendida por los modelos de caso de uso para discernir la navegación del proceso, modela la funcionalidad del sistema según lo perciben los usuarios externos, llamados actores. Un caso de uso es un caso coherente de funcionalidad, expresada como transacción entre los actores y el sistema, con el propósito de enumerar los actores y casos de uso. Ver figura 2.6.

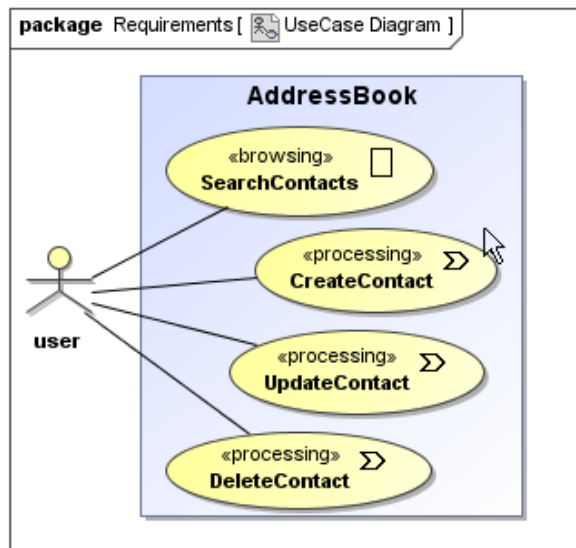


Figura 2.6 Diagrama de Casos de Uso
 Fuente: (UWE – Ingeniería web basado en UML, 2016)

2.5.2 MODELO DE CONTENIDO

Representa los diagramas de clases de UML, que muestra como un conjunto de interfaces, colaboraciones y sus relaciones, comparte las propiedades comunes al resto de los diagramas. Los elementos que contiene son: Clases, Interfaces y Relaciones de dependencia, generalización y asociación.

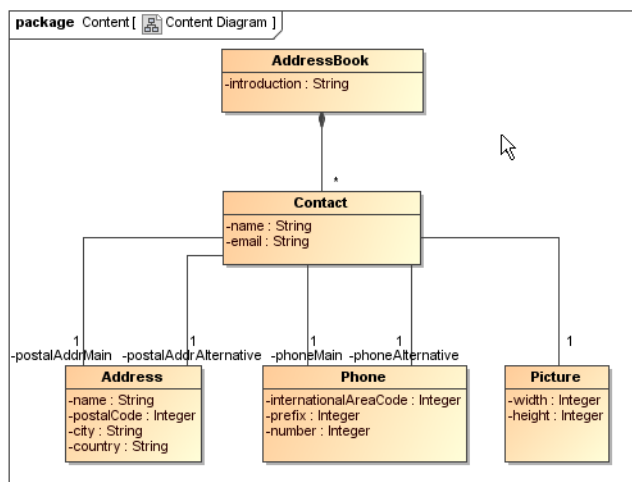


Figura 2.7 Diagrama de Clases
 Fuente: (UWE – Ingeniería web basado en UML, 2016)

2.5.3 MODELO DE NAVEGACIÓN

Para un sistema de páginas web, es necesario conocer como están vinculados entre sí, para esto necesitamos un diagrama que contenga nodos y enlaces.

Los nodos son unidades de navegación conectadas por enlaces, los nodos se pueden mostrar en diferentes paginas o en la misma página.

Para obtener un diagrama de navegación es hacer una transformación de contenido a navegación, para los nodos y enlaces se utilizará estereotipos, <<navigationClass>> y <<navigationLink>>. (Ver Figura 2.8).

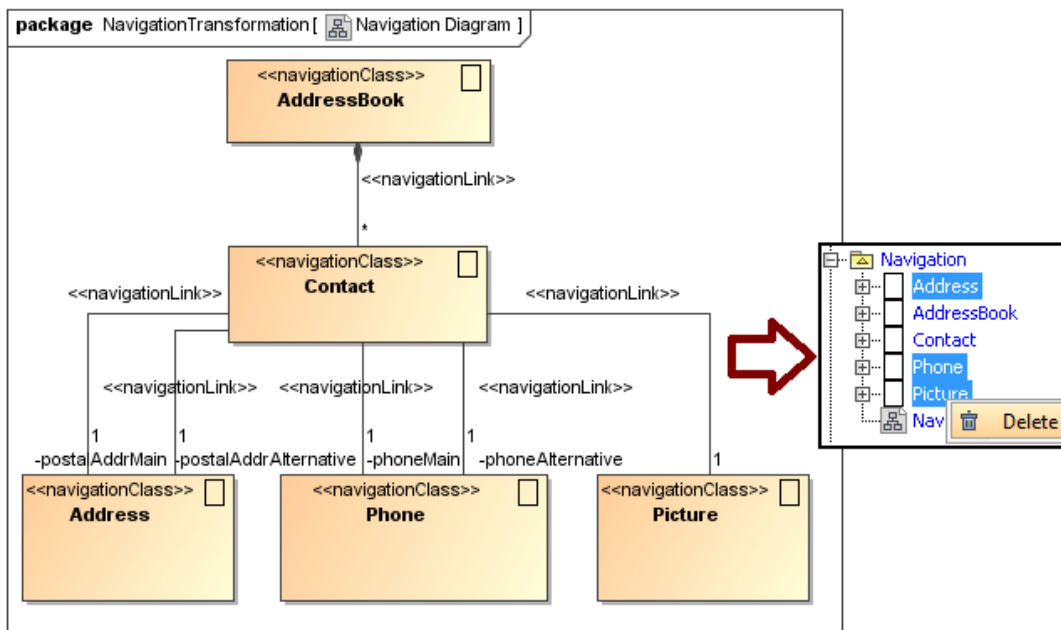


Figura 2.8 Diagrama de Navegación

Fuente: (UWE – Ingeniería web basado en UML, 2016)

Según la figura 2.8 *AddressBook* sería nuestra página principal o *index* y los demás nodos serían las operaciones en las que se puede acceder desde la página principal.

En UWE se puede utilizar un menú para navegar a diferentes clases como se muestra en la figura 2.9.

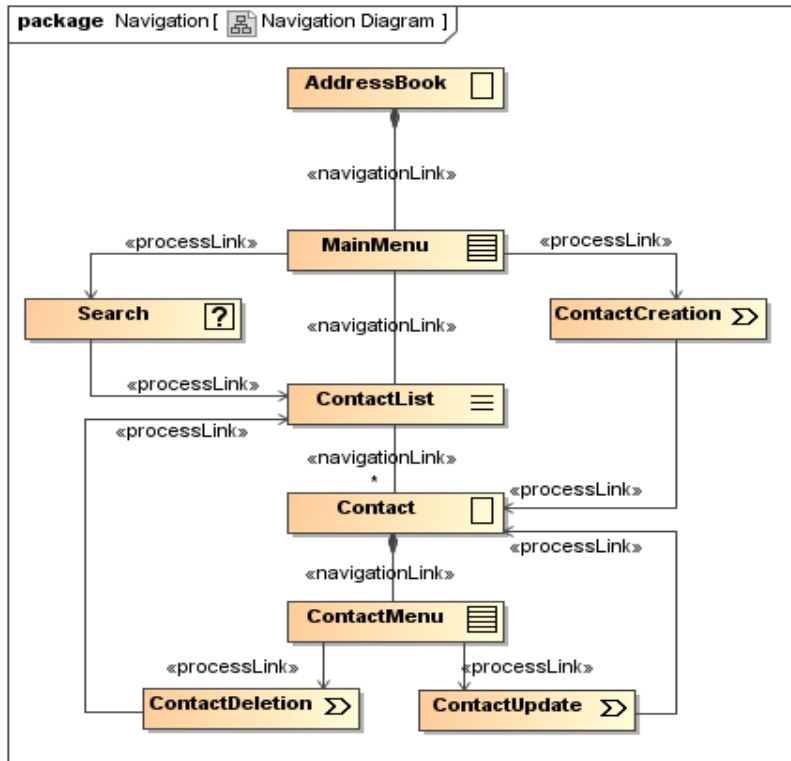


Figura 2.9 Diagrama de Navegación con menú
 Fuente: (UWE – Ingeniería web basado en UML, 2016)

2.5.4 MODELO DE PRESENTACIÓN

El modelo de presentación proporciona una vista abstracta de la interfaz de usuario de una aplicación web, es decir, que elementos de la interfaz de usuario (texto, imágenes, anclajes, formularios, tablas, y otros) se utilizan para presentar los nodos de navegación.

nombres-estereotipos y sus iconos

presentaciónGrupo	presentaciónPágina
texto	entrada de texto
ancla	Subir archivos
botón	imagen
inputForm	customComponent
presentaciónAlternativas	selección

Figura 2.5 Estereotipos de Presentación
 Fuente: (UWE – Ingeniería web basado en UML, 2016)

Un ejemplo de diagrama de presentación se puede apreciar en la siguiente figura:

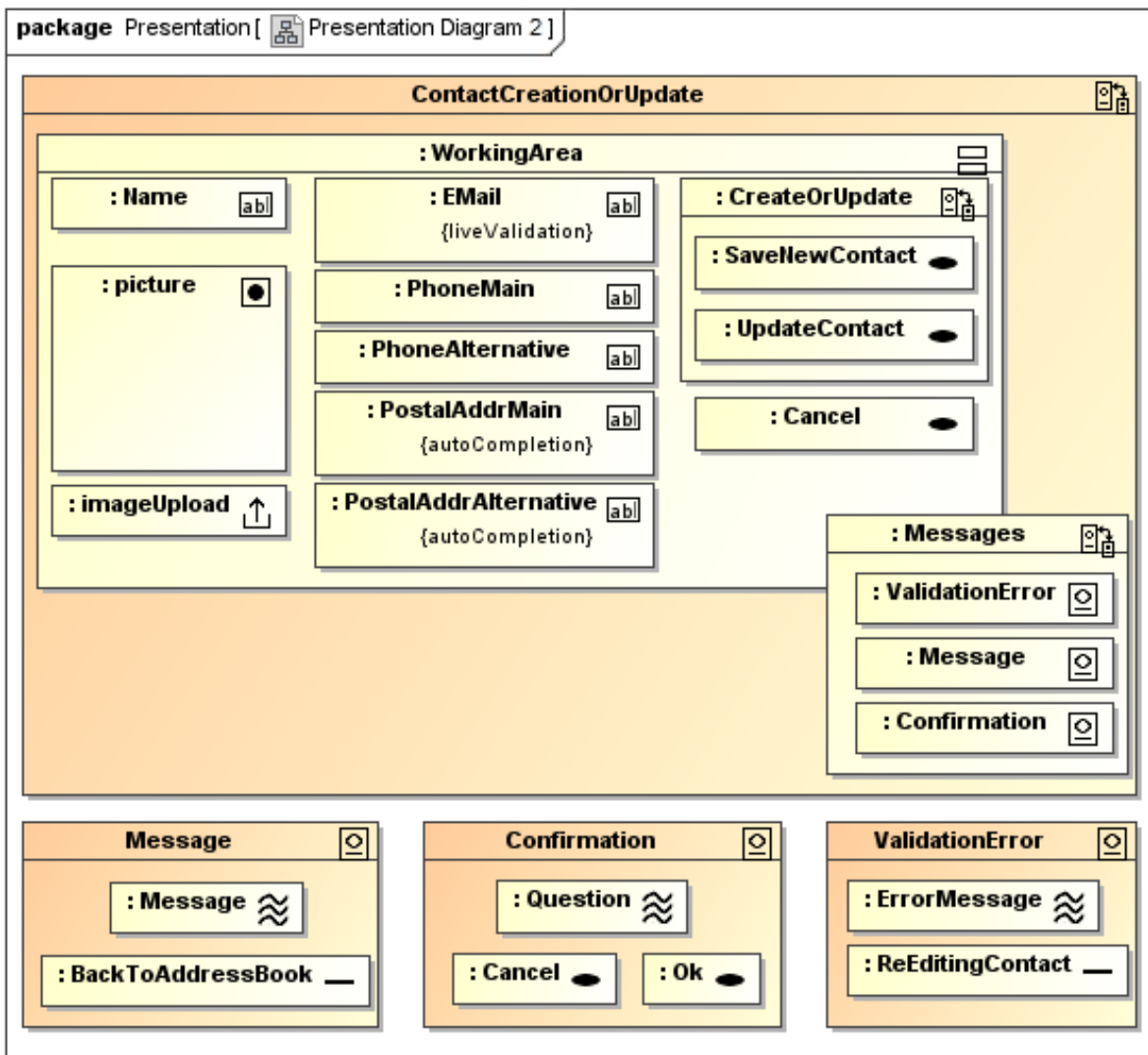


Figura 2.6 Diagrama de Presentación
 Fuente: (UWE – Ingeniería web basado en UML, 2016)

2.5.5 MODELO DE PROCESO

Describe el comportamiento de una clase, si el usuario navega por esa clase que sucederá en detalle. Es un diagrama de actividades de UML. (Ver figura 2.12).

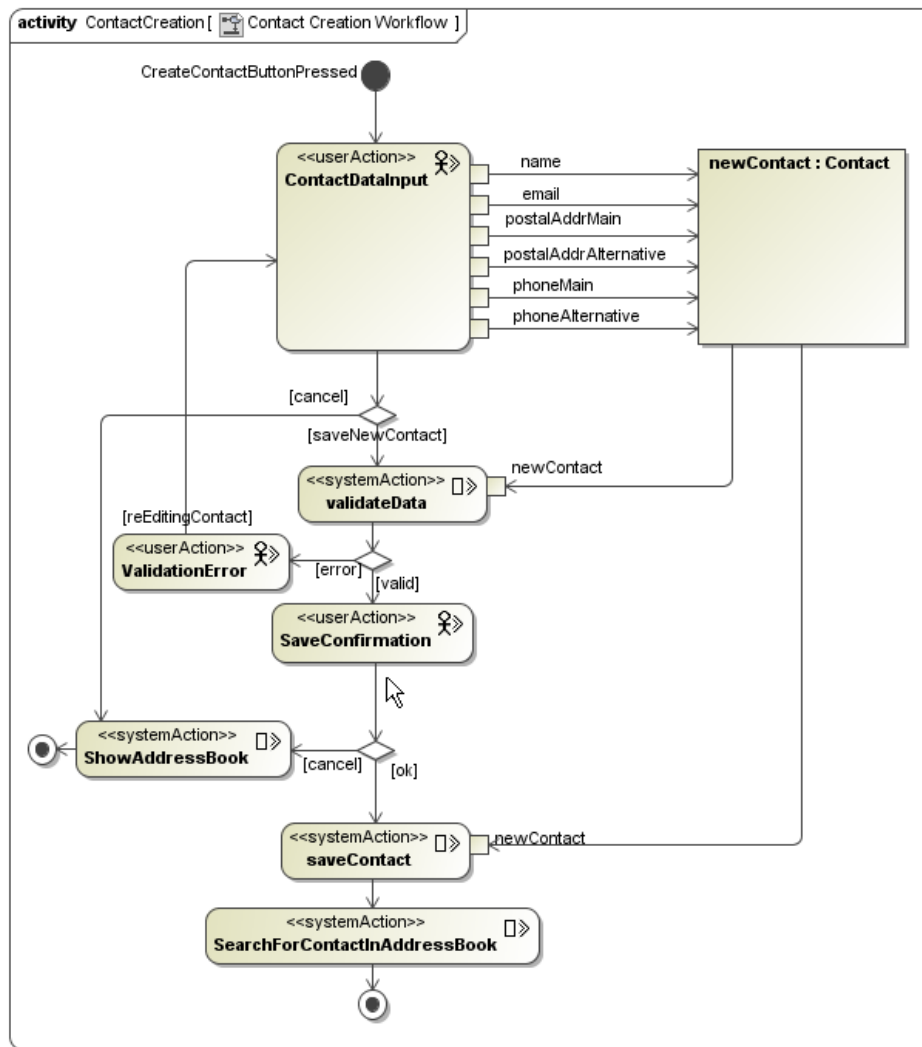


Figura 2.7 Diagrama de Proceso

Fuente: (UWE – Ingeniería web basado en UML, 2016)

2.6 PATRON ARQUITECTONICO MVC

Según Bahit (s.f.), la arquitectura de software es la forma en la que se organizan los componentes de un sistema, interactúan y se relacionan entre sí y con el contexto. El patrón MVC está encargado de separar la lógica negocio de la interfaz de usuario, está dividida en tres niveles de abstracción:

- **Modelo:** representa la lógica de negocio, es el encargado de acceder de forma directa a

los datos, actuando como intermediario con la base de datos.

- **Vista:** es la encargada de mostrar la información al usuario de forma gráfica y amigable.
- **Controlador:** es el intermediario entre la vista y el modelo, es quien controla las interacciones del usuario solicitando los datos del modelo y entregándolos a la vista.

En la figura 2.13 se puede observar el funcionamiento básico, el usuario realiza una petición, el controlador captura el evento hace una llamada al modelo, el modelo será el encargado de interactuar con la base de datos y retorna la información al controlador, por último, el controlador recibe la información y le envía a la vista.

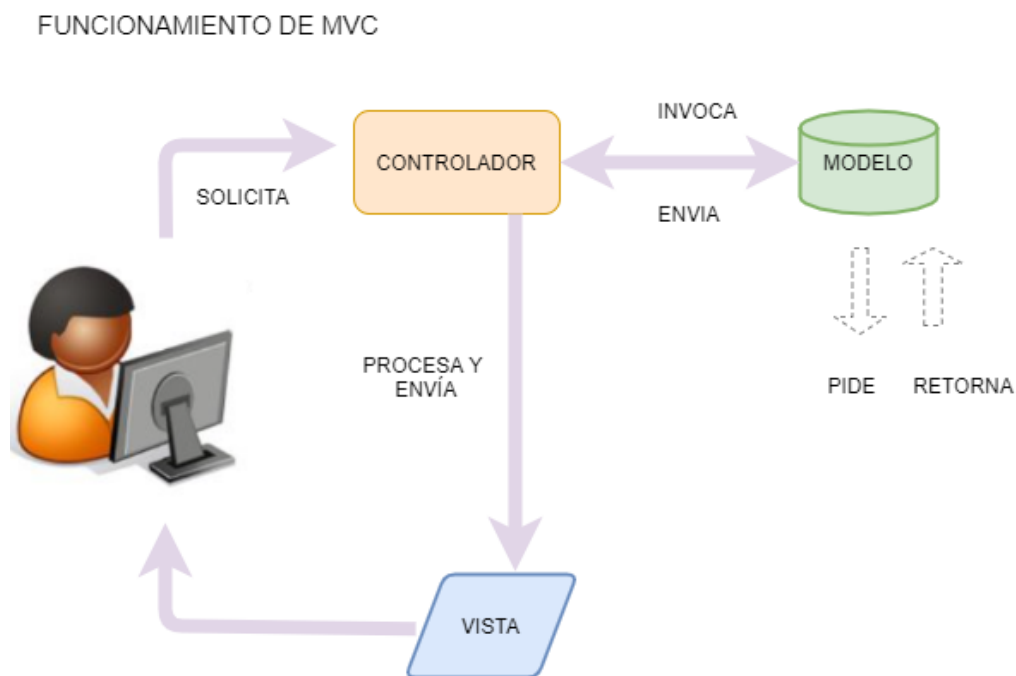


Figura 2.8 Fundamentos del patrón Modelo Vista Controlador

Fuente: Elaboración propia

2.7 HERRAMIENTAS TECNOLÓGICAS

Las herramientas tecnológicas son programas y aplicaciones fácil de utilizar y sin la necesidad de tener que pagar por ello, nos ofrece intercambiar información y conocimiento, para este caso vamos a separar las herramientas en dos campos.

2.7.1 BACKEND

El *backend* procesa la información que alimenta al *frontend* de datos, consiste en un servidor, una aplicación y una base de datos, se va utilizar las siguientes herramientas:

- Node JS: es un entorno en tiempo de ejecución multiplataforma de código abierto para la capa del servidor basado en Javascript.
- Express: es un marco de aplicación web de back-end para Node.js
- MySQL: es el gestor de base de datos más utilizado para el desarrollo de aplicaciones web, es totalmente administrable.

2.7.2 FRONTEND

Es el lado del cliente, se dedica a la parte frontal de un sitio web, es todo el código que se ejecuta en el navegador, para el presente proyecto se va utilizar las siguientes herramientas:

- Bootstrap: es el marco de trabajo más popular del mundo para crear sitios con capacidad de respuesta, es un kit de herramientas para desarrollar con HTML, CSS y JavaScript. (Bootstrap, 2021)
- Angular: es un *framework* para aplicaciones web desarrollado en TypeScript, de código abierto.
- Angular Material: es un módulo construido por y para Angular. Dicho módulo permite implementar componentes Angular con un diseño basado en Material Design.

3. CAPITULO III MARCO APLICATIVO

3.1 INTRODUCCIÓN

La ingeniería de software nos propone para el desarrollo de un sistema realizar actividades como ser: comunicación, planeación, modelado, construcción, acciones y tareas. La metodología ágil SCRUM cubre estas actividades en un marco de trabajo iterativo e incremental y para el modelado se implementa el modelo de desarrollo uwe basado en uml, a continuación, presentamos las fases del SCRUM incorporando el modelado uwe y se puede visualizar de manera gráfica en la figura 3.1.



Figura 3.1. Trabajando UWE con SCRUM

Fuente: Elaboración propia

Pre Game: Es la preparación del proyecto, comprender la lógica de negocio, definir el proyecto es identificar las necesidades del producto y del cliente asignando roles y obteniendo las historias de usuario, definir el *product backlog* es la lista de requerimientos y definir los entregables es establecer criterios para hacer pequeñas entregas. Para el modelado se implementa: los diagramas de casos de uso, diagrama de clases, diagrama entidad relación y diagrama relacional

Game: En esta etapa planificamos el *sprint* seleccionamos la lista *backlog* del producto,

además de desarrollar el sprint durante un periodo realizando 3 reuniones para obtener un incremento. Para el modelado y diseño web se implementó: modelo de requisitos, modelo de contenido, modelo de navegación, modelo de presentación, modelo de proceso y modelo de datos para cada sprint. Finalmente se realizó las pruebas correspondientes.

Post Game: Es el cierre, fin del proyecto, documentación necesaria y presentación de las interfaces de pantalla.

3.2 PRE-GAME

3.2.1 PLANEAMIENTO Y ANÁLISIS

La obtención de requisitos es una de las etapas fundamental, mediante las entrevistas se realizará las historias de usuario y se arma la pila de producto backlog. El diagrama de caso de uso identifica a los actores y la interacción que tiene con el sistema, el diagrama de clases nos ayudara en las relaciones y colaboraciones entre los componentes, y el diagrama de datos refleja el comportamiento del sistema.



Figura 3.2. Desarrollo de SCRUM y UWE
Fuente: (Palacios y Ruata, 2009).

3.2.2 ASIGNACIÓN DE ROLES

En base al análisis y los antecedentes de la institución, se ha identificado a tres tipos de usuarios que interactúan con el sistema, en la tabla 3.1 se puede observar las características y tareas específicas que tendrá cada uno de ellos. En la tabla 3.2 se observa las responsabilidades y roles del SCRUM.

USUARIO	DESCRIPCIÓN	TAREAS
Administrador y/o empadronamiento	Es el encargado de la unidad de Servicios públicos	Administrar a los usuarios Asignación de roles Realizar el registro de los conductores y vehículos Digitalizar los requisitos de registro Visualizar, modificar y eliminar los registros Generar la tarjeta de identificación del conductor escolar (T.I.C.E.) Gestión de los antecedentes, ante las denuncias Generar Reportes y datos estadísticos
Policía boliviana	Brinda servicios de defensa a la sociedad y conservación del orden público	Buscar al conductor por el número de carnet o código de tarjeta Registro de alguna falta o denuncia por parte de los conductores

Tabla 3.1 Tipos de usuario
Fuente: Elaboración propia

Los roles para la metodología SCRUM fueron designados como sigue a continuación:

Roles	Nombre
SCRUM Master	Rosario Verónica Quispe Alanoca
Equipo de trabajo (SCRUM Team)	Rosario Verónica Quispe Alanoca
Product Owner	Sgto Mayor Angel Quispe

Tabla 3.2 Roles del SCRUM
Fuente: Elaboración propia

3.2.3 HISTORIAS DE USUARIO

En base a las entrevistas realizadas a los usuarios identificados que intervienen con el sistema se elaboró una descripción escrita sintetizando la conversación para obtener los requisitos, tomando en cuenta la prioridad y la estimación. A continuación, se observa las historias de usuario.

En la tabla 3.3 se puede observar la historia de usuario correspondiente a la gestión del mismo.

Id: 1		Historia de Usuario	
Descripción: Como administrador quiero registrar a todos los usuarios que intervienen con el sistema para que puedan acceder al mismo			
Usuario: Administrador			
Prioridad: Alta		Estimación: 3	

Tabla 3.3 Gestión de usuario
Fuente: Elaboración propia

En la siguiente tabla se visualiza la historia de usuario para el registro de conductores, vehículos y establecimientos educativos.

Id: 2 Historia de Usuario	
Descripción: Como administrador quiero registrar a los conductores, sus vehículos y el establecimiento educativo a la que brinda el servicio para obtener información y generar datos estadísticos del mismo	
Usuario: Administrador	
Prioridad: Alta	Estimación: 3

Tabla 3.4 Registro de conductores, vehículos y establecimiento educativo
Fuente: Elaboración propia

La tabla siguiente muestra la historia de usuario para las altas, bajas y cambios de registro de conductores y vehículos.

Id: 3 Historia de Usuario	
Descripción: Como administrador quiero poder visualizar los registros, modificar los datos por si ocurre un error de tipeo y borrar a los conductores y vehículos para una mejor administración de los datos	
Usuario: Administrador	
Prioridad: Media	Estimación: 3

Tabla 3.5 CRUD de los registros de conductores y vehículos
Fuente: Elaboración propia

La tabla 3.6 muestra la historia de usuario para la digitalización de requisitos de afiliación.

Id: 4 Historia de Usuario	
Descripción: Como administrador quiero digitalizar los requisitos de filiación como fotografía, carnet, cartas y demás para evitar el archivo de documentación física.	

Usuario: Administrador	
Prioridad: Baja	Estimación: 3

Tabla 3.6 Digitalizar los requisitos de afiliación

Fuente: Elaboración propia

En la tabla siguiente se muestra la historia de usuario para la tarjeta de identificación del conductor escolar.

Id: 5	Historia de Usuario
Descripción: Como administrador encargado de la unidad quiero generar la tarjeta de identificación del conductor escolar (T.I.C.E.) con los datos del conductor y vehículo dos ejemplares para los conductores y para la unidad.	
Usuario: Administrador	
Prioridad: Alta	Estimación: 4

Tabla 3.7 Tarjeta de identificación del conductor escolar

Fuente: Elaboración propia

La tabla siguiente muestra la historia de usuario para el registro de una nueva gestión, mismo que corresponde a la generación de tarjetas TICE.

Id: 6	Historia de Usuario
Descripción: Como administrador quiero registrar o habilitar una nueva gestión por que la generación de las tarjetas es anual.	
Usuario: Administrador	
Prioridad: Alta	Estimación: 3

Tabla 3.8 Registro de una nueva gestión

Fuente: Elaboración propia

La tabla 3.9 muestra la historia de usuario para el reporte de los conductores, vehículos, unidades educativas y tarjetas TICE.

Id: 7 Historia de Usuario	
Descripción: Como administrador quiero generar reportes de los usuarios, conductores, vehículos y las tarjetas de identificación (T.I.C.E.) por gestión para realizar los informes.	
Usuario: Administrador	
Prioridad: Media	Estimación: 3

Tabla 3.9 Reporte de los conductores y las tarjetas tice
Fuente: Elaboración propia

La siguiente tabla muestra la historia de usuario para la búsqueda de conductores, vehículos, unidades educativas y sindicatos.

Id: 8 Historia de Usuario	
Descripción: Como miembro activo de la policía boliviana quiero buscar al conductor por el nombre, carnet, código de la tarjeta o placa del vehículo mostrando toda la información, para facilitar en los operativos	
Usuario: Policía boliviana, Administrador	
Prioridad: Media	Estimación: 3

Tabla 3.10 Buscar y mostrar a los conductores
Fuente: Elaboración propia

La tabla 3.11 muestra la historia de usuario para la gestión de antecedentes de conductores

Id: 9 Historia de Usuario	
Descripción: Como administrador ante las denuncias realizadas por parte de los padres de familia y/o policía boliviana quiero registrar en los antecedentes del conductor para tomar acción ante las instancias correspondientes	
Usuario: Policía boliviana	

Prioridad: Media	Estimación: 4
-------------------------	----------------------

Tabla 3.11 Registro de antecedentes

Fuente: Elaboración propia

A continuación se puede visualizar la historia de usuario para los datos estadísticos, mismo que fue solicitado para el administrador.

Id: 11	Historia de Usuario
Descripción: Como administrador quiero generar datos estadísticos de los conductores, tarjetas de identificación, unidades educativas y antecedentes por gestión y en general	
Usuario: Administrador	
Prioridad: Alta	Estimación: 3

Tabla 3.13 Datos estadísticos

Fuente: Elaboración propia

3.2.4 PRODUCT BACKLOG

En base a las historias de usuario y un análisis minucioso se elaboró una lista de todas las funcionalidades o requisitos del sistema, tomando en cuenta la prioridad, una descripción de la tarea, la estimación en función al tiempo, y el módulo al que corresponde. Como se observa en la tabla 3.14.

ID	PRIORIDAD	DESCRIPCIÓN	TIEMPO EN DIAS	MÓDULO
1	Alta	Diseño, modelado e implementación de la base de datos	10	
2	Alta	Registro, modificación y eliminación de los usuarios	7	Módulo de Gestión de usuarios
3	Alta	Autenticación de los usuarios	5	Módulo de Gestión de usuarios
4	Alta	Registro y administración de los		Módulo

		conductores con sus vehículos	5	Gestión de los conductores y vehículos
5	Media	Registro y administración de las unidades educativas	2	Módulo Gestión de los conductores y vehículos
6	Baja	Verificar los requisitos de afiliación	4	Módulo Gestión de los conductores y vehículos
7	Media	Registro de una nueva gestión	3	Módulo Gestión de los conductores y vehículos
8	Baja	Generar la tarjeta de identificación del conductor escolar	6	Módulo T.I.C.E.
9	Media	Visualizar datos estadísticos por gestión de las tarjetas	5	Módulo T.I.C.E.
10	Baja	Generar un código QR	3	Módulo T.I.C.E.
11	Media	Gestión de antecedentes	5	Módulo de antecedentes
12	Alta	Generar un reporte de los conductores, vehículos y establecimientos educativos	4	Módulo de reportes
13	Media	Buscar y mostrar datos de los conductores	5	Módulo de reportes
14	Media	Generar datos estadísticos de los conductores, tarjetas de identificación, unidades educativas y antecedentes por gestión y en general	7	Módulo de reportes

Tabla 3.14 Product Backlog

Fuente: Elaboración propia

3.2.5 DIAGRAMA DE CASOS DE USO

Este diagrama nos ayuda a identificar los actores que intervienen con el sistema, el proceso y las funcionalidades. En la figura 3.2 se puede observar el diagrama de casos de uso del negocio, más adelante en el *game* vamos a desglosar y desarrollar los módulos y funcionalidades.

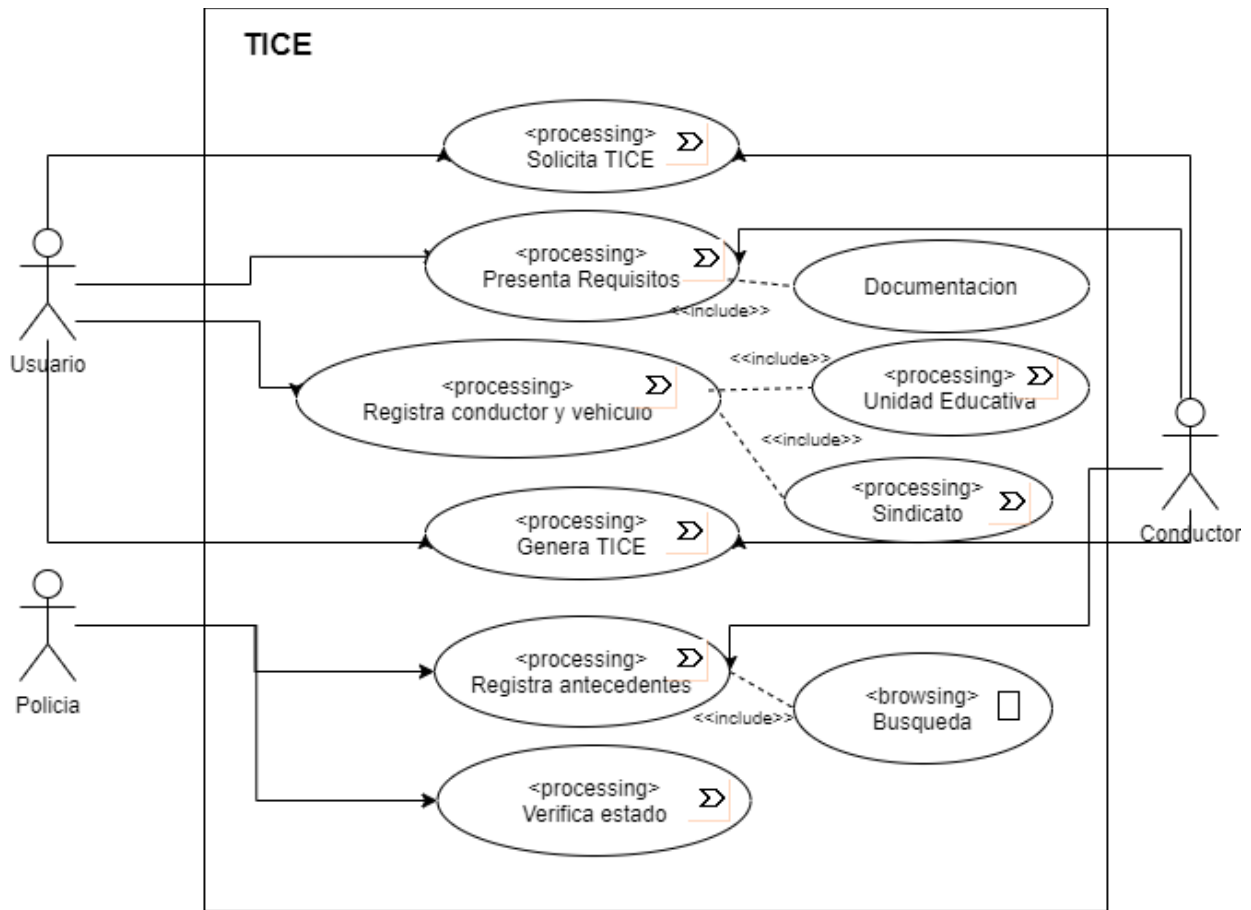


Figura 3.3. Diagrama de Casos de Uso
Fuente: Elaboración Propia

3.2.6 DIAGRAMA DE DATOS

3.2.6.1 MODELO ENTIDAD RELACIÓN

En la figura 3.4 se observa el modelo entidad relación del sistema, que es una herramienta

para el modelo de datos, con la ayuda de los requerimientos, los casos de uso y el diagrama de clases se realizó una percepción, colección de objetos denominados entidad.

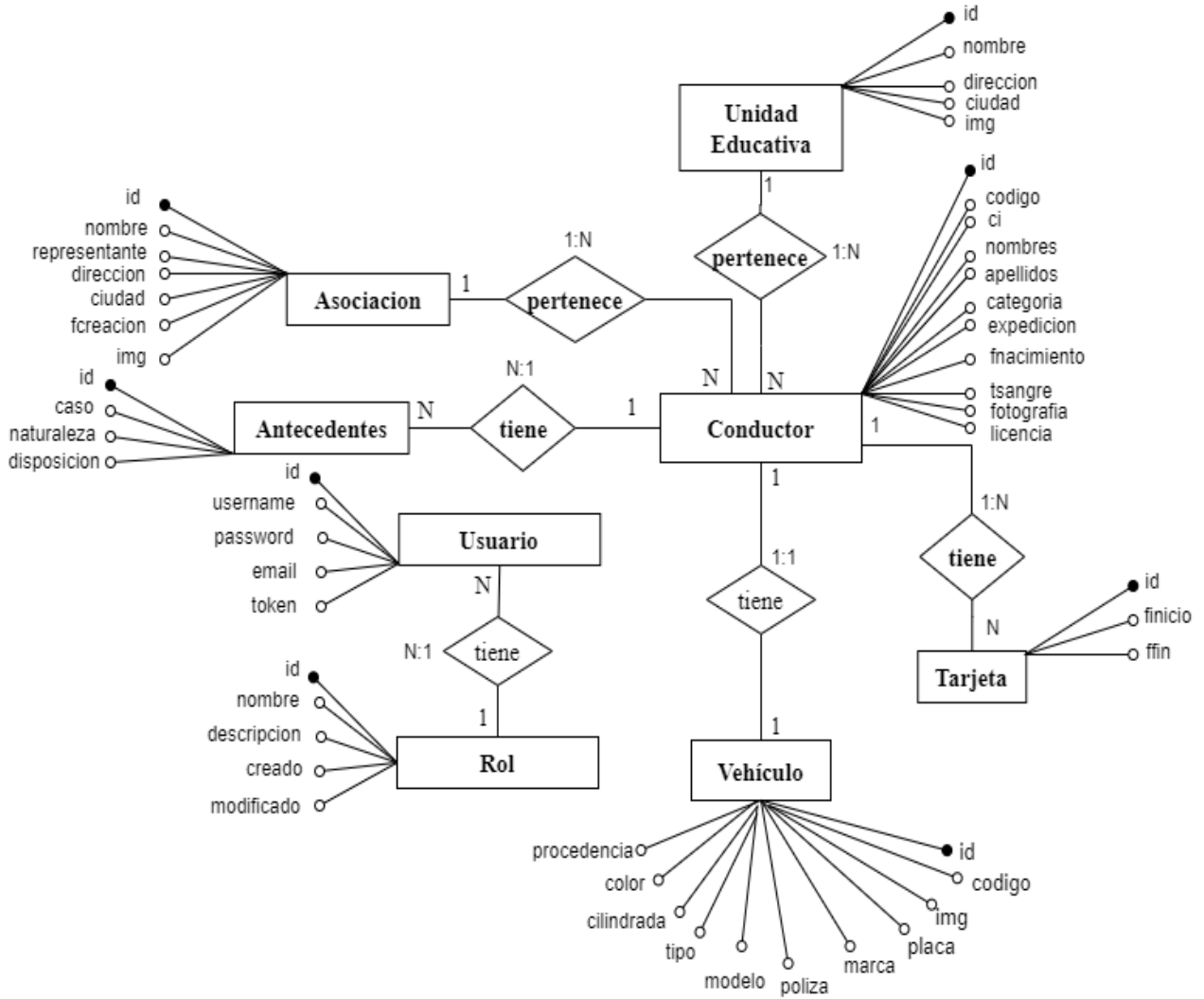


Figura 3.4 Modelo Entidad Relación

Fuente: Elaboración Propia

3.2.6.2 MODELO RELACIONAL

En la figura 3.5 se puede observar el modelo relacional para la gestión de base de datos, en este modelo todos los datos son almacenados en relaciones denominados tablas, por lo tanto, tenemos las tablas, atributos y relaciones necesarios para cumplir los requisitos del sistema.

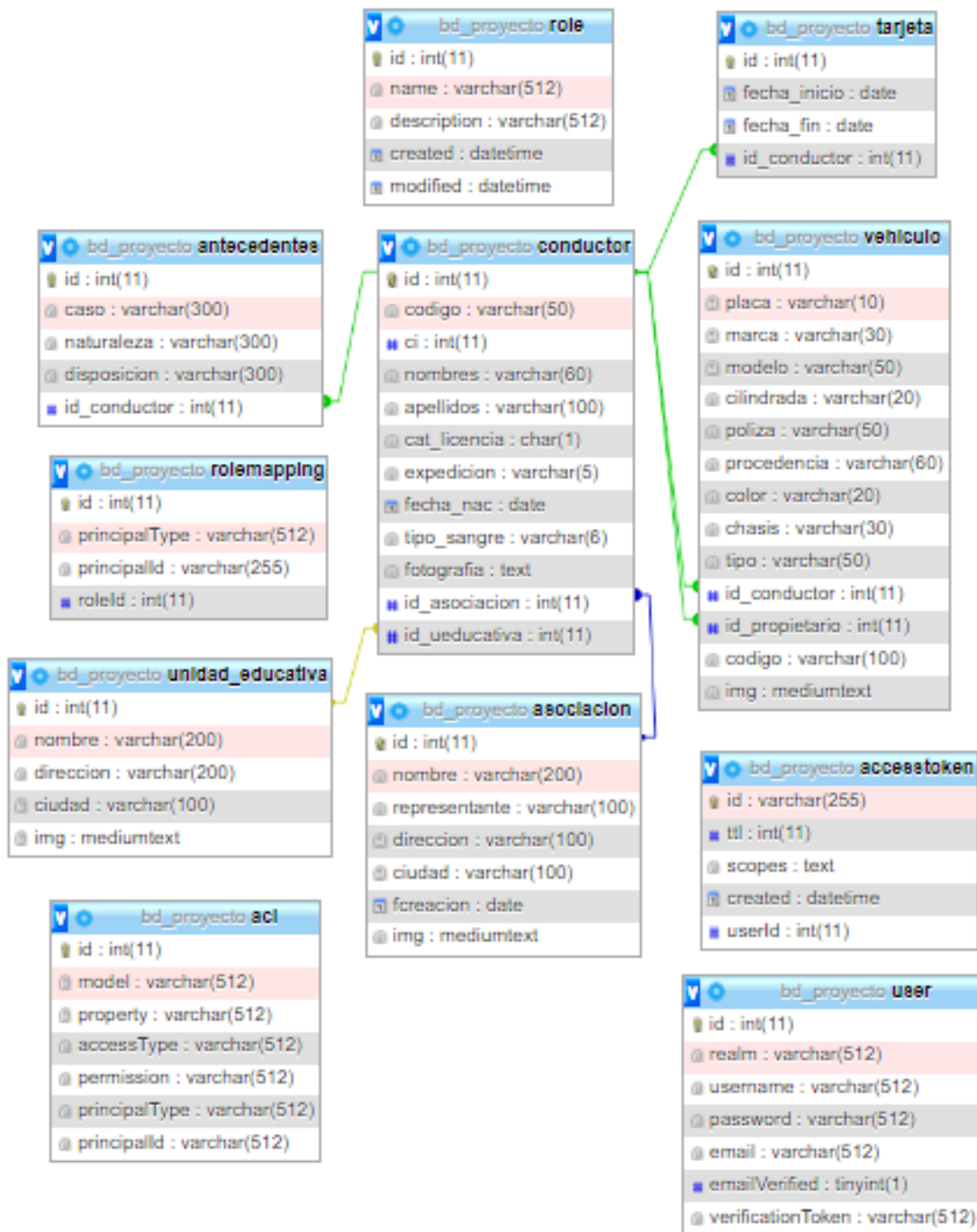


Figura 3.5 Modelo Relacional
Fuente: Elaboración Propia

3.3 GAME

En la etapa de desarrollo se hace un análisis y se prioriza la pila de producto, a partir de ahí se obtiene cada *sprint*, se realiza los modelos correspondientes hasta obtener un incremento, finalmente se realiza las pruebas. Una vez identificado los sprint de la lista *backlog* procedemos a la planificación, desarrollo y modelado para obtener el incremento y realizar las pruebas correspondientes.

3.3.1 SPRINT 1

Para este sprint se trabajó en el módulo de gestión de usuarios, identificando a quienes intervienen con el sistema, dando una fecha de inicio y una estimación del tiempo a culminar como se observa en la tabla 3.15.

Sprint	Módulo	Descripción	Fecha inicio	Duración días
Sprint 1	Módulo de gestión de usuario	Se ha identificado a 2 tipos de usuario que interviene de manera directa con el sistema el administrador y la policía, y 1 usuarios de manera indirecta los PPF	6/10/2021	22

Tabla 3.15 Sprint 1 Módulo de gestión de usuario
Fuente: Elaboración Propia

3.3.1.1 PLANIFICACIÓN

La planificación del presente sprint se puede observar en la tabla 3.16, tenemos el *product backlog* priorizado con las respectivas tareas asignadas y la estimación del tiempo.

Nro.	Product Backlog	Tareas	Duración	Estado
1	Gestión de usuarios	<ul style="list-style-type: none">• Definir los atributos y crear la base de datos• Registrar nuevo usuario• Modificar usuario• Eliminar usuario• Cambiar contraseña	5	Terminado

2	Asignación de roles	<ul style="list-style-type: none"> Definir los atributos y crear la base de datos Asignar rol al nuevo usuario 	3	Terminado
3	Autenticación de usuarios	<ul style="list-style-type: none"> Definir los atributos y crear la base de datos Autenticación de ingreso Validar ingreso Visualizar gestión de ingreso 	4	Terminado

Tabla 3.16 Planificación del Sprint 1

Fuente: Elaboración Propia

3.3.1.2 DESARROLLO

MODELO DE REQUERIMIENTOS

En la figura 3.6 se observa a 2 actores, los usuarios y los policías que intervienen con las funciones del sistema.

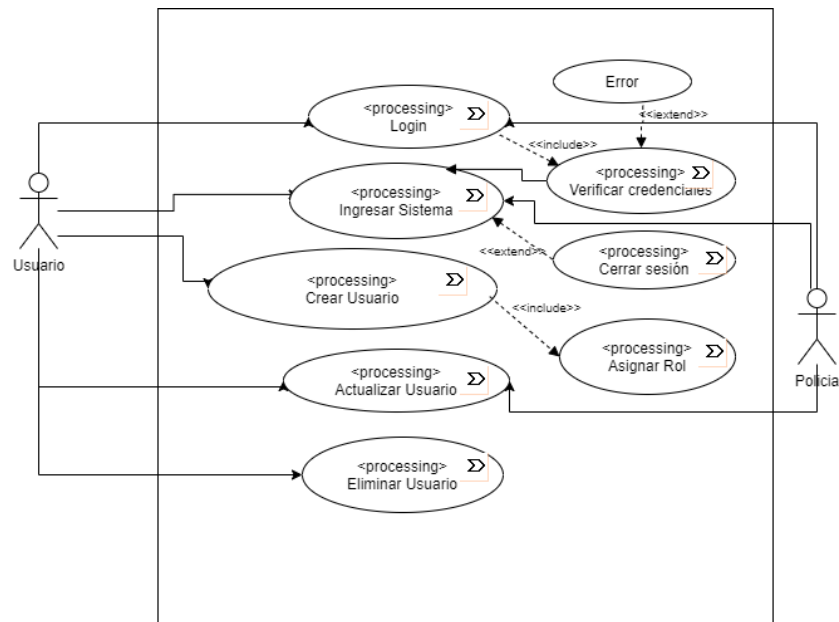


Figura 3.6 Modelo de Requerimientos

Fuente: Elaboración Propia

MODELO DE CONTENIDO

En la figura 3.7 se puede observar el modelo de contenido o diagrama de clases en UML para la gestión de usuarios

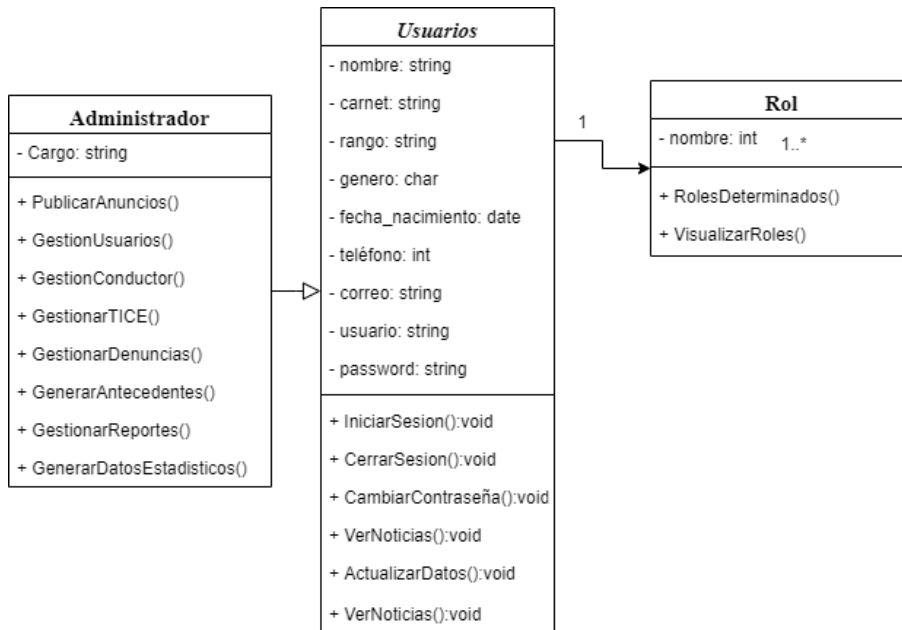


Figura 3.7 Modelo de Contenido
Fuente: Elaboración Propia

MODELO DE NAVEGACIÓN

En la figura 3.8 se observa el modelo de navegación desde el *login* hasta la administración de usuarios

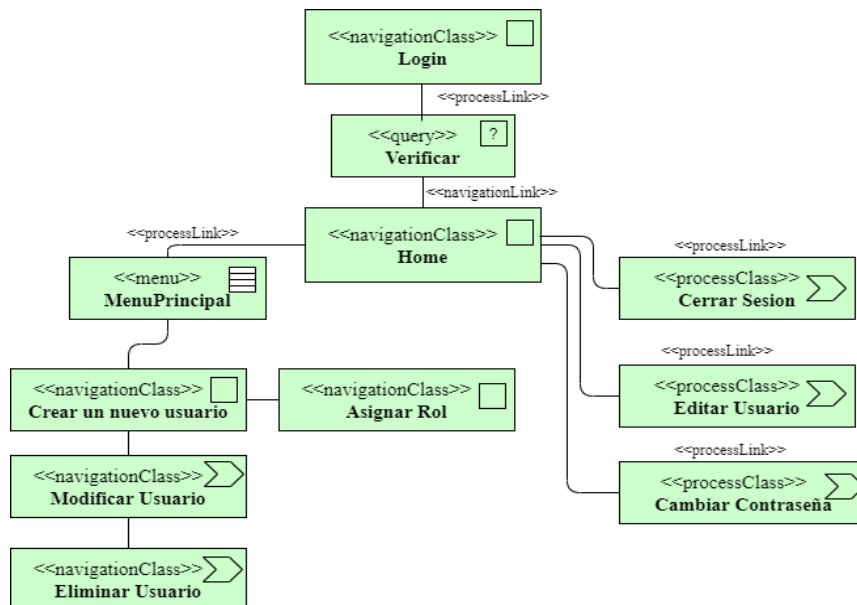


Figura 3.8 Modelo de Navegación
Fuente: Elaboración Propia

MODELO DE PRESENTACIÓN

En la figura 3.9 se observa el modelo de presentación para el acceso al sistema.

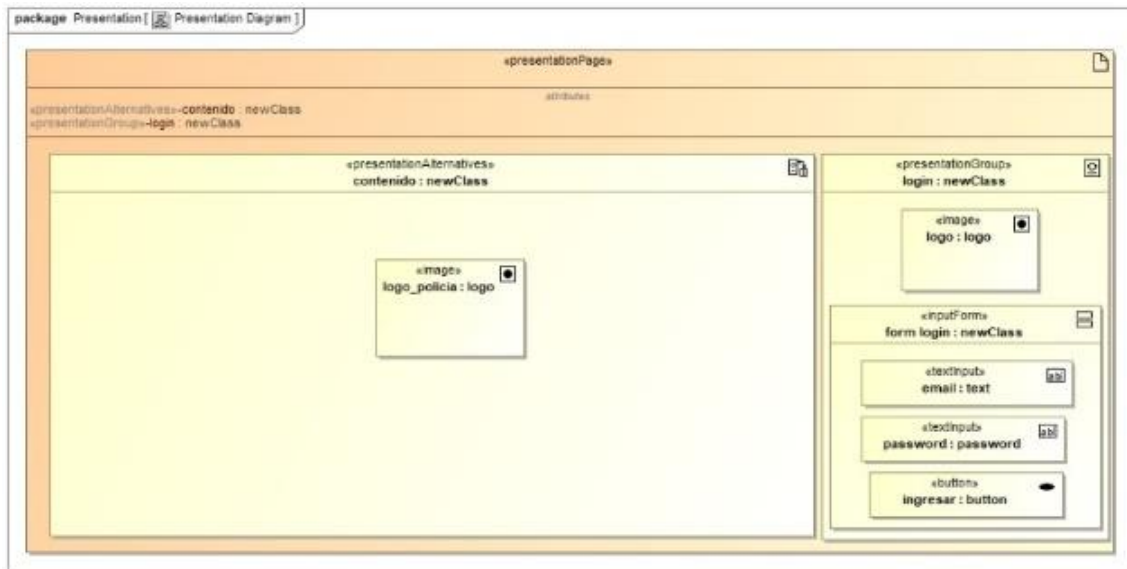


Figura 3.9 Modelo de Presentación para el Login

Fuente: Elaboración Propia

En la siguiente figura se puede visualizar el modelo de presentación para el menú principal.



Figura 3.10 Modelo de Presentación – Menú Principal

Fuente: Elaboración Propia

3.3.1.3 PRUEBAS

Se ha realizado una evaluación del presente módulo con las tareas asignadas obteniendo los siguientes resultados, como se muestra en la tabla 3.17.

PRUEBAS DE CAJA NEGRA		
Nro.	Entrada	Salida
1	Se tiene las tablas necesarias, los atributos y las relaciones correspondientes en la base de datos	Cumple
2	Registra, modifica y elimina a los usuarios	Cumple
3	Asignar roles a los usuarios	Cumple
4	Valida y verifica las credenciales del login	Cumple
5	Inicia sesión según el tipo de usuario	Cumple
6	Visualiza a los usuarios según el tipo	Cumple
7	Genera un reporte general de todos los usuarios	Cumple

Tabla 3.17 Pruebas y resultados obtenidos del Sprint 1

Fuente: Elaboración Propia

3.3.2 SPRINT 2

Para este sprint se trabajó en el módulo de gestión de los conductores y vehículos, dando una fecha de inicio y una estimación del tiempo a culminar como se observa en la tabla 3.20.

Sprint	Módulo	Descripción	Fecha inicio	Duración días
Sprint 2	Módulo de Gestión de los conductores y vehículos	Registro de conductores, vehículos, digitalización de los requisitos de filiación y reportes	05/11/2021	14

Tabla 3.18 Modulo de Gestión de los conductores y vehículos

Fuente: Elaboración Propia

3.3.2.1 PLANIFICACIÓN

La planificación del presente sprint se puede observar en la tabla 3.19, tenemos la gestión de los conductores y vehículos con las respectivas tareas asignadas y la estimación del tiempo.

Nro.	Product Backlog	Tareas	Duración	Estado
1	Registro y administración de los conductores con sus vehículos	<ul style="list-style-type: none"> • Definir los atributos y crear la base de datos • Registrar, modificar y eliminar a los conductores • Registrar, modificar y eliminar a los vehículos. 	4	Terminado
2	Registro y administración de las unidades educativas	<ul style="list-style-type: none"> • Definir los atributos y crear la base de datos • Registrar, modificar y eliminar a las unidades educativas 	3	Terminado
3	Verificación de los requisitos de afiliación	<ul style="list-style-type: none"> • Definir los atributos y crear la base de datos • Verificar los requerimientos de afiliación • Validar los datos y registrar 	4	Terminado
4	Registro de una nueva gestión	<ul style="list-style-type: none"> • Definir los atributos y crear la base de datos • Registrar, modificar y eliminar una nueva gestión 	3	Terminado

Tabla 3.19 Planificación del Sprint 2
Fuente: Elaboración Propia

3.3.2.2 DESARROLLO

MODELO DE REQUERIMIENTOS

En la figura 3.11 se observa a 2 actores, el usuario o encargado de la unidad y los conductores con las funciones que interviene al sistema, para este caso la gestión de los conductores y vehículos.

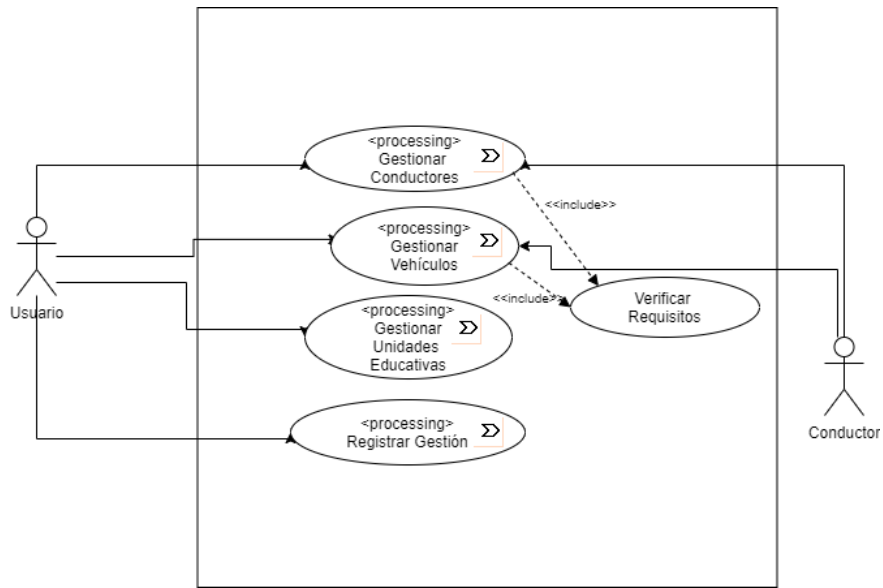


Figura 3.11 Modelo de Requerimientos
Fuente: Elaboración Propia

MODELO DE CONTENIDO

En la figura 3.12 se puede observar el modelo de contenido o diagrama de clases en UML, para la gestión de los conductores y vehículos

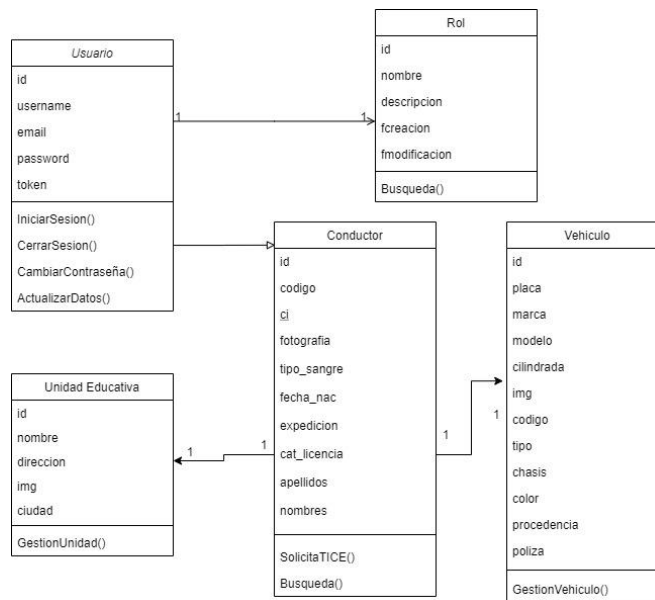


Figura 3.12 Modelo de Contenido
Fuente: Elaboración Propia

MODELO DE NAVEGACIÓN

En la figura 3.13 se observa el modelo de navegación para la gestión de los conductores y vehículos.

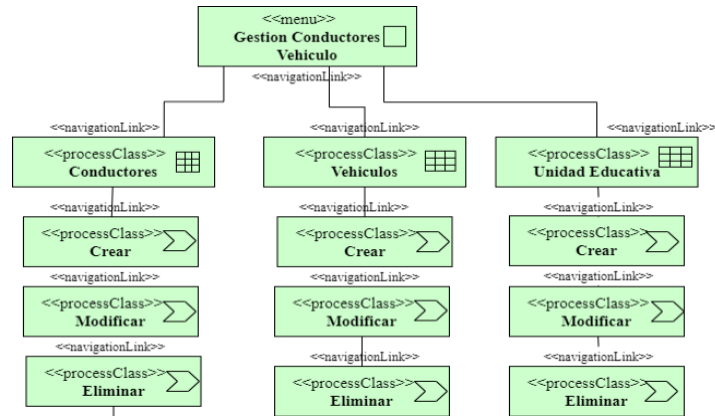


Figura 3.13 Modelo de Navegación
Fuente: Elaboración Propia

MODELO DE PRESENTACIÓN

En la figura 3.14 se observa el modelo de presentación la interfaz de usuario

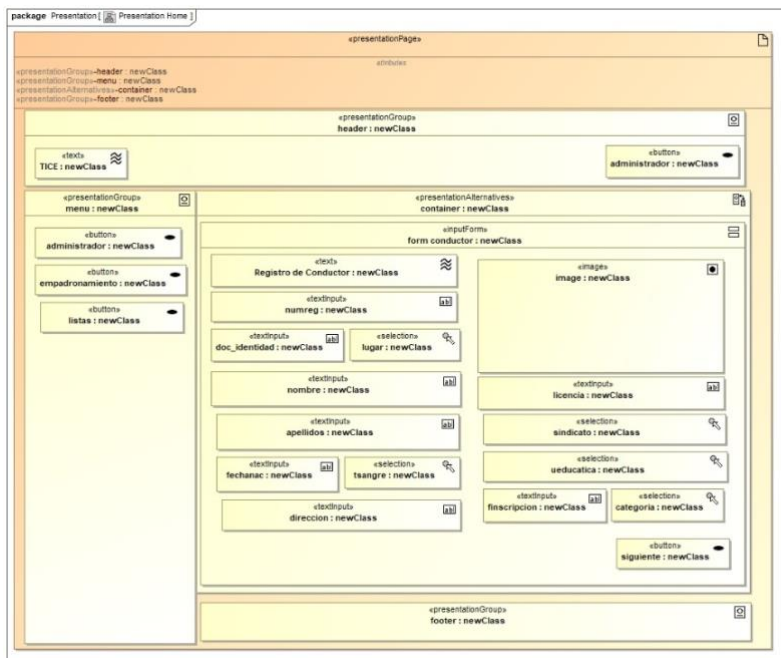


Figura 3.14 Modelo de presentación
Fuente: Elaboración Propia

3.3.2.3 PRUEBAS

Se ha realizado una evaluación del presente módulo con las tareas asignadas obteniendo los siguientes resultados, como se muestra en la tabla 3.20

PRUEBAS DE CAJA NEGRA		
Nro.	Entrada	Salida
1	Se tiene las tablas necesarias, los atributos y las relaciones correspondientes en la base de datos	Cumple
2	Registra, modifica y elimina a los conductores	Cumple
3	Registra, modifica y elimina a los vehículos	Cumple
4	Registra, modifica y elimina a las unidades educativas	Cumple
5	Busca a los conductores	Cumple
6	Verificar los requisitos de afiliación	Cumple
7	Actualizar los requisitos	Cumple

Tabla 3.20 Pruebas y resultados obtenidos del Sprint 2

Fuente: Elaboración Propia

3.3.3 SPRINT 3

Para este sprint se trabajó en el módulo tarjeta de identificación, dando una fecha de inicio y una estimación del tiempo a culminar como se observa en la tabla 3.21

Sprint	Módulo	Descripción	Fecha inicio	Duración días
Sprint 3	Módulo tarjeta de identificación y reportes	Generar la tarjeta de identificación del conductor escolar (T.I.C.E.)	26/11/2021	30

Tabla 3.21 Módulo tarjeta de identificación

Fuente: Elaboración Propia

3.3.3.1 PLANIFICACIÓN

La planificación del presente sprint se puede observar en la tabla 3.22, tenemos la gestión tarjeta de identificación con las respectivas tareas asignadas y la estimación del tiempo.

Nro.	Product Backlog	Tareas	Duración	Estado
1	Gestión periodo	<ul style="list-style-type: none"> Definir los atributos y crear la base de datos Registrar, modificar y eliminar una nueva gestión 	3	Terminado
2	Generar la tarjeta de identificación del conductor escolar	<ul style="list-style-type: none"> Generar la tarjeta de identificación 	4	Terminado
3	Visualizar datos estadísticos por gestión de las tarjetas	<ul style="list-style-type: none"> Mostrar datos estadísticos por género, por gestión, por antecedentes, por tipo de vehículo. 	4	Terminado
4	Generar un código QR	<ul style="list-style-type: none"> Generar un número único Mediante una librería generar un código QR para incorporar a la tarjeta 	3	Terminado
5	Generar reporte de conductores	<ul style="list-style-type: none"> Generar el pdf que contiene el reporte de conductores. 	1	Terminado
6	Generar reporte de unidades educativas	<ul style="list-style-type: none"> Generar el pdf que contiene el reporte de conductores. 	1	Terminado
7	Buscar y mostrar datos de los conductores	<ul style="list-style-type: none"> Generar funcionalidad para la búsqueda de conductor por cualquier campo 	2	Terminado
8	Buscar y mostrar datos de las unidades educativas.	<ul style="list-style-type: none"> Generar funcionalidad para la búsqueda de conductor por cualquier campo 	2	Terminado
9	Generar datos estadísticos.	<ul style="list-style-type: none"> Generar estadística de cantidad de conductores por sindicato Generar estadística de cantidad de tarjetas generadas por mes Generar estadística de tipos de vehículos registrados 	10	Terminado

		<ul style="list-style-type: none"> • Generar cantidad de conductores con y sin antecedentes 		
--	--	--	--	--

Tabla 3.22 Planificación del Sprint 3

Fuente: Elaboración Propia

3.3.3.2 DESARROLLO

MODELO DE REQUERIMIENTOS

En la figura 3.15 se observa a 2 actores, el usuario encargado de la unidad y los conductores con las funciones que interviene al sistema, para este caso tenemos la gestión de tarjeta de identificación

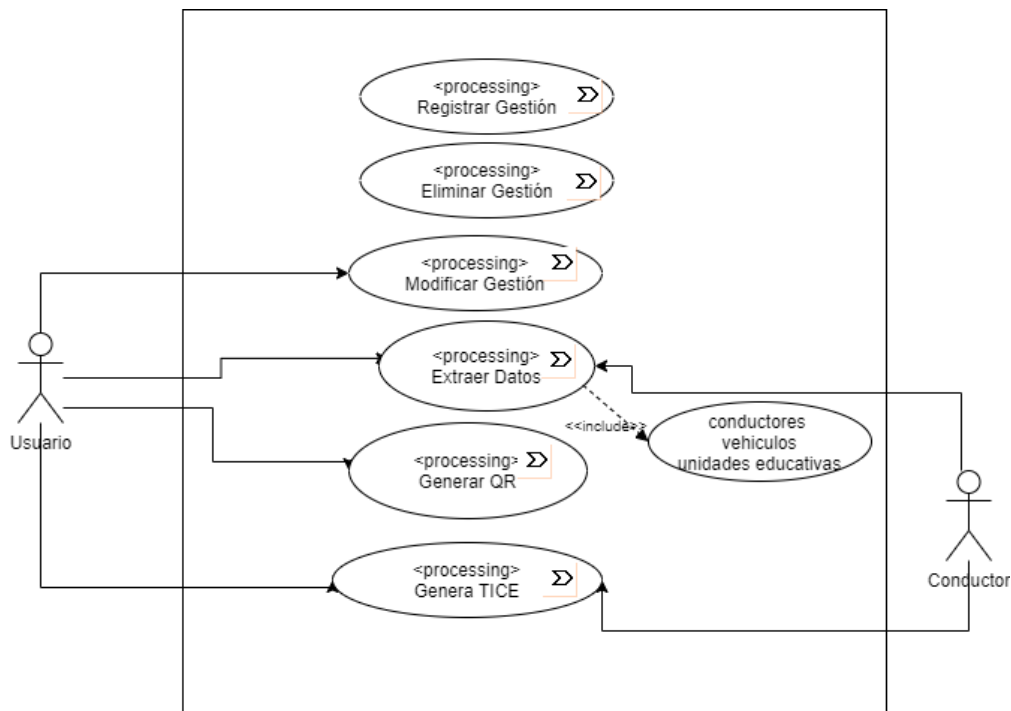


Figura 3.15 Modelo de Requerimientos

Fuente: Elaboración Propia

MODELO DE CONTENIDO

En la figura 3.16 se puede observar el modelo de contenido o diagrama de clases en UML, para la gestión de la tarjeta de identificación.

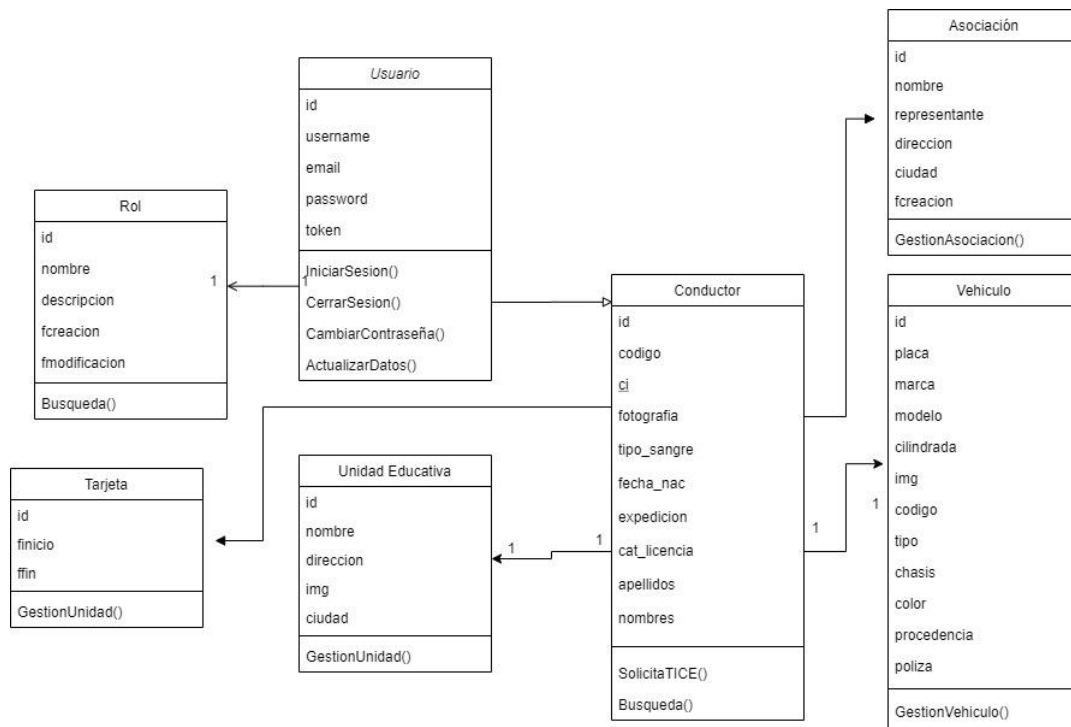


Figura 3.16 Modelo de Contenido
Fuente: Elaboración Propia

MODELO DE NAVEGACIÓN

En la figura 3.17 se observa el modelo de navegación para el módulo de tarjeta de identificación.

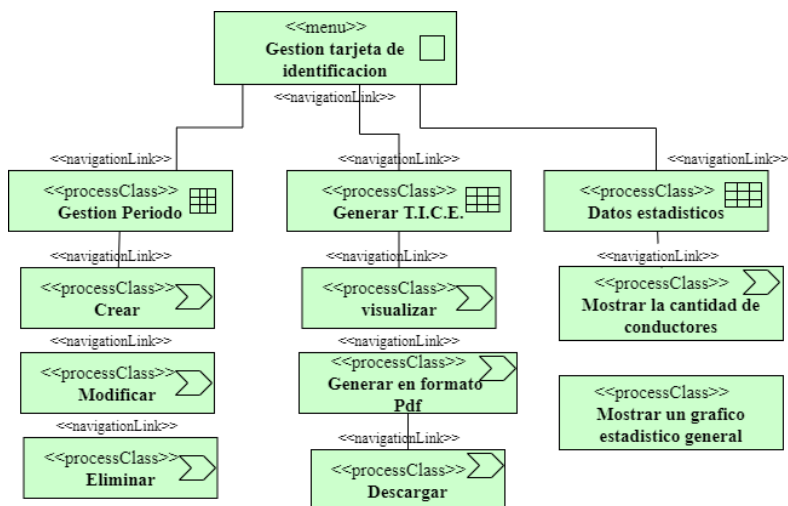


Figura 3.17 Modelo de Navegación
Fuente: Elaboración Propia

MODELO DE PRESENTACIÓN

En la figura 3.18 se observa el modelo de presentación, el administrativo realiza la inscripción, se visualiza un cronograma de rotación y se genera reportes.

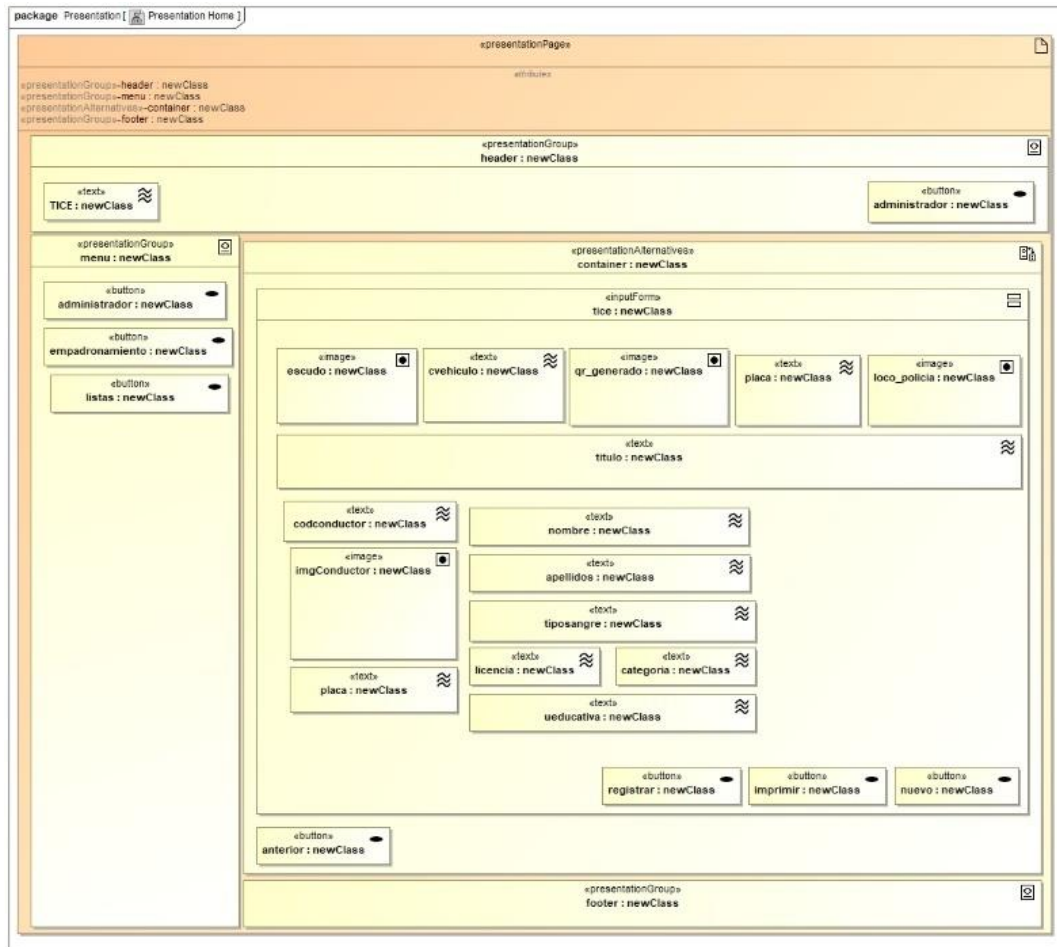


Figura 3.18 Modelo de Presentación
Fuente: Elaboración Propia

3.3.3.3 PRUEBAS

Se ha realizado una evaluación del presente módulo con las tareas asignadas obteniendo los siguientes resultados, como se muestra en la tabla 3.23.

PRUEBAS DE CAJA NEGRA		
Nro.	Entrada	Salida
1	Se tiene las tablas necesarias, los atributos y las relaciones correspondientes en la base de datos	Cumple
2	Registra, modifica y elimina la gestión de la gestión	Cumple
3	Generar la tarjeta de identificación.	Cumple
4	Descargar y registrar la tarjeta de identificación	Cumple
5	Visualizar y generar datos estadísticos	Cumple

Tabla 3.23 Pruebas y resultados obtenidos del Sprint 3

Fuente: Elaboración Propia

3.4 POST GAME

Una vez culminada la planificación, el desarrollo y las pruebas de cada sprint con las respectivas iteraciones, esta es la última etapa en la metodología SCRUM, donde vemos el acceso de usuario y la presentación de interfaz de cada módulo.

3.4.1 INTERFAZ DE USUARIO

3.4.1.1 MODULO DE GESTIÓN DE USUARIOS

En la figura 3.19 se puede apreciar el formulario de inicio de sesión, en el cual el usuario, cualquiera sea su rol puede acceder al sistema con sus credenciales e interactuar con él.



Figura 3.19 Formulario de inicio de sesión

Fuente: Elaboración propia

Ya iniciada la sesión de usuario, el sistema interactúa con el mismo a partir de la selección de cada uno de los ítems presentes en el menú lateral izquierdo.

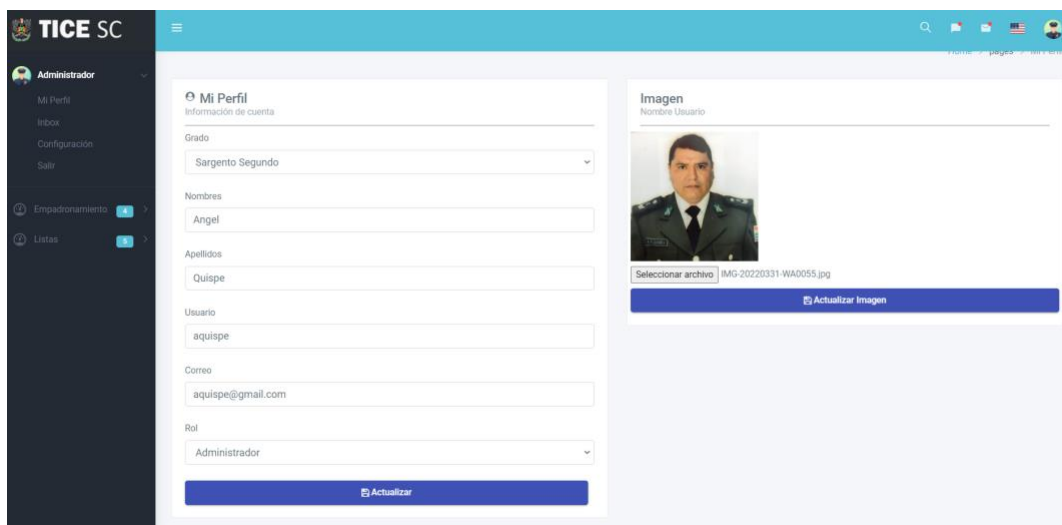


Figura 3.20 Menú Principal
Fuente: Elaboración propia

3.4.1.2 MODULO DE GESTIÓN DE CONDUCTORES Y VEHÍCULOS

En la figura 3.21 se puede apreciar el formulario de registro de conductores.

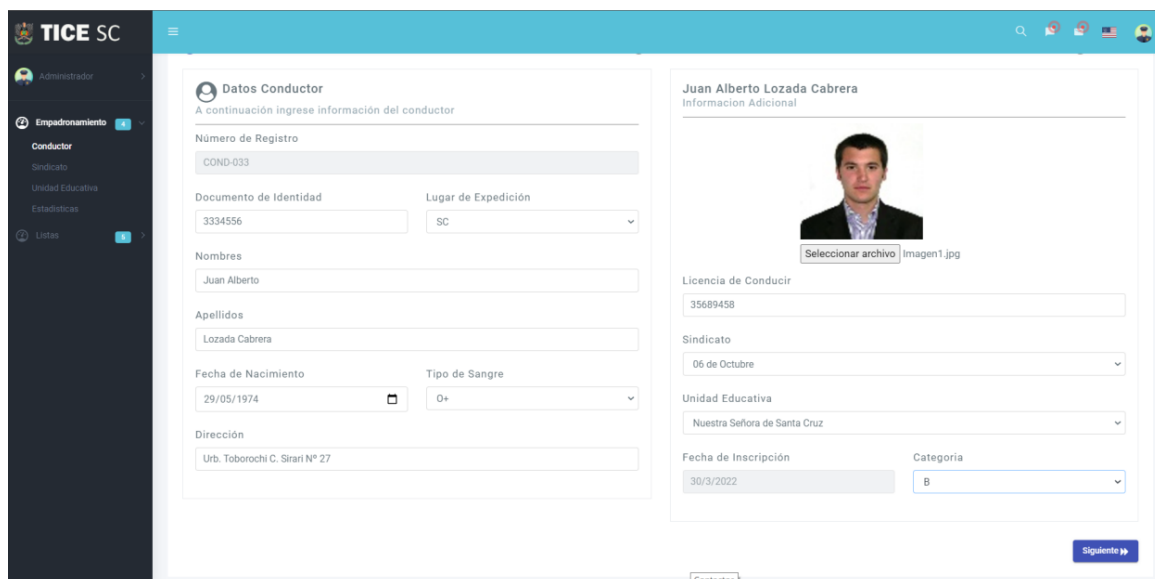


Figura 3.21 Formulario de Registro de Conductor
Fuente: Elaboración propia

En la figura 3.22 se muestra el formulario de registro de vehículo que juntamente con el registro de conductor (Ver imagen 3.21) dan origen a la generación de la tarjeta TICE.

Figura 3.22 Formulario de Registro de Vehículo
Fuente: Elaboración propia

Una vez registrada la información podemos realizar la búsqueda de la información que se desee en el listado de conductores que ofrece el sistema.

#	Conductor	Cédula	Unidad Educativa	Sindicato	Placa	Tipo	Fecha de Registro	Acciones
1	Angel Quispe Alanoca	123456789	Colegio Jesus Obrero	Asoc Los Amigos	124ABC	Bus Escolar	3/19/22, 8:00 PM	[Red X] [Blue Edit]
2	Juan Alberto Lozada Cabrera	3334556	Nuestra Señora De Santa Cruz	06 de Octubre	1245ASD	Taxi	3/19/22, 8:00 PM	[Red X] [Blue Edit]
3	Enrique Benavides Torrez	8256421	Pedro Domingo Murillo	06 de Octubre	3025AFB	Otros	3/19/22, 8:00 PM	[Red X] [Blue Edit]
4	Ramiro Sierra Urquiza	100652	Colegio Fuerza Aérea Argentina	14 de Septiembre	4112THD	Minibus	3/20/22, 8:00 PM	[Red X] [Blue Edit]
5	Irineo Jimenez Lima	633254	Pedro Domingo Murillo	14 de Septiembre	255RED	Bus Escolar	3/20/22, 8:00 PM	[Red X] [Blue Edit]

Figura 3.23 Lista de Conductores
Fuente: Elaboración propia

3.4.1.3 MODULO DE GESTIÓN DE TARJETA TICE

En la figura 3.24 se puede apreciar la tarjeta TICE ya generada a partir de la información introducida de conductor y vehículo, así también la puesta a impresión de la misma.



Figura 3.24 Tarjeta TICE generada
Fuente: Elaboración propia

3.4.1.4 MODULO DE REPORTE

A partir de los listados existentes, el usuario puede generar e imprimir los reportes de usuarios, conductores, vehículos, sindicatos, unidades educativas, etc. Como se muestra en la figura 3.25.

#	Conductor	Cedula	Unidad Educativa	Medico	Placa	Tipo	Fecha de Registro	Acciones
1	234234 234234	324234	Colegio Jesus Obispo	06 de Octubre	abf	1	3/30/22, 8:00 PM	
2	wentse wew	2342432	Colegio Jesus Obispo	Angelos del infierno	eddfdf		1/05/22, 8:00 PM	
3	Rosario Quispe Alarcos	10099422	Pedro Domingo Murillo	Calamares	12ABC	2	1/30/22, 8:00 PM	
4	and and	78785	Colegio Jesus Obispo	level 3	54abcd	1	2/18/22, 8:00 PM	
5	Pepeita Perez	123456	Colegio Jesus Obispo	14 de Septiembre	123ABC	Truck	2/18/22, 8:00 PM	

Figura 3.25 Reporte de Conductores
Fuente: Elaboración propia

Así también se cuenta con los reportes estadísticos como se puede apreciar en la figura 3.26.

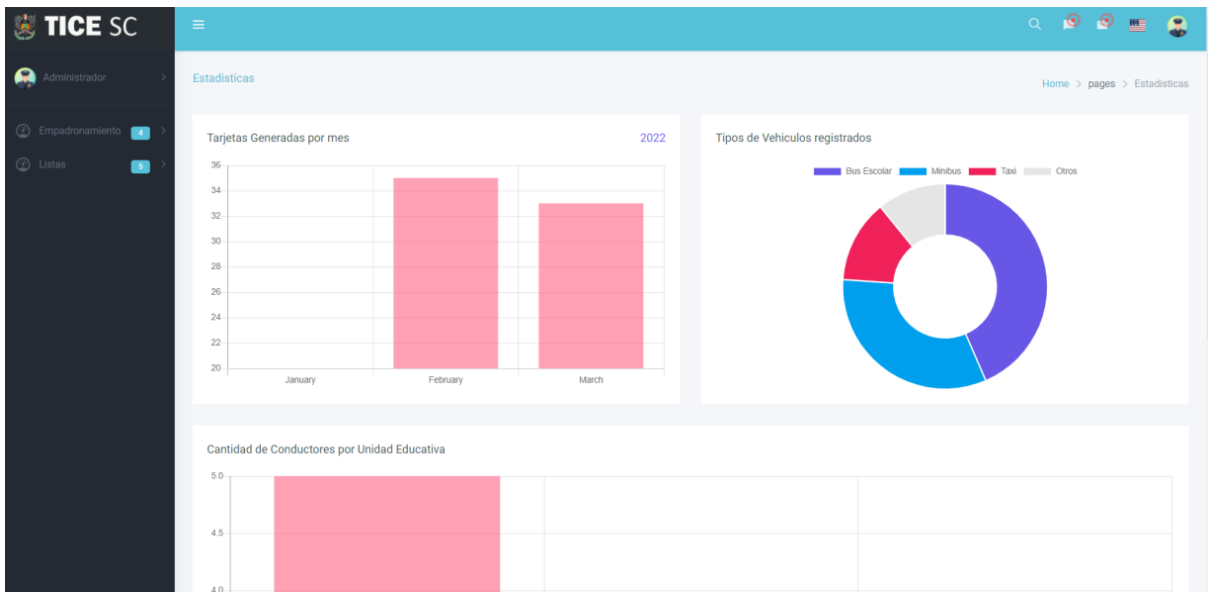


Figura 3.26 Reportes estadísticos
Fuente: Elaboración propia

3.4.2 PRUEBAS DE STRESS

Las pruebas de Stress se utilizan para encontrar el volumen de datos o de tiempo en que la

aplicación comienza a fallar o es incapaz de responder a las peticiones. Son pruebas de carga o rendimiento, pero superando los límites esperados en el ambiente de producción y/o determinados en las pruebas. Estas pruebas fueron realizadas en *Webserver Stress Tool*.

A continuación, se muestra el tiempo de conexión de los usuarios al sistema y se puede observar que el uso de la memoria del sistema puede llegar a 100%, pero después de un tiempo medido en segundos, el uso de la memoria va disminuyendo y se puede observar que al llegar a los 500 [s], al finalizar una petición el uso de la memoria cae al 0%.

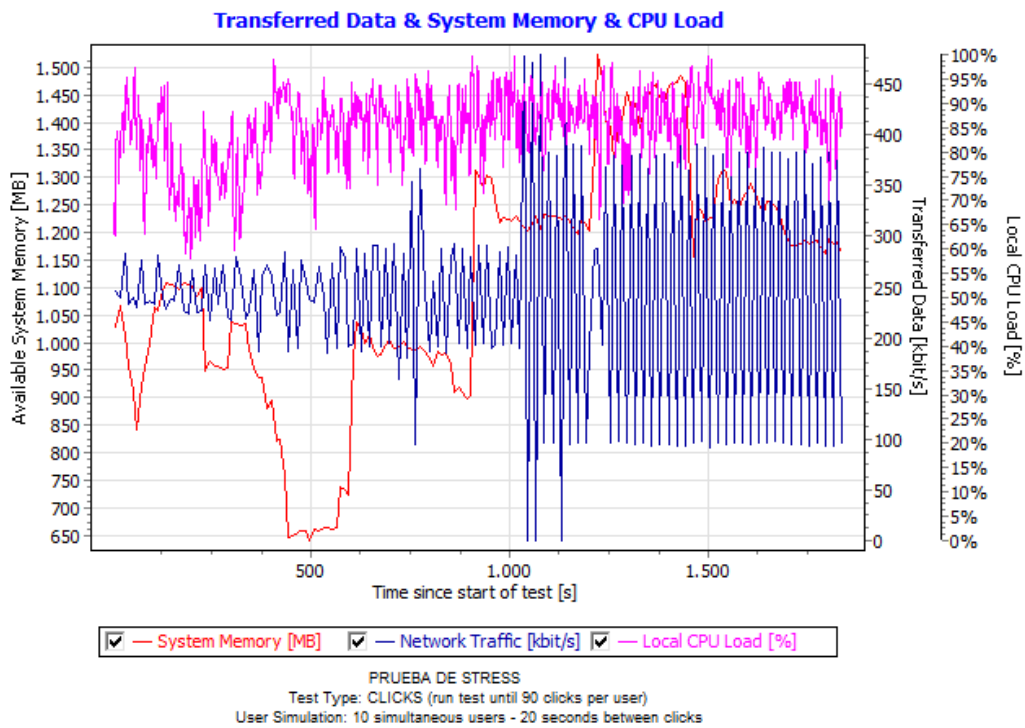


Figura 3.27. Datos transferidos, memoria del sistema y carga de la CPU

Fuente: Elaboración propia

La siguiente figura muestra la transferencia de datos y el tiempo de respuesta del sistema a las peticiones del usuario.

La línea roja indica los tiempos de petición del usuario, la línea verde nos indica el tiempo de respuesta del servidor, la línea celeste representa el tiempo de recibo de respuesta del

servidor, y la línea azul representa el tráfico de datos de kb/s del servidor. Donde se puede observar que el servidor responde respecto al tiempo de acuerdo con el tamaño de la petición del usuario y este se comporta de manera similar.

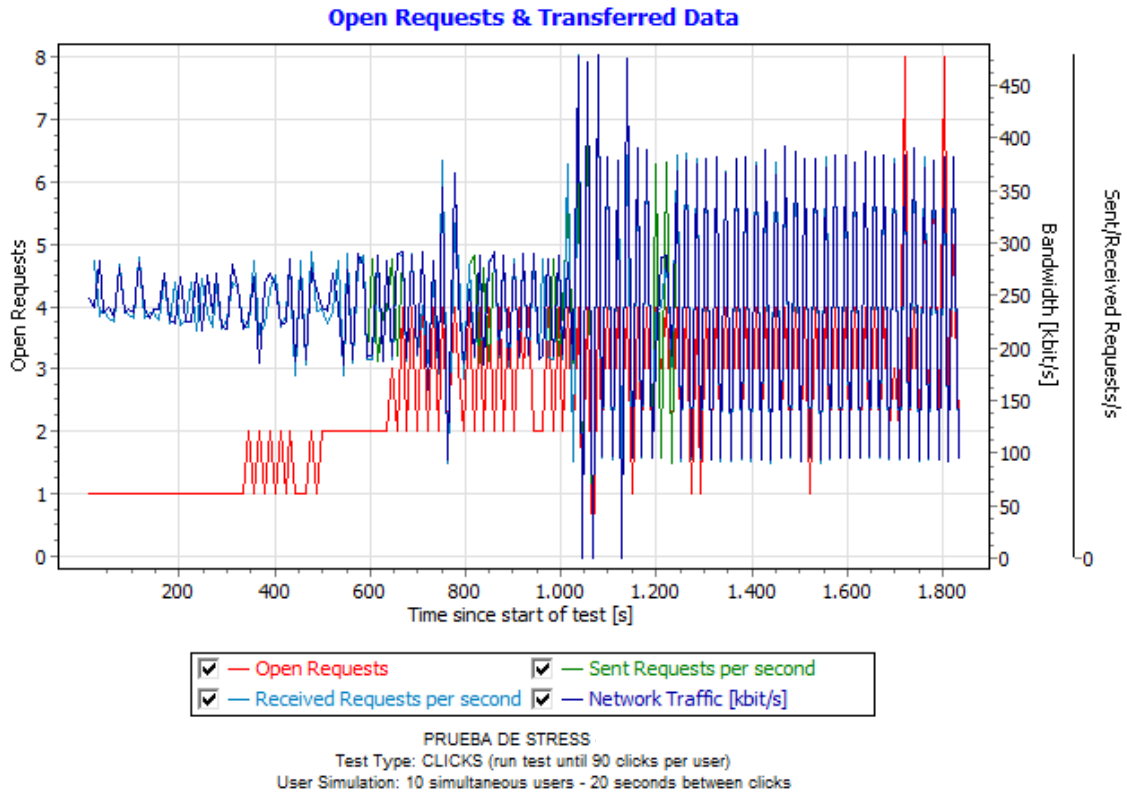


Figura 3.28. Solicitud de apertura y transferencia de datos
Fuente: Elaboración propia

En la siguiente figura se muestra las peticiones de usuario hecho al sistema en milisegundos, donde se puede observar, que el resultado obtenido no presenta errores en peticiones de página durante los 1800000 [ms]

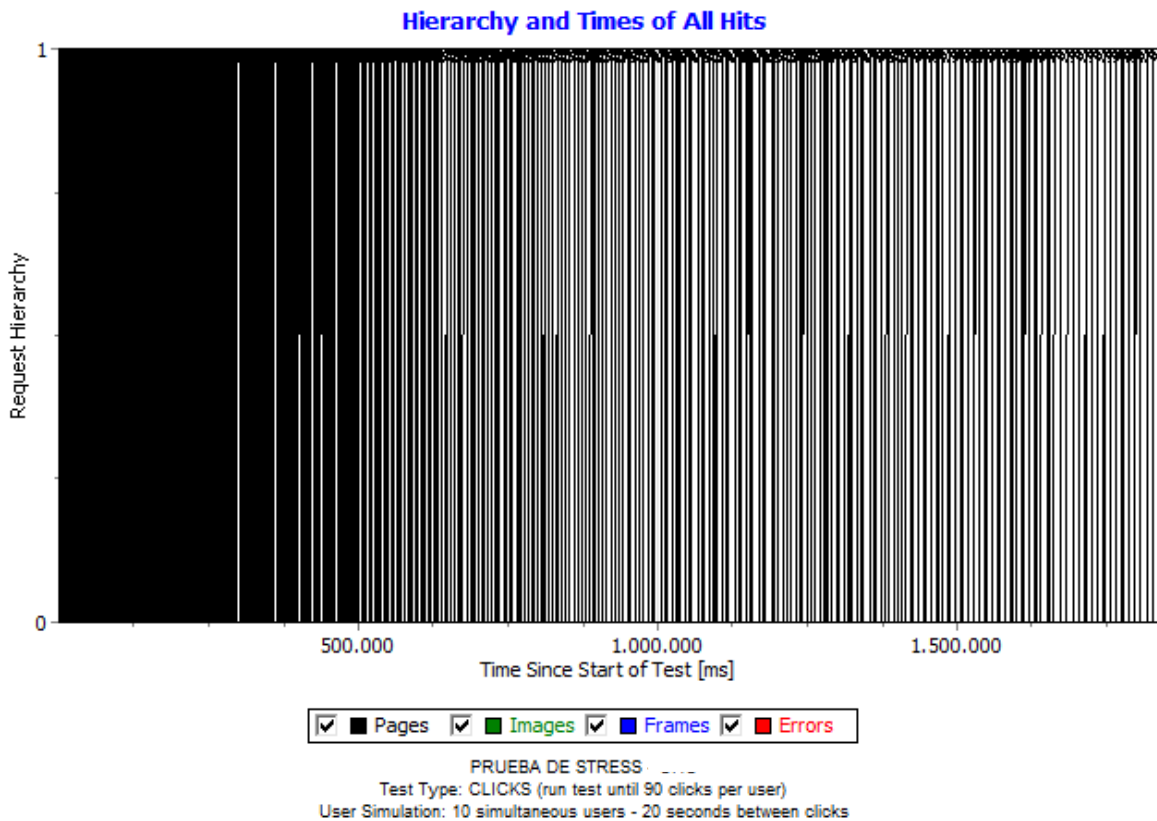


Figura 3.29 Peticiones del usuario

Fuente: Elaboración propia

En la siguiente figura se puede observar la capacidad máxima de uso simultáneo de usuarios activos. Se puede observar que la línea azul inicia desde el tiempo 0 [s] hasta los 80 [s], durante este tiempo, los usuarios activos van aumentando de forma exponencial y el punto de flexión de la línea azul con la intersección con la línea verde nos indica el tiempo mínimo de respuesta aceptable del servidor.

La línea azul al llegar al punto de embotellamiento *gridlock*, nos indica la capacidad, máxima de carga de usuarios activos simultáneamente.

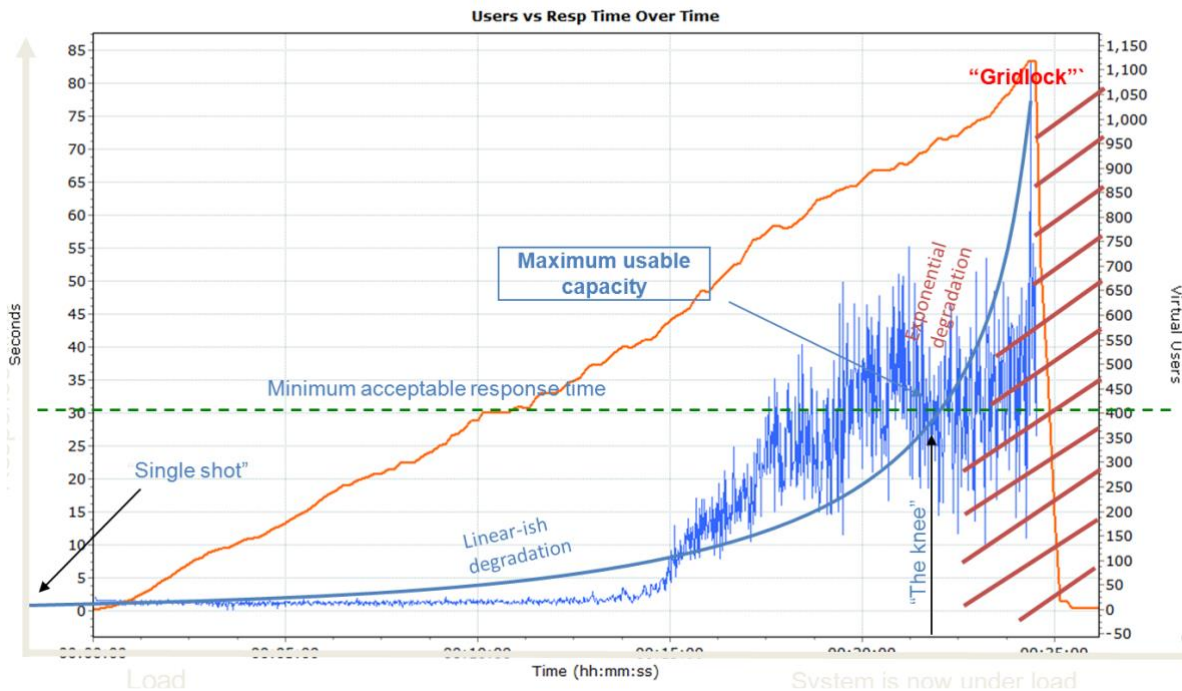


Figura 3.30 Capacidad de carga del sistema

Fuente: Elaboración propia

Por lo tanto, el sistema web con la simulación realizada con usuarios virtuales, se puede concluir que la capacidad de tiempo de respuesta es estable con cada usuario como se puede observar en la figura 3.30, además de que el servidor es capaz de responder a un máximo de 1119 usuarios activos simultáneamente y que el tiempo y usuarios mínimos aceptables es de 30 segundos con la capacidad de 400 usuarios activos simultáneamente.

4. CAPITULO IV CALIDAD Y SEGURIDAD DE SOFTWARE

4.1 INTRODUCCIÓN

El desarrollo de software tiene un fuerte crecimiento, para lo cual evaluar la calidad del producto se ha convertido en un requerimiento esencial, se ha implementado la **ISO/IEC 25010**, que detalla el modelo de calidad tanto del producto como de la calidad en uso, así mismo nos apoyaremos en el modelo de medición **GQM**, se basa en recopilar los datos medidos y usar eficazmente las métricas para mejorar la toma de decisiones. (Calabrese y Muños, 2018)

“Desarrollar un software con calidad implica la utilización de estándares, metodologías y procesos para el análisis, el diseño, la programación y las pruebas, con el fin de lograr confiabilidad, efectividad y productividad en el control de la calidad. Es por ello que medir la calidad de un producto de software implica evaluar un producto final y los procesos que se siguieron para llegar a este.” (Calabrese y Muños, 2018, p. 13)

4.2 NORMA ISO/IEC 25010

Es un modelo que representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto, es decir, se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios. (ISO-25010, 2021)

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las siguientes características y subcaracterísticas como se muestra en la tabla 4.1

CARACTERISTICA	SUBCARACTERISTICA
Adecuación funcional	Compleitud funcional
	Corrección funcional
	Pertinencia funcional
Eficiencia de desempeño	Capacidad
	Comportamiento temporal

	Utilización de recursos
Compatibilidad	Coexistencia
	Interoperabilidad
Usabilidad	Accesibilidad
	Aprendizaje
	Estética
	Inteligibilidad
	Operabilidad
	Protección frente a errores
Fiabilidad	Capacidad de recuperación
	Disponibilidad
	Madurez
	Tolerancia a fallos
Seguridad	Autenticidad
	Confidencialidad
	Integridad
	No repudio
	Responsabilidad
Mantenibilidad	Analizabilidad
	Capacidad de ser modificado
	Capacidad de ser probado

	Modularidad
	Reusabilidad
Portabilidad	Adaptabilidad
	Capacidad de ser reemplazado
	Facilidad de instalación

Tabla 4.1 Características y subcaracterísticas definidas en ISO/IEC 25010
Fuente: (Calabrese y Muños, 2018)

4.3 GQM

El modelo de medición GQM tiene tres niveles:

Nivel Conceptual (Objetivo): Identifican lo que se quiere lograr respecto a los productos, procesos o recursos, con respecto a varios modelos de calidad, desde varios puntos de vista y relativo a un entorno particular.

Nivel Operativo (Pregunta): Se refiere a un conjunto de preguntas a partir del objetivo, con el propósito de verificar su cumplimiento. Las preguntas tratan de caracterizar el objetivo de medición con respecto a una cuestión de calidad.

Nivel Cuantitativo (Métrica): Se asocia un conjunto de datos para cada pregunta formulando métricas, con el fin de proporcionar una respuesta de manera cuantitativa

Para obtener la calidad del producto se debe establecer los requisitos de la evaluación y especificar la evaluación. Esto consiste en seleccionar las características y subcaracterísticas del modelo ISO/IEC 25010.

4.4 APLICACIÓN DE LA NORMA ISO/IEC 25010 Y GQM

Se ha tomado una muestra de 15 usuarios encuestados como se observa en la tabla 4.2.

USUARIOS	NRO. ENCUESTADOS
Administrador	2
Policía de tránsito	10
Empadronador	3
Total	15

Tabla 4.2 Número de usuarios encuestados

Fuente: Elaboración propia

Para aplicar GQM que consiste en establecer como objetivo de negocio las características de la norma 25010, luego se genera un conjunto de preguntas para satisfacerlos, dichas preguntas son simples de responder si o no, las respuestas son combinadas de forma lógica generando los denominados criterios de evaluación los cuales tienen asociado un puntaje que será identificador al momento de definir las métricas, hacemos un análisis y lo comparamos con los criterios de decisión, donde el puntaje total es la suma de los pts., el puntaje obtenido es la suma de las rptas., el resultado es la división del puntaje obtenido con el puntaje total y por ultimo tenemos la toma de decisión como se observa en la tabla 4.3 (Calabrese y Muños, 2018).

INACEPTABLE	MIN. ACEPTABLE	RANGO OBJETIVO	EXCEDE LOS REQ.
0.00 – 0.20	0.21 – 0.60	0.61 - 0.90	0.91 – 1

Tabla 4.3 Criterios de decisión

Fuente: Elaboración propia

4.4.1 USABILIDAD

Es la capacidad del producto para ser entendido, aprendido, usado y resulta atractivo para el usuario.

- **Accesibilidad:** permite que el producto sea utilizado por usuarios con determinadas

características y discapacidades.

- **Aprendizaje:** permite al usuario aprender su aplicación
- **Estética:** capacidad de la interfaz de usuario agradable
- **Inteligibilidad:** capacidad del producto que permite al usuario si el software es adecuado para sus necesidades
- **Operabilidad:** capacidad del producto que permite al usuario operarlo y controlarlo con facilidad
- **Protección frente a errores de usuario:** capacidad del sistema para proteger al usuario contra errores.

GENERACIÓN DE PREGUNTAS Y RESPUESTAS

En función del objetivo descrito anteriormente, se define un conjunto de preguntas realizadas a los usuarios con el promedio de respuestas obtenidas fácil de responder si o no para satisfacer el mismo como se muestra la tabla 4.4.

ID	PREGUNTA	RESP.
P1	¿El sistema permite cambiar los colores del mismo para adecuarse a las necesidades del usuario?	NO
P2	¿El sistema permite cambiar el tamaño de la letra de sus textos?	SI
P3	¿El sistema está preparado para la lectura de pantalla de voz?	NO
P4	¿El sistema presenta textos difíciles de comprender?	NO
P5	¿El sistema posee textos con información irrelevante?	NO
P6	¿El sistema posee texto y/o palabras con faltas ortográficas?	NO
P7	¿El sistema permite deshacer una acción realizada?	SI
P8	¿El sistema presenta textos escritos en diferentes idiomas?	NO

P9	¿El sistema brinda la opción de cambiar el idioma?	NO
P10	¿El sistema posee una interfaz amigable?	SI
P11	¿El sistema indica la sección en la que se encuentra el usuario?	SI
P12	¿El sistema indica las secciones accedidas hasta el momento?	SI
P13	¿El sistema posee más de un término para referirse a una misma acción?	SI
P14	¿El contenido de los listados del sistema se organiza en páginas?	NO
P15	¿El sistema presenta consistencia de colores en todas las secciones?	SI
P16	¿El sistema posee errores visuales?	NO
P17	¿El sistema informa mediante un mensaje si una operación fue realizada con éxito o error?	SI
P18	¿El sistema permite salir de alguna manera de cada sección?	SI
P19	¿El sistema posee atajos de teclado?	SI
P20	¿El sistema posee iconos para el acceso a las diferentes funcionalidades?	SI
P21	Ante un error ¿El sistema explica cómo solucionarlo?	SI
P22	Ante un error ¿El sistema explica el error?	SI
P23	Ante un error ¿El sistema explica cómo prevenir?	NO
P24	Ante varios errores ¿El mensaje de error se mantiene consistente?	NO
P25	Ante un formulario ¿El sistema indica el tipo de información?	SI
P26	Ante un formulario ¿El sistema indica cuales de sus campos son obligatorios?	SI
P27	Ante un formulario ¿El sistema permite ingresar datos incorrectos?	NO

P28	Ante un formulario ¿Existe información pre cargada?	SI
P29	En cada sección del sistema ¿se brinda una pequeña ayuda?	SI
P30	¿El sistema posee una sección de ayuda? (Manual de usuario)	SI
P31	¿El sistema posee una sección de preguntas frecuentes?	NO
P32	Al utilizar la ayuda ¿Se pudo resolver la inquietud?	SI
P33	¿El sistema posee un acceso rápido de ayuda?	NO

Tabla 4.4 Preguntas y respuestas de usabilidad
Fuente: Elaboración propia

CRITERIOS DE EVALUACIÓN

Las respuestas se combinan de forma lógica, generando los criterios de evaluación que las transformaran en valores cuantitativos como se observa en la tabla 4.5

METRICA	PROPÓSITO	NOMBRE	FORMULA	PTOS.	RPTA.
Accesibilidad	¿Cuán capaz es el sistema de permitir que sea utilizado por usuarios con determinadas características y discapacidades?	Visual	$P3 = SI$	0.5	0.0
			$P3 \& (P2 P1) = SI$	0.75	0.0
			$(P1 P2) \& -(P3) = SI$	0.25	0.25
			$P1 \& P2 \& P3 = SI$	1	0.0
		Atajos	$P19 = SI$	1	1
		Idiomas	$-(P8) \& P9 = SI$	1	0.0
$-(P8) P9 = SI$	0.75		0.75		
Aprendizaje	¿Cuán capaz es el sistema de permitir al usuario aprender la aplicación?	Contexto de usuario	$P11 = SI$	0.75	0.75
			$P12 = SI$	0.5	0.5
			$P11 \& P12 = SI$	1	1
		Ayuda	$P29 P30 P31 = SI$	1	1

		Formularios	P26 = SI	1	1
			P25 & -(P27) & P28 = SI	1	1
			(-P27 P28) = SI	0.5	0.5
			-P27 & P28 = SI	0.75	0.75
			P25 = SI	0.25	0.25
		Manejo de errores	P21 & P22 & P23 = SI	1	0.0
			P21 & P22 = SI	0.75	0.75
			P21 P22 = SI	0.5	0.5
Estética	¿Cuán capaz es el sistema de agradar y satisfacer la interacción con el usuario mediante su interfaz?	Errores visuales	P16 = No	1	1
		Interfaz amigable	P10 & P14 = SI	1	0.0
			P10 = SI	0.75	0.75
			P14 = SI	0.25	0.0
		Consistencia de interfaz	(-P13&P15) P24 = SI	1	0.0
			-P13 P15 =SI	0.5	0.5
			P24 & (P13&-P15) = SI	0.25	0.0
		Colores y formas	P15 & P18 & P33 = SI	1	0.00
			P15 & (P18 P33) =SI	0.75	0.75
			P15 = SI	0.5	0.5
			P18 P33 = SI	0.25	0.25
		Apariencia	-P4 & -P5 & -P6 = SI	1	1

Inteligibilidad	¿Cuán capaz es el sistema de permitir al usuario entender si el software es adecuado para sus necesidades?	Acceso de ayuda	P33 = SI	1	0.0
		Uso de ayuda	P29 P30 P31 & P33 = SI	1	0.0
		Formularios	P25 -P27 P28 = SI	1	1
Operabilidad	¿Cuán capaz es el sistema de permitir al usuario operarlo o controlarlo con facilidad?	Ayuda	P18 & P19 & P29 = SI	1	1
			P29 P30 P31 = SI	0.75	0.75
		Atajos	P18 & P19 & P20 = SI	1	1
			P18 P19 P20 = SI	0.5	0.5
		Interfaz física	(-P13&P15) P24 = SI	1	0.0
			-P13 P15 = SI	0.5	0.5
			P25 & (P13 & -P15) = SI	0.25	0.0
		Avisos	(P11 P12) & P17 = SI	1	1
			P17 = SI	1	1
			P11 P12 = SI	1	1
		Apariencia	-P4 & -P5 & -P6 = SI	1	1
		Interfaz amigable	P10 = SI	1	1
		Deshacer	P7 = SI	1	1
Protección frente a errores	¿Cuán capaz es el sistema de proteger a los	Prevención de reincidencia	P23 = si	1	0.0

	usuarios de hacer errores?	de error			
		Prevención de error (datos)	P25 & -P27 & P28 = si	1	1
			-27 & P28 = si	0.75	0.75
			-27 P28 = si	0.5	0.5
			P25 = SI	0.25	0.25
		Prevención de error (campo)	P26 = SI	1	1
		Manejo de errores	P21 & P22 & P23 = SI	1	0.0
			P21 & P22 = SI	0.75	0.75
			P21 P22 = SI	0.5	0.5

Tabla 4.5 Criterios de evaluación para la usabilidad

Fuente: Elaboración propia

CRITERIOS DE DECISIÓN

Para cada una de las subcaracterísticas, se estableció el siguiente criterio de decisión obteniendo el promedio como se visualiza en la tabla 4.6.

METRICA	PUNTAJE TOTAL	PUNTAJE OBTENIDO	RESULTADO	DECISIÓN
Accesibilidad	5.25	0.38	0.38	Min. Aceptable
Aprendizaje	9	8	0.88	Rango objetivo
Estética	8.25	4.75	0.57	Min. Aceptable
Inteligibilidad	3	1	0.33	Min. Aceptable
Operabilidad	11	9.75	0.88	Rango objetivo

Protección frente a errores	6.75	4.75	0.70	Rango objetivo
Resultado promedio			0.62	Rango objetivo

Tabla 4.6 Criterios de decisión para usabilidad

Fuente: Elaboración propia

4.4.2 SEGURIDAD

Según Aguilera (2011), se puede definir a la seguridad informática como la disciplina encargada de plantear y diseñar las normas, procedimientos, métodos y técnicas con el fin de obtener que un sistema de información sea seguro, confiable y sobre todo que tenga disponibilidad. La principal tarea es de minimizar riesgos, capacidad de protección de la información de manera que personas o sistemas no autorizados no puedan acceder o modificar.

Lo que debe contemplar la seguridad se puede clasificar en:

Los **usuarios**: Son considerados como el eslabón más débil de la cadena, ya que a las personas es difícil de controlar, un usuario puede un día cometer un error y olvidar algo o tener un accidente y este suceso puede echar a perder el trabajo de mucho tiempo, en muchos casos el sistema debe de protegerse del mismo usuario.

La **Información**: Se considera como el oro de la seguridad informática ya que es lo que se desea proteger y lo que tiene que estar a salvo, es el principal activo

La **Infraestructura**: Este puede ser uno de los medios más controlados, pero no implica que corre menos riesgo, se deben considerar aspectos como el acceso no permitido, robo de identidad, robo de equipos, redes, o cualquier desastre natural.

Los pilares de la seguridad de la información son:

- **Autenticidad**: capacidad de demostrar la identidad de un sujeto o recurso.
- **Confidencialidad**: capacidad de protección contra el acceso de datos e información no autorizado. Consiste en asegurar que solo el personal autorizado accede a la información

que le corresponde, para garantizar la confidencialidad se recurre a tres recursos: autenticación de usuario, gestión de privilegios y cifrado de la información (Irene, 2018)

- **Integridad:** capacidad del sistema para prevenir el acceso, consiste en asegurarse de que la información no se pierda ni se ve comprometida voluntaria e involuntariamente, para garantizar la integridad de la información se debe considerar lo siguiente: Monitorear el tráfico de la red, auditar los sistemas, es decir, quien hace, cuando hace, implementar sistemas de control de cambios y hacer back-up copias de seguridad (Irene, 2018)
- **No repudio:** capacidad de demostrar las acciones que han tenido lugar, de manera que dichas acciones no pueden ser repudiados posteriormente
- **Responsabilidad:** capacidad de rastrear de forma inequívoca las acciones de una identidad.

GENERACIÓN DE PREGUNTAS Y RESPUESTAS

En función del objetivo descrito anteriormente, se define un conjunto de preguntas realizadas a los usuarios con el promedio de respuestas obtenidas fácil de responder si o no para satisfacer el mismo como se muestra la tabla 4.7.

ID	PREGUNTA	RESP.
P1	¿Se requiere que la contraseña posea al menos 8 caracteres?	SI
P2	¿Se requiere que la contraseña posea letras mayúsculas y minúsculas?	SI
P3	¿Se requiere que la contraseña posea números y letras?	SI
P4	¿Se requiere que la contraseña posea caracteres especiales?	NO
P5	¿El sistema utiliza conexión segura mediante HTTPS?	SI
P6	¿La base de datos posee los datos encriptados?	SI
P7	¿El sistema permite acceder a funcionalidades en las cuáles no se tiene permiso?	NO
P8	¿El sistema permite que cualquier persona tenga acceso a la base de datos?	NO
P9	¿El sistema permite que cualquier persona tenga acceso al código del	NO

	servidor de la aplicación?	
P10	¿Cualquier persona tiene acceso al servidor físico?	NO
P11	¿Cualquier persona tiene acceso al servidores remoto?	NO
P12	¿El sistema posee re direccionamiento hacia sitios no seguros?	NO
P13	¿el sistema solicita una confirmación de registro mediante un mail a la hora de registrarse?	NO
P14	¿El sistema permite que cualquier persona puede modificar la base de datos?	NO
P15	¿El sistema permite que cualquier persona pueda modificar el código del servidor de la aplicación?	NO
P16	¿El sistema permite inyecciones SQL?	NO
P17	¿El sistema posee un historial de acciones realizadas?	NO
P18	¿El sistema posee algoritmos de cifrado de datos?	SI
P19	¿El sistema posee un mecanismo criptográfico, como firma digital?	NO
P20	¿El sistema solicita confirmación a la hora de realizar una acción?	SI
P21	¿El sistema posee una protección con certificados SSL?	SI
P22	¿El sistema da aviso cuando se es accedido desde una ubicación desconocida?	NO
P23	¿El sistema informa vía mail las operaciones realizadas?	NO
P24	¿El sistema guarda un registro de fecha y hora de ingreso al mismo?	SI
P25	¿El sistema registra el tipo de navegación y sistema de operación utilizado para ingresar al sitio?	NO
P26	¿El sistema registra la dirección IP desde la cual se ingresa al sitio?	NO
P27	¿El sistema realiza una comprobación de identidad mediante un certificado digital?	NO
P28	¿El sistema posee un sistema de verificación en dos pasos?	SI
P29	¿Es requerida una clave de segundo nivel para el ingreso al sistema?	SI
P30	¿El sistema realiza una comprobación de identidad mediante datos	NO

	biométricos?	
P31	¿El sistema realiza una comprobación de identidad mediante tarjeta de coordenadas?	NO
P32	¿El sistema realiza una comprobación de identidad mediante credenciales?	SI
P33	¿El sistema realiza una comprobación de identidad mediante una firma electrónica?	NO

Tabla 4.7 Preguntas y respuestas para seguridad

Fuente: Elaboración propia

CRITERIOS DE EVALUACIÓN

Las respuestas se combinan de forma lógica, generando los criterios de evaluación que las transformaran en valores cuantitativos como se observa en la tabla 4.8.

METRICA	PROPÓSITO	NOMBRE	FORMULA	PTOS.	RPTA.
Confidencialidad	¿Cuán eficiente es el sistema a la hora de proteger el acceso de datos e información no autorizado, ya sea accidental o deliberadamente?	Conexiones seguras	P5 & -P12 = SI	1	1
		Control de acceso	P7 P8 P9 P10 P11 = NO	1	1
		Encriptación de datos	P6 = SI	1	1
		Contraseña	P1 P2 P3 P4 = NO	0.7	0.7
			P1 & P2 & P3 & P4 = SI	1	0.0
			P1 P2 P3	0.5	0.5

			P4 = SI		
Integridad	¿Cuán capaz es el sistema a la hora de prevenir accesos o modificaciones no autorizados a datos o programas del ordenador?	Prevención de acceso	P7 P8 P9 P16 = NO	1	1
		Prevención de modificaciones	P14 P15 = NO	1	1
		Confirmación de datos	P13 = SI	1	0.0
No-Repudio	¿Cuán capaz es el sistema de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente?	Operaciones realizadas	P17 p23 = si	1	0.0
		Mecanismo de cifrado	P18 P19 P21 = SI	1	1
		Confirmación de acciones	P20 = SI	1	1
		Registro de ubicación	P22 = SI	1	0.0
Autenticidad	¿Cuán capaz es el sistema de demostrar la	Comprobación de identidad	P27 P30 P31 P32 P33 = SI	1	1

	identidad de un sujeto o un recurso?	Comprobaciones adicionales	P28 P29 P13 = SI	1	1
Responsabilidad	¿Cuán capaz es el sistema de rastrear de forma inequívoca las acciones de una entidad?	Registro de acciones y datos	P17 P24 P25 P26 = SI	1	1
		Control de ubicación	P22 = SI	1	0.0

Tabla 4.8 Criterios de evaluación para la seguridad

Fuente: Elaboración propia

CRITERIOS DE DECISIÓN

Para cada una de las subcaracterísticas, se estableció el siguiente criterio de decisión obteniendo el promedio como se visualiza en la tabla 4.9

METRICA	PUNTAJE TOTAL	PUNTAJE OBTENIDO	RESULTADO	DECISIÓN
Confidencialidad	5.2	4.2	0.81	Rango objetivo
Integridad	3	2	0.66	Rango objetivo
No-Repudio	4	2	0.5	Min. Aceptable
Autenticidad	2	2	1	Excede los req.
Responsabilidad	2	1	0.5	Min. Aceptable
Resultado promedio			0.69	Rango objetivo

Tabla 4.9 Criterios de decisión para la seguridad

Fuente: Elaboración propia

4.4.3 PORTABILIDAD

Capacidad del producto de ser transferido de forma efectiva y eficiente.

- **Adaptabilidad:** capacidad del producto que le permite ser adaptado a diferentes entornos determinados de hardware, software y operaciones de uso
- **Capacidad de ser remplazado:** capacidad del producto para ser utilizado en lugar de otro producto, con el mismo propósito y el mismo entorno
- **Facilidad de instalación:** facilidad con la que el producto se puede instalar o desinstalar de forma exitosa en un determinado entorno.

GENERACIÓN DE PREGUNTAS Y RESPUESTAS

En función del objetivo descrito anteriormente, se define un conjunto de preguntas realizadas a los usuarios con el promedio de respuestas obtenidas fácil de responder si o no para satisfacer el mismo como se muestra la tabla 4.10.

ID	PREGUNTA	RESP.
P1	¿El sistema posee una aplicación móvil?	NO
P2	¿El sistema funciona correctamente en diferentes computadoras con diferentes características?	SI
P3	¿El sistema se utiliza correctamente en un navegador de una Tablet?	SI
P4	¿El sistema puede ser utilizado en dispositivos con cualquier sistema operativo?	SI
P5	¿El sistema funciona correctamente en cualquier navegador de internet?	SI
P6	¿El sistema funciona correctamente en el navegador de un dispositivo móvil?	SI
P7	¿El sistema puede ser reemplazado con otro sistema de las mismas características?	NO

P8	¿El sistema puede ser utilizado en otro servidor?	SI
----	---	----

Tabla 4.10 Generación de preguntas y respuestas para portabilidad
Fuente: Elaboración propia

CRITERIOS DE EVALUACIÓN

Las respuestas se combinan de forma lógica, generando los criterios de evaluación que las transformaran en valores cuantitativos como se observa en la tabla 4.11.

METRICA	PROPÓSITO	NOMBRE	FORMULA	PTOS.	RPTA.
Adaptabilidad	¿Cuán adaptable es el producto a diferentes entornos de hardware, software?	Web responsivo	P3 & P6 = SI	1	1
		Aplicación móvil	P1 = SI	1	0.0
		Hardware	P2 = SI	1	1
		Software	P4 & P5 = SI	1	1
Capacidad de ser reemplazado	¿Cuán capaz es el producto de ser reemplazado?	Reemplazar	P7 = SI	1	0.0
		Reemplazado	P8 = SI	0.7	0.7
Facilidad de instalación	¿Cuán capaz es el sistema de ser instalado o ejecutado?	Navegador web	P2 & P3 & P4 = SI	1	1

Tabla 4.11 Criterios de evaluación para portabilidad
Fuente: Elaboración propia

CRITERIOS DE DECISIÓN

Para cada una de las subcaracterísticas, se estableció el siguiente criterio de decisión obteniendo el promedio como se visualiza en la tabla 4.12.

METRICA	PUNTAJE TOTAL	PUNTAJE OBTENIDO	RESULTADO	DECISIÓN
Adaptabilidad	4	2	0.5	Min. Aceptable
Capacidad de ser reemplazado	1.7	0.7	0.41	Min. Aceptable
Facilidad de instalación	1	1	1	Excede los req.
Resultado promedio			0.64	Rango objetivo

Tabla 4.12 Criterios de decisión para portabilidad

Fuente: Elaboración propia

4.4.4 COMPATIBILIDAD

Capacidad de dos o más sistemas para intercambiar información y llevar a cabo sus funciones requeridas cuando comparten el mismo entorno de hardware o software.

- **Coexistencia:** capacidad del producto para coexistir con otro software independiente.
- **Interoperabilidad:** capacidad de dos o más sistemas para intercambiar información .

GENERACIÓN DE PREGUNTAS Y RESPUESTAS

En función del objetivo descrito anteriormente, se define un conjunto de preguntas realizadas a los usuarios con el promedio de respuestas obtenidas fácil de responder si o no para satisfacer el mismo como se muestra la tabla 4.13

ID	PREGUNTA	RESP.
P1	¿Es posible utilizar los recursos de la computadora por otros sistemas mientras se utiliza el sistema?	SI
P2	¿Es posible ejecutar el sistema si se están ejecutando otros?	SI

P3	¿Se producen errores inesperados al ejecutar el sistema cuando hay otros ejecutándose?	No
P4	¿El sistema permite intercambiar información con otros sistemas?	NO
P5	¿Es posible utilizar información brindada por otro sistema?	NO
P6	¿La información del sistema puede ser utilizada por otro sistema?	SI
P7	¿Se producen errores de algún tipo al intentar utilizar información compartida con otros sistemas?	SI
P8	¿La funcionalidad del sistema se ve alterada por el uso de otro sistema al mismo tiempo?	NO

Tabla 4.13 Generación de preguntas y respuestas para la compatibilidad

Fuente: Elaboración propia

CRITERIOS DE EVALUACIÓN

Las respuestas se combinan de forma lógica, generando los criterios de evaluación que las transformaran en valores cuantitativos como se observa en la tabla 4.14.

METRICA	PROPÓSITO	NOMBRE	FORMULA	PTOS.	RPTA.
Coexistencia	¿Cuán capaz es el producto de coexistir con otro software, en un entorno común?	Coexistencia entre sistemas	P2 = SI	1	1
			P2 & -P8 = SI	1	1
			P2 & P8 = SI	0.25	0.0
		Utilización de recursos	P1 = SI	1	1
		Errores inesperados	P3 = SI	1	0.0
Interoperabilidad	¿Cuán capaz son	Intercambio de	P4 = NO	1	1

	dos o más sistemas de intercambiar información y utilizarla?	información	P4 & P5 & P6 = SI	1	0
			(P4&P5) (P4&P6) = SI	0.75	0
	Errores inesperados	P7 = NO	1	0	

Tabla 4.14 Criterios de evaluación para la compatibilidad

Fuente: Elaboración propia

CRITERIOS DE DECISIÓN

Para cada una de las subcaracterísticas, se estableció el siguiente criterio de decisión obteniendo el promedio como se visualiza en la tabla 4.15.

METRICA	PUNTAJE TOTAL	PUNTAJE OBTENIDO	RESULTADO	DECISIÓN
Coexistencia	4.25	3	0.70	Rango objetivo
Interoperabilidad	3.75	1	0.26	Min. Aceptable
Resultado promedio			0.48	Min. Aceptable

Tabla 4.15 Criterios de decisión para la compatibilidad

Fuente: Elaboración propia

4.4.5 MANTENIBILIDAD

Capacidad del producto para ser modificado eficientemente.

- **Analizabilidad:** facilidad con la que se puede evaluar el impacto de un determinado cambio al producto
- **Capacidad de ser modificado:** permite que el producto sea modificado de forma efectiva

- **Modularidad:** capacidad de un sistema que permite que un cambio tenga un impacto mínimo en los demás
- **Reusabilidad:** capacidad de un activo que permite que sea utilizado en más de un sistema

GENERACIÓN DE PREGUNTAS Y RESPUESTAS

En función del objetivo descrito anteriormente, se define un conjunto de preguntas realizadas a los usuarios con el promedio de respuestas obtenidas fácil de responder si o no para satisfacer el mismo como se muestra la tabla 4.16

ID	PREGUNTA	RESP.
P1	¿Es entendible los componentes del sistema?	SI
P2	En caso de cambios ¿Hay un cambio brusco?	NO
P3	En caso de cambios ¿El sistema presenta problemas?	NO
P4	En caso de cambios ¿El sistema afecta a otros sistemas?	SI
P5	¿Un componente es reutilizado?	SI

Tabla 4.16 Generación de preguntas y respuestas para la mantenibilidad

Fuente: Elaboración propia

CRITERIOS DE EVALUACIÓN

Las respuestas se combinan de forma lógica, generando los criterios de evaluación que las transformaran en valores cuantitativos como se observa en la tabla 4.17.

MÉTRICA	PROPÓSITO	NOMBRE	FORMULA	PTOS.	RPTA.
Analizabilidad	¿Cuán capaz es de evaluar el impacto del cambio del	Componentes del sistema	$P1 = SI$	1	1

	producto?				
Capacidad de ser modificado	¿Cuán capaz es el producto de ser modificado?	interfaz	P2 = NO	0.75	0.75
		Errores	P2 P3 = NO	0.5	0.5
Reusabilidad	¿Cuán capaz es el sistema de que un activo sea reutilizado?	Componentes del sistema	P1 & P5 = SI	0.75	0.75

Tabla 4.17 Criterios de evaluación para la mantenibilidad

Fuente: Elaboración propia

CRITERIOS DE DECISIÓN

Para cada una de las subcaracterísticas, se estableció el siguiente criterio de decisión obteniendo el promedio como se visualiza en la tabla 4.18

METRICA	PUNTAJE TOTAL	PUNTAJE OBTENIDO	RESULTADO	DECISIÓN
Analizabilidad	4.25	3	0.70	Rango objetivo
Capacidad de ser modificado	3.75	1	0.26	Min. Aceptable
Reusabilidad	0.75	0.75	1	Excede los req.
Resultado promedio			0.65	Rango objetivo

Tabla 4.18 Criterios de decisión para la mantenibilidad

Fuente: Elaboración propia

4.4.6 FIABILIDAD

Capacidad de un sistema para desempeñar las funciones especificadas.

- **Capacidad de recuperación:** capacidad del producto para recuperar los datos y

restablecer el estado del sistema.

- **Disponibilidad:** capacidad del sistema para estar disponible cuando se requiera, la información para que resulte útil y valiosa debe estar disponible para quien la necesite, se debe implementar las medidas necesarias para que tanto la información y los servicios estén disponibles (Irene, 2018).
- **Madures:** capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
- **Tolerancia a fallos:** capacidad del sistema para operar según lo previsto en presencia de fallos de hardware o software.

GENERACIÓN DE PREGUNTAS Y RESPUESTAS

En función del objetivo descrito anteriormente, se define un conjunto de preguntas realizadas a los usuarios con el promedio de respuestas obtenidas fácil de responder si o no para satisfacer el mismo como se muestra la tabla 4.19.

ID	PREGUNTA	RESP.
P1	¿El sistema tiene la facilidad de realizar un backup?	SI
P2	¿El sistema tiene la facilidad de recuperar el historial?	NO
P3	¿El sistema realiza copias de seguridad?	SI
P4	¿El sistema presenta un respaldo de los recursos?	NO
P5	¿El tiempo de carga es rápido?	SI
P6	¿El sistema está disponible en todo momento?	SI
P7	¿El sistema gestiona el historial?	NO
P8	¿El sistema realiza la detección y corrección de fallas?	NO
P9	¿El sistema genera resultados duplicados?	NO

P10	¿El sistema visualiza los errores por pantalla?	SI
-----	---	----

Tabla 4.19 Generación de preguntas y respuestas para la fiabilidad

Fuente: Elaboración propia

CRITERIOS DE EVALUACIÓN

Las respuestas se combinan de forma lógica, generando los criterios de evaluación que las transformaran en valores cuantitativos como se observa en la tabla 4.20.

METRICA	PROPÓSITO	NOMBRE	FORMULA	PTOS.	R.
Capacidad de recuperación	¿Cuán capaz es el sistema para recuperar datos?	Recuperación del sistema	P1 & P2 = SI	1	0
			P1 P2 = SI	0.7	0.7
		Recuperabilidad de datos	P3 P4 = SI	1	1
Disponibilidad	¿Cuán capaz es el sistema para estar disponible cuando se requiera?	Acceso al sistema	P6 = SI	1	1
		Acceso a la funcionalidad	P5 & P6	0.7	0.7
Madurez	¿Cuán capaz es el sistema para satisfacer las necesidades de fiabilidad en condiciones normales?	Historial	P7 & P8 = SI	1	0
		Robustez	P9 = NO	1	1
Tolerancia a fallos	¿Cuán capaz es el sistema para operar ante fallos?	Fallos	P10 = si	1	1

Tabla 4.20 Criterios de evaluación para la fiabilidad

Fuente: Elaboración propia

CRITERIOS DE DECISIÓN

Para cada una de las subcaracterísticas, se estableció el siguiente criterio de decisión obteniendo el promedio como se visualiza en la tabla 4.21.

METRICA	PUNTAJE TOTAL	PUNTAJE OBTENIDO	RESULTADO	DECISIÓN
Capacidad de recuperación	2.7	1.7	0.58	Min. Aceptable
Disponibilidad	1.7	1.7	1	Excede los req.
Madurez	2	1	0.5	Min. Aceptable
Tolerancia a fallos	1	1	1	Excede los req.
Resultado promedio			0.77	Rango objetivo

Tabla 4.21 Criterios de decisión para la fiabilidad

Fuente: Elaboración propia

4.4.7 EFICIENCIA DE DESEMPEÑO

Representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones.

- **Capacidad:** grado en que los límites máximos de un parámetro de un producto cumplen con los requisitos
- **Comportamiento temporal:** se refiere a los tiempos de respuesta y procesamiento del sistema
- **Utilización de recursos:** las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función

GENERACIÓN DE PREGUNTAS Y RESPUESTAS

En función del objetivo descrito anteriormente, se define un conjunto de preguntas realizadas a los usuarios con el promedio de respuestas obtenidas fácil de responder si o no para

satisfacer el mismo como se muestra la tabla 4.22.

ID	PREGUNTA	RESP.
P1	¿El sistema cumple con los requerimientos?	SI
P2	¿El sistema presenta requerimientos no funcionales?	SI
P3	¿El tiempo de ejecución es demasiado?	NO
P4	¿El tiempo de carga sobre pasa lo esperado?	NO
P5	¿El sistema presenta el rendimiento requerido?	SI
P6	¿El sistema consume muchos recursos de la red?	SI
P7	¿El sistema rinde eficientemente en todos los equipos?	NO
P8	¿El sistema requiere de una buena computadora?	SI

Tabla 4.22 Generación de preguntas y respuestas para la eficiencia de desempeño

Fuente: Elaboración propia

CRITERIOS DE EVALUACIÓN

Las respuestas se combinan de forma lógica, generando los criterios de evaluación que las transformaran en valores cuantitativos como se observa en la tabla 4.23.

METRICA	PROPÓSITO	NOMBRE	FORMULA	PTOS.	RPTA.
Capacidad	¿Cuán capaz es el sistema para cumplir los requisitos?	Requisitos funcionales	P1 = SI	1	1
			P2 = NO	0.5	0
		Requisitos no funcionales	P2 = SI	0.25	0.25
Comportamiento	¿Cuán capaz es el sistema para	Tiempo de respuesta y	P3 P4 = NO	0.5	0.5

temporal	los tiempos de respuesta?	espera			
		Rendimiento	P5 = SI	1	1
Utilización de recursos	¿Cuán capaz es el sistema para utilizar los recursos?	Recursos de hardware y software	P6 & P7 & P8 = SI	1	0
			P6 P7 P8 = SI	0.7	0.7
			P8 & P6 = SI	0.25	0.25

Tabla 4.23 Criterios de evaluación para la eficiencia de desempeño

Fuente: Elaboración propia

CRITERIOS DE DECISIÓN

Para cada una de las subcaracterísticas, se estableció el siguiente criterio de decisión obteniendo el promedio como se visualiza en la tabla 4.24.

METRICA	PUNTAJE TOTAL	PUNTAJE OBTENIDO	RESULTADO	DECISIÓN
Capacidad	1.75	1.25	0.71	Rango objetivo
Comportamiento temporal	1.5	1.5	1	Excede los req.
Utilización de recursos	1.95	0.95	0.48	Min. Aceptable
Resultado promedio			0.73	Rango objetivo

Tabla 4.24 Criterios de decisión para la eficiencia de desempeño

Fuente: Elaboración propia

4.4.8 ADECUACIÓN FUNCIONAL

Representa la capacidad del producto para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuando el producto se utiliza en las condiciones

especificadas.

- **Complejidad funcional:** cubre todas las tareas y objetivos del usuario
- **Corrección funcional:** capacidad del producto para proveer resultados correctos
- **Pertinencia funcional:** capacidad del producto para proporcionar funciones a las tareas de los usuarios

GENERACIÓN DE PREGUNTAS Y RESPUESTAS

En función del objetivo descrito anteriormente, se define un conjunto de preguntas realizadas a los usuarios con el promedio de respuestas obtenidas fácil de responder si o no para satisfacer el mismo como se muestra la tabla 4.25.

ID	PREGUNTA	RESP.
P1	¿El sistema cumple con todos los objetivos planteados?	SI
P2	¿El sistema cumple con todas las tareas asignadas?	NO
P3	¿Se ha proporcionado los materiales de apoyo?	SI
P4	¿Se ha cumplido con las actividades, reuniones colaborativas?	NO
P5	¿El sistema retorna los resultados esperados?	SI

Tabla 4.25 Generación de preguntas y respuestas para la adecuación funcional
Fuente: Elaboración propia

CRITERIOS DE EVALUACIÓN

Las respuestas se combinan de forma lógica, generando los criterios de evaluación que las transformaran en valores cuantitativos como se observa en la tabla 4.26.

METRICA	PROPÓSITO	NOMBRE	FORMULA	PTOS.	RPTA.
Complejidad	¿El sistema	Objetivos y	$P1 \& P2 = SI$	1	0

funcional	cumple con todas las tareas?	tareas	P1 P2 = SI	0.7	0.7
		Materiales y actividades	P3 & P4 = SI	0.5	0
			P3 P4 = SI	0.25	0.25
Corrección funcional	¿El sistema retorna los resultados esperados?	Resultados y errores	P2 & P5 = SI	1	1
			P4 & P5 = SI	0.7	0

Tabla 4.26 Criterios de evaluación para la adecuación funcional

Fuente: Elaboración propia

CRITERIOS DE DECISIÓN

Para cada una de las subcaracterísticas, se estableció el siguiente criterio de decisión obteniendo el promedio como se visualiza en la tabla 4.27.

METRICA	PUNTAJE TOTAL	PUNTAJE OBTENIDO	RESULTADO	DECISIÓN
Compleitud funcional	1.55	0.95	0.62	Rango objetivo
Corrección funcional	1.7	1	0.58	Min. Aceptable
Resultado promedio			0.60	Min. Aceptable

Tabla 4.27 Criterios de decisión para la adecuación funcional

Fuente: Elaboración propia

5. CAPITULO V ESTIMACIÓN DE COSTO Y BENEFICIO

5.1 INTRODUCCIÓN

Una de las tareas de mayor importancia en el desarrollo de software es la estimación de costos. Se implementa en la etapa de planeamiento que permite decidir cuantas personas son necesarias para llevar a cabo el proyecto y establecer el cronograma adecuado. Para controlar el progreso del proyecto, es decir, evaluar si el proyecto está evolucionando de acuerdo al cronograma, para esto se requiere contar con métricas que permite medir el nivel de cumplimiento del desarrollo.

La estimación de costo beneficio tiene como objetivo fundamental proporcionar una medida sobre la rentabilidad de un proyecto, haciendo una comparación de los costos previstos con los beneficios esperados en la realización del mismo (Boehm et. al, 1995).

5.2 COCOMO II

Según Boehm (1995), Modelo Constructivo de Costos, este modelo realiza estimaciones en función al tamaño del software, costo y escala. Los objetivos principales que se tuvieron en cuenta para construir el modelo fueron:

- Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de la década de los 90 como a las futuras
- Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación
- Implementar una herramienta de software que soportara el modelo
- Proveer una marca analítico cuantitativo y un conjunto de herramientas y técnicas que evaluaran el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo

COCOMO II Está compuesto por tres modelos que se adaptan a las necesidades de los diferentes sectores descriptivos, como el tipo y cantidad de información disponible en cada etapa del ciclo de vida del desarrollo denominados:

- **Composición de Aplicación:** se emplea en el desarrollo de software durante la etapa de prototipación
- **Diseño Temprano:** se utiliza en las primeras etapas del desarrollo en las cuales se evalúan las alternativas de hardware y software de un proyecto. En esta etapa se tiene poca información lo que concuerda el uso de punto función para estimar tamaño y uso de un número reducido de factores de costo
- **Post Arquitectura:** se aplica en la etapa de desarrollo propiamente dicho, después que se define la arquitectura del sistema y en la etapa de mantenimiento. Este modelo utiliza punto función o líneas de código fuente para estimar el tamaño.

5.2.1 ESTIMACIÓN DEL ESFUERZO

El esfuerzo necesario para concretar un proyecto de desarrollo de software, cualquiera sea el modelo empleado, se expresa en meses persona (PM) y representa los meses de trabajo de una persona requeridos para desarrollar el proyecto

Composición de aplicación:

$$PM = NOP/PROD$$

$$NOP = OP * (100 - \%reuso)/100$$

Donde:

NOP (Nuevos Puntos Objeto): es el tamaño del nuevo software a desarrollar expresado en puntos objeto

OP (Puntos Objeto): tamaño del software a desarrollar expresado en puntos objeto

%reúso: porcentaje de reúso que se espera lograr en el proyecto

PROD: es la productividad promedio determinada a partir del análisis de datos de proyectos, como se puede observar en la tabla 5.1

Experiencia y capacidad de los desarrolladores	Muy Bajo	Bajo	Normal	Alto	Muy Alto
Madurez y Capacidad del ICASE	Muy Bajo	Bajo	Normal	Alto	Muy Alto
PROD	4	7	13	25	50

Tabla 5.1 Productividad para el modelo Composición de aplicación
Fuente: (Boehm, 1995)

5.2.2 ESTIMACIÓN DEL CRONOGRAMA

Nos presenta la siguiente ecuación:

$$TDEV = \left[3.0 * PM_*^{(0.33+0.2*(B-1.01))} \right] * \frac{SCED\%}{100}$$

Donde:

TDEV: es el tiempo calendario en meses que transcurre desde la determinación de los requerimientos a la culminación de una actividad que certifique que el producto cumple con las especificaciones

PM_{*}: es el esfuerzo expresado en meses persona, calculado sin tener en cuenta el multiplicador de esfuerzo SCED

SCED%: es el porcentaje de comprensión/expansión del cronograma (Piattini, 2003)

5.2.3 MÉTRICAS DEL SOFTWARE

En la estimación del tamaño del software COCOMO II utiliza tres técnicas:

- a) Puntos Objeto:
 - Determinar la cantidad de objetos: estimar la cantidad de pantallas, reportes y componentes que contempla la aplicación
 - Clasificar cada instancia de un objeto según sus niveles de complejidad (simple, media o difícil) de acuerdo a la tabla 5.2

Para Pantallas			
Cantidad de Vistas Contenidas	Cantidad y fuente de las tablas de datos		
	Total < 4 (< 2 servidor < 3 cliente)	Total < 8 (< 2 - 3 servidor < 3 - 5 cliente)	Total 8 + (> 3 servidor < 5 cliente)
< 3	Simple	Simple	Media
3 - 7	Simple	Media	Difícil
> 8	Media	Difícil	Difícil
Para Reportes			
Cantidad de Vistas Contenidas	Cantidad y fuente de las tablas de datos		
	Total < 4 (< 2 servidor < 3 cliente)	Total < 8 (< 2 - 3 servidor < 3-5 cliente)	Total 8 + (> 3 servidor < 5 cliente)
0 o 1	Simple	Simple	Media
2 o 3	Simple	Media	Difícil
4 +	Media	Difícil	Difícil

Tabla 5.2 Esquema de clasificación de puntos objetos
Fuente: (Boehm, 1995)

- Dar el peso a cada objeto según el nivel de complejidad. Los pesos reflejan el esfuerzo relativo requerido para implementar una instancia de ese nivel de complejidad

Tipo de Objeto	Complejidad - Peso		
	Simple	Media	Difícil
Pantalla	1	2	3
Reporte	2	5	8
Componente 3GL			10

Tabla 5.3 Peso de un punto objeto
Fuente: (Boehm, 1995)

- Determinar la cantidad de puntos objeto, sumando todos los pesos de las instancias de los tipos de objeto especificados
- b) Puntos Función: Procuran cuantificar la funcionalidad de un sistema de software. La meta es obtener un número que caracterice completamente al sistema

$$FP = UFP * TCF$$

Donde:

FP: Punto Función

UFP: Puntos Función no Ajustados

- Entradas Externas (Inputs): Entrada de datos del usuario que ingresa desde el exterior al sistema para agregar y/o cambiar datos de un archivo lógico interno
- Salidas Externas (Outputs): Salida de datos del usuario o de control que deja el límite del sistema de software
- Archivos Lógicos Internos (Archivos): Incluye cada archivo lógico, es decir cada grupo lógico de datos que es generado, usado, o mantenido por el sistema de software
- Archivos Externos de Interfaz (Interfaces): Archivos Transferidos o compartidos entre sistemas de software
- Solicitudes Externas (*Queries*): Combinación única de entrada y salida, donde una entrada causa y genera una salida inmediata, como un tipo de solicitud externa

Una vez identificado los ítems se clasifican de acuerdo al grado de complejidad en: bajo, promedio o alto. Se asigna un peso a cada ítem según el tipo y grado de complejidad correspondientes. Finalmente, los UFP se calcula mediante la sumatoria de los pesos de todos los ítems identificados.

$$UFP = \sum_{i=1}^{15} (\text{cantidad_items_tipo}_i) * (\text{peso}_i)$$

La tabla 5.4 muestra cómo se determina los niveles de complejidad de cada ítem en función del número y tipos de elementos de datos y archivos involucrados

Para archivos lógicos internos y archivos externos de interfase				Para salidas y consultas externas				Para entradas externas			
Elementos de Registro	Elementos de datos			Tipos de archivos	Elementos de datos			Tipos de archivos	Elementos de datos		
	1-19	20-50	51+		1-5	6-19	20+		1-4	5-15	16+
1	Bajo	Bajo	Prom.	0 ó 1	Bajo	Bajo	Prom.	0 ó 1	Bajo	Bajo	Prom.
2-5	Bajo	Prom.	Alto	2-3	Bajo	Prom.	Alto	2-3	Bajo	Prom.	Alto
6+	Prom.	Alto	Alto	4+	Prom.	Alto	Alto	3+	Prom.	Alto	Alto

Tabla 5.4 Determinación del peso

Fuente: (Boehm, 1995)

La tabla 5.5 muestra las ponderaciones asociadas a cada tipo de ítem. Estas ponderaciones han sido derivadas y validadas empíricamente mediante la observación de una gran variedad de proyectos

Tipo de función	Peso del Factor de Complejidad		
	Bajo	Promedio	Alto
Entradas Externas (Inputs)	3	4	6
Salidas Externas (Outputs)	4	5	7
Archivo Lógicos Internos (Archivos)	7	10	15
Archivos Externos de Interfase (Interfases)	5	7	10
Consultas Externas (Queries)	3	4	6

Tabla 5.5 Peso del factor de complejidad

Fuente: (Boehm, 1995)

TCF: Factor de Complejidad Técnica

$$TCF = 0.65 + 0.01 * \sum_{i=1}^{14} F_i$$

Donde: F_i corresponde a los pesos asignados a los siguientes factores

F1: Mecanismos de recuperación y back-up confiables

F2: Comunicación de datos

F3: Funciones de Procesamiento Distribuido

F4: Performance

- F5: Configuración usada rigurosamente
- F6: Entrada de datos on-line
- F7: Factibilidad operativa
- F8: Actualización de archivos on-line
- F9: Interfaces complejas
- F10: Procesamiento interno complejo
- F11: Reusabilidad
- F12: Fácil de instalación
- F13: Soporte de múltiples instalaciones
- F14: Facilidad de cambios y amigabilidad

- **Líneas de Código Fuente:** Para determinar el esfuerzo nominal, los puntos función no ajustados UFP tiene que ser convertidos a líneas de código fuente considerando el lenguaje de implementación (assembler, lenguaje de alto nivel, lenguaje de cuarta generación, etc.) esto se realiza para los modelos Diseño temprano y Post arquitectura

Lenguaje	SLOC/ Pto. Función
Ada	71
AI Shell	49
APL	32
Assembler	320
Assembler (macro).	213
ANSI/Quick/Turbo Basic.	64
Basic - Compilado	91
Basic – Interpretado	128
C	128
C++	29
ANSI Cobol 85	91
Fortran 77	105
Forth	64
Jovial	105
Lisp	64
Modula 2	80
Pascal	91
Prolog	64
Generador de Reportes	80
Planilla de Cálculo	6

Tabla 5.6 Conversión de UFP a SLOC

Fuente: (Boehm, 1995)

5.3 APLICACIÓN DE COCOMO II

5.3.1 ESTIMACIÓN DEL ESFUERZO

Se aplicó las siguientes fórmulas para determinar el esfuerzo y el cronograma:

$$PM = A * (KSLOC)^B \quad ; \quad TDEV = C * (PM)^D$$

Donde:

PM: es el esfuerzo estimado, representa los meses–persona necesarios para ejecutar el proyecto

TDEV: representa los meses de trabajo que se necesitan para ejecutar el proyecto

KSLOC: es el tamaño del software a desarrollar en miles de líneas de código

A, B, C, D: son coeficientes que varían según el modo de desarrollo (orgánico, semiacoplado, empotrado) como se muestra en la tabla 5.7

Para el presente proyecto se utilizará el modo de desarrollo **semiacoplado**, por tener un tamaño y complejidad media, nivel intermedio de experiencia y conocimiento del sistema en desarrollo.

Modo de Desarrollo	Esfuerzo	Cronograma
Orgánico	$PM=2.4 \times (KSLOC)^{1.05}$	$TDEV=2.5 \times (PM)^{0.38}$
Semiacoplado	$PM=3.0 \times (KSLOC)^{1.12}$	$TDEV=2.5 \times (PM)^{0.35}$
Empotrado	$PM=3.6 \times (KSLOC)^{1.20}$	$TDEV=2.5 \times (PM)^{0.32}$

Tabla 5.7 Ecuaciones del modelo básico de COCOMO

Fuente: (Boehm, 1995)

Para hallar el valor de *KSLOC* resolvemos la siguiente formula:

$$KSLOC = \frac{PF * FactorLCD}{1000}$$

Donde:

PF: es el punto función

FactorLCD: es el lenguaje de programación que se está implementando, lo podemos obtener de la tabla 5.8, tomando en cuenta que se ha implementado en el lenguaje Javascript.

Lenguaje	FactorLCD
JAVASCRIPT	47
VISUAL	35
ASP	36
JAVA	53
ANSI BASIC	64
C	128

Tabla 5.8 FactorLCD

Fuente: (Gomes, Migani, Lopez, & Otazu, 2021)

Para hallar el valor de (PF) punto función resolvemos la siguiente formula:

$$PF = UFP * TCF$$

Donde:

UFP: Es el factor de ponderación

TCF: es el punto función no ajustado

Para obtener el valor de *UFP* aplicando la formula, multiplicando el número de ítems identificado con el peso del factor de complejidad promedio como se observa en la tabla 5.9

$$UFP = \sum_{i=1}^{15} (\text{cantidad_items_tipo}_i) * (\text{peso}_i)$$

Tipo de función	ítem	Peso del factor de complejidad			Total
		Bajo	Promedio	Alto	
Entradas externas (input)	10	3	4	6	40
Salidas externas (output)	8	4	5	7	40
Archivo lógico interno	23	7	10	15	230
Archivo externo de interfaz	12	5	7	10	84
Consultas externas	4	3	4	6	16
UFP					410

Tabla 5.9 UFP

Fuente: Elaboración propia

Para obtener el valor de TCF realizamos la sumatoria de los valores del factor de complejidad dando un valor en el rango del 1 al 5, donde 1: incidental, 2: moderado, 3: medio, 4: significativo y 5: esencial como se observa en la tabla 5.10

$$TCF = 0.65 + 0.01 * \sum_{i=1}^{14} F_i$$

#	Factor de complejidad	Ponderación
1	Mecanismos de recuperación y back-up confiables	3
2	Comunicación de datos	4
3	Funciones de Procesamiento Distribuido	3
4	Performance	3
5	Configuración usada rigurosamente	4
6	Entrada de datos on-line	4
7	Factibilidad operativa	4
8	Actualización de archivos on-line	3
9	Interfaces complejas	3
10	Procesamiento interno complejo	5
11	Reusabilidad	5
12	Fácil de instalación	5
13	Soporte de múltiples instalaciones	4
14	Facilidad de cambios y amigabilidad	4
Sumatoria de F_i		54

Tabla 5.10 Factor de complejidad

Fuente Elaboración propia

Aplicando la formula obtenemos el valor de TCF

$$TCF = 0.65 + 0.01 * 54$$

$$TCF = 1.19$$

Ahora hallamos el valor de (PF) punto función:

$$PF = UFP * TCF$$

$$PF = 410 * 1.19$$

$$PF = 487.9$$

Ahora hallamos el valor de KSLOC:

$$KSLOC = \frac{PF * FactorLCD}{1000}$$

$$KSLOC = \frac{487.9 * 47}{1000}$$

$$KSLOC = 22.931$$

Una vez obtenido el tamaño del software a desarrollar en líneas de código *KSLOC*, finalmente procedemos a hallar el esfuerzo *PM* estimado que representa los meses–persona necesarios para ejecutar el proyecto.

$$PM = A * (KSLOC)^B$$

$$PM = 3 * (22.931)^{1.12}$$

$$PM = 100.183$$

Ahora hallamos *TDEV* que representa el esfuerzo de los meses de trabajo que se necesitan para ejecutar el proyecto

$$TDEV = C * (PM)^D$$

$$TDEV = 2.5 * (100.183)^{0.35}$$

$$TDEV = 12.53$$

Finalmente hallamos el número de programadores que se necesita para desarrollar el sistema, haciendo la siguiente división:

$$\text{Nro. programadores} = \frac{100.183}{12.53}$$

$$\text{Nro. programadores} = \frac{100.183}{12.53} = 7.99 \cong 8$$

Ahora hallamos el tiempo de desarrollo T expresado en meses:

$$T = C * (\text{Nro. programadores})^D$$

$$T = 2.5 * (8)^{0.35}$$

$$T = 5.17 \cong 5 \text{ (meses)}$$

5.3.2 COSTO DE SOFTWARE

Para el costo de software solo tomamos en cuenta el costo de desarrollo. El salario aproximado de un programador es de 500\$, así, obtenemos el costo de desarrollo:

$$\text{Costo de desarrollo} = \text{Num. programadores} * \text{salario} * \text{duracion}$$

$$\text{Costo de desarrollo} = 8 * 500 * 5$$

$$\text{Costo de desarrollo} = 20.000 \$$$

$$\text{Costo del Software} = 20.000 \$$$

Descripción	Costo(\$us)
Costo de elaboración del proyecto	1500
Costo del software desarrollado	20000
Costo de implementación	500
Total	22000

Tabla 5.11 Costo total del proyecto

Fuente: Elaboración propia

Por lo tanto, el total del proyecto tomando los puntos considerados es de 22.000 \$.

5.4 INDICADORES

Se realizará el cálculo de beneficio utilizando el valor neto actual y la tasa interna de retorno

5.4.1 VALOR ACTUAL NETO – VAN

El valor actual neto además de ser muy importante es un procedimiento que permite calcular el valor presente de un determinado número de flujos de futuros ingresos y egresos que tendrá el proyecto, se utiliza para estimar si cierta compra o inversión traerá más beneficios a largo plazo, la fórmula que permite calcular el valor neto actual es:

$$VAN = \sum_{t=0}^n \frac{F^t}{(1+i)^t} - I_0$$

Donde:

VAN: es el valor actual neto

t: es el periodo

n: numero de periodos de 4 años

i: tasa de retorno o descuento del 5% (0.05)

I₀: es la inversión que realizara la institución (23.000 \$)

La proyección de flujo de caja para 4 años, con una tasa de retorno del 5%, por lo tanto, la tasa esperada en los próximos cuatro años se refleja en la tabla

$$VAN = \left(\frac{7300}{(1-0.05)^1} + \frac{9200}{(1-0.05)^2} + \frac{12000}{(1-0.05)^3} + \frac{15800}{(1-0.05)^4} \right) - 23000$$

$$VAN = 51272.58 - 23000$$

$$VAN = 28272$$

Por tanto, la rentabilidad estimada es de 28272 donde se observa que el VAN es mayor a cero lo cual determina que el proyecto es viable y rentable

5.4.2 TASA INTERNA DE RETORNO – TIR

Es la tasa de retorno que hace que el valor actual neto sea igual a cero, es decir, el TIR es la máxima tasa de retorno que puede tener el proyecto, para que sea rentable, la fórmula es la siguiente:

$$TIR = \sum_{t=0}^n \frac{B^t}{(1+i)^t} - I_0$$

El objeto es obtener el valor de i :

$$TIR = \left(\frac{7300}{(1+i)^1} + \frac{9200}{(1+i)^2} + \frac{12000}{(1+i)^3} + \frac{15800}{(1+i)^4} \right) - 23.000 = 0$$
$$i = 0.0425$$

Es decir, el TIR es igual al 4.25%, que es la tasa máxima con la que se debe trabajar para que el proyecto sea rentable.

5.4.3 COSTO BENEFICIO - C/B

Para hallar la relación del costo beneficio se utiliza la siguiente formula:

$$\frac{\text{Costo}}{\text{Beneficio}} = \frac{\left(\frac{7300}{(1+0.0425)^1} + \frac{9200}{(1+0.0425)^2} + \frac{12000}{(1+0.0425)^3} + \frac{15800}{(1+0.0425)^4} \right)}{\left(\frac{7300}{(1+0.05)^1} + \frac{9200}{(1+0.05)^2} + \frac{12000}{(1+0.05)^3} + \frac{15800}{(1+0.05)^4} \right)}$$
$$\frac{\text{Costo}}{\text{Beneficio}} = \frac{23300}{18062} = 1.29$$

Lo cual representa que, por cada 1\$ invertido en el proyecto, la institución genera una ganancia de 0.29 centavos

6. CAPITULO VI CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

Considerando los requerimientos de la institución, se ha cumplido con el objetivo planteado cumpliendo los requerimientos solicitados por el usuario implementando el Sistema Web para el control de conductores y vehículos del transporte escolar caso: División de servicios públicos dependientes de la dirección departamental de tránsito, transporte y seguridad vial Santa Cruz.

Se desarrollo el sistema bajo la metodología de desarrollo ágil SCRUM y modelado UWE, que fue una herramienta de mucha utilidad, permitiendo realizar un seguimiento y control eficiente de todos los módulos denominados sprint, actividades relevantes en determinados tiempos para terminar el producto. A continuación, se detallan las mejoras que trajo consigo la implementación del sistema web:

- Se logro centralizar la información de conductores, vehículos, sindicatos y unidades educativas del departamento de Santa Cruz.
- Se logró Generar la Tarjeta de Identificación del Conductor Escolar (T.I.C.E.) para los conductores, mismo que es de gran utilidad para los operativos que realiza la policía boliviana
- Se logro agilizar la verificación de antecedentes de un determinado conductor.
- Se logro optimizar el tiempo en la verificación de estados de conductores entre activo e inactivo.
- Se logro representar la información de conductores mediante reportes detallando el servicio, a que sindicato pertenece, cuál es su recorrido, y demás consultas.
- Se logro representar la información de sindicatos, unidades educativas, tarjetas y demás mostrándolos en datos estadísticos, centralizar la información.
- Se logro optimizar el tiempo de búsqueda de determinados conductores, su vehículo, tipo de servicio, sindicato, entre otros.

Con respecto a la calidad y seguridad del software se aplicó la norma ISO/IEC 25010 y GQM,

obteniendo los siguientes resultados: usabilidad, seguridad, portabilidad, mantenibilidad, eficiencia de desempeño y fiabilidad tienen un rango objetivo, compatibilidad y adecuación funcional se encuentra en un rango mínimo aceptable.

Con la estimación del costo beneficio se obtuvo los siguientes resultados: el número de programadores son 8, el tiempo de ejecución son de 5 meses, el costo del sistema es de 22000 \$ y el beneficio es de 0.29 centavos por cada 1 \$ invertido.

6.2 RECOMENDACIONES

A la conclusión del presente proyecto de grado se recomienda:

- Mantener el servidor y base de datos activos y actualizados,
- Implementar una aplicación móvil para los policías de tránsito.
- Asignar personal capacitado en el manejo del sistema.
- Parametrizar el sistema web a fin de que este pueda ser implementado no solo en Santa Cruz, sino también en distintos departamentos.

BIBLIOGRAFÍA

Aguilera, P. (2011). *Redes Seguras y Seguridad Informática*. Madrid, España: Editex.

Arias Odon, F. G. (2006). *El Proyecto de Investigación* (Sexta Edición ed.). Caracas, Venezuela: Epistema.

Bahit, E. (2012). *SCRUM y Extreme Programming*. Buenos Aires, Argentina: s.n.

Bahit, E. (s.f.). *POO Y MVC EN PHP El Paradigma de la Programación Orientada a Objetos en PHP y el patron arquitectonica de software MVC*. Buenos Aires, Argentina: Creative Commons.

Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., & Sevi, L. (1995). *The COCOMO II 2.0 Software Cost Estimation Model*. California: s.n.

- Bootstrap. (2021). *Crea sitios rapidos y receptivos con Bootstrap*. Obtenido de <https://getbootstrap.com/>
- Calabrese, J., & Muños, R. (2018). *Asistente para la evaluacion de Calidad de producto de software segun la familia de norma iso/iec 25000 utilizando el enfoque GQM*. s.l.: s.n.
- Conde, O. (2016). *Sistema Municipal de infracciones de movilidad y transporte caso: Gobierno Autónomo Municipal de La Paz*. Proyecto de Grado, Carrera Informática, La Paz –Bolivia.
- Condori, M. (2016). *Sistema de Administración Municipal de la Movilidad (SAMM) del Gobierno Autónomo Municipal de La Paz caso: Gobierno Autónomo Municipal de La Paz – GAML P*. Proyecto de Grado, Carrera Informática, La Paz –Bolivia
- Deener, P., Benefield, G., Lerman, C., & Bodde, V. (2009). *The SCRUM Primer*. San Francisco, California: Creative Commons.
- Gomes, A., Migani, S., Lopez, C., & Otazu, A. (20 de noviembre de 2021). *Wordpress*. Obtenido de <https://blogadmi1.files.wordpress.com>
- Instituto de Informática - Unidad de Investigación de programación e Ingeniería de software. (2016). *UWE - Ingeniería Web basado en uml*. Obtenido de <https://uwe.pst.ifi.lmu.de/index.html>
- Irene, M. (2018). *Introducción a la seguridad informática y el analisis de vulnerabilidades*. Ecuador: Area de Innovación S.L.
- ISO-25000. (2021). *ISO 25000 Calidad de Software y Datos*. Obtenido de <https://iso25000.com/index.php/normas-iso-25000>
- ISO-25010. (2021). *ISO/IEC 25010 Calidad de Software y datos*. Obtenido de <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>

- Kniberg, H., & Skarin, M. (2010). *Kanban y SCRUM - Obteniendo lo mejor de ambos*. Estados Unidos: C4Media.
- Larman, C. (2003). *UML y Patrones* (Segunda Edición ed.). Madrid, España: Pearson Educacion S.A.
- Leokhoa. (2020). *Laragon*. Obtenido de <https://laragon.org/>
- Otwell, T. (2021). *Laravel*. Obtenido de <https://laravel.com/>
- Merino, A. (2016). *Sistema de Matrícula y Consulta de Notas para la Universidad Peruana Austral del Cusco*. Proyecto de Grado, Universidad Austral del Cusco, Cusco – Perú.
- Palacios, J. (2015). *SCRUM Manager I*. España: Safe Creative.
- Palacios, J., & Ruata, C. (2009). *SCRUM Manager Gestión de Proyectos*. España: Safe Creative.
- Pressman, R. S. (2010). *Ingeniería de Software UN ENFOQUE PRACTICO* (Septima Edición ed.). Mexico: Sponsor.
- Quispe, A., Rocha Aranibar, V., Monasterio Benegas, E., & Pereira Loza, J. (2018). *Registro para los buses de servicio publico que realizan el transporte escolar TARJETA DE IDENTIFICACION DEL CONDUCTOR ESCOLAR T.I.C.E*. Santa Cruz: s.l.
- Rumbaugh, J., Jacobson, I., & Booch, G. (2007). *El lenguaje de modelado unificado*. Madrid, España: Pearson educacion S.A.
- Sommerville, I. (2005). *Ingeniería de Software* (Septima Edición ed.). Madrid: Pearson Educacion S.A.
- Trigas Gallego, M. (s.f.). *Gestión de Proyectos Informáticos Metodología SCRUM*.

You, E. (2021). *VUE el marco de javascript progresivo*. Obtenido de <https://vuejs.org/>

ANEXOS

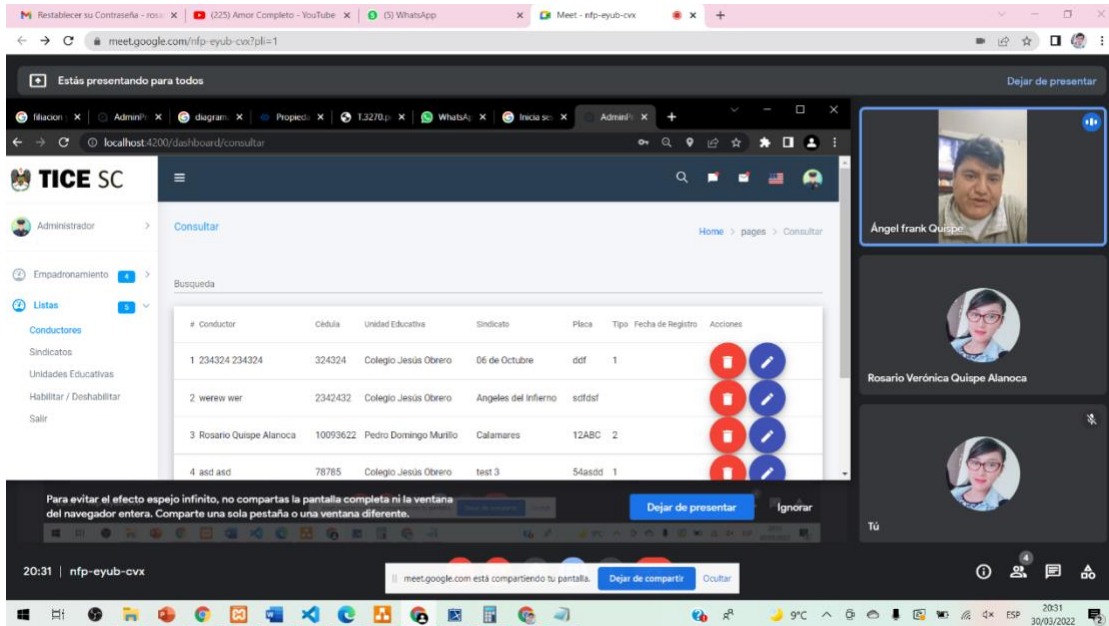


Figura 1 Reunión de Seguimiento con solicitante
Fuente: Elaboración propia

A continuación, se muestra el contenido que ofrece la página principal del sistema.



Figura 2 Acceso al Sistema
Fuente: Elaboración propia

TICE SC

Administrador

Empadronamiento

Conductor

Sindicato

Unidad Educativa

Estadísticas

Listas

Datos Conductor

A continuación ingrese información del conductor

Número de Registro: COND-033

Documento de Identidad: 3334556

Lugar de Expedición: SC

Nombres: Juan Alberto

Apellidos: Lozada Cabrera

Fecha de Nacimiento: 29/05/1974

Tipo de Sangre: O+

Dirección: Urb. Tobarochi C. Sirani N° 27

Juan Alberto Lozada Cabrera
Información Adicional

Imagen: Imagen1.jpg

Licencia de Conducir: 35689458

Sindicato: 06 de Octubre

Unidad Educativa: Nuestra Señora de Santa Cruz

Fecha de Inscripción: 30/3/2022

Categoría: B

Siguiete

Figura 3 Registro de Conductor
Fuente: Elaboración propia

TICE SC

Administrador

Empadronamiento

Conductor

Sindicato

Unidad Educativa

Estadísticas

Listas

Si el conductor ya existiese y desea renovar su tarjeta, debe consultar su existencia en el menu de listado de conductores

1 Conductor

2 Vehículo

3 Tarjeta TICE

Datos Vehículo

A continuación ingrese información del vehículo

Código de Vehículo: VSC-0001

Número de Placa: 1245ASD

Póliza: 456789A

Procedencia: China

Chasis: 14AS458S56A6

Color: Blanco

Tipo de Servicio: Taxi

Cilindrada: 2500

1245ASD
Toyota 1990

Imagen: 20200806...44447.jpg

Marca: Toyota

Modelo: 1990

Anterior

Siguiete

Figura 4 Registro de Vehículo
Fuente: Elaboración propia



Figura 5 Tarjeta TICE
Fuente: Elaboración propia

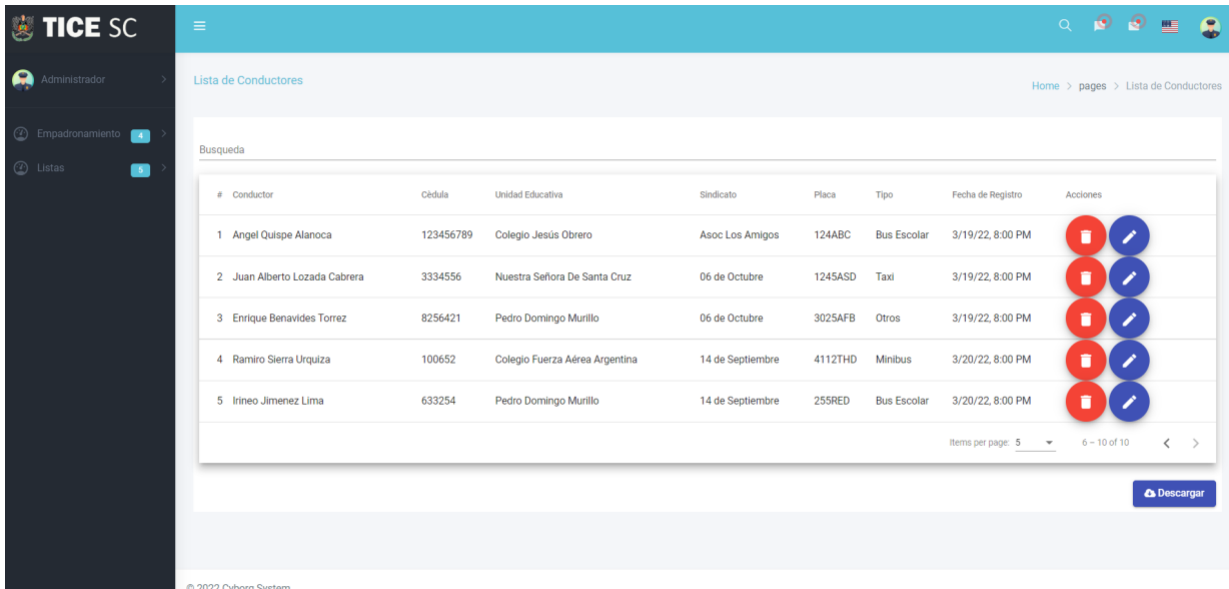


Figura 6 Lista de Conductores
Fuente: Elaboración propia

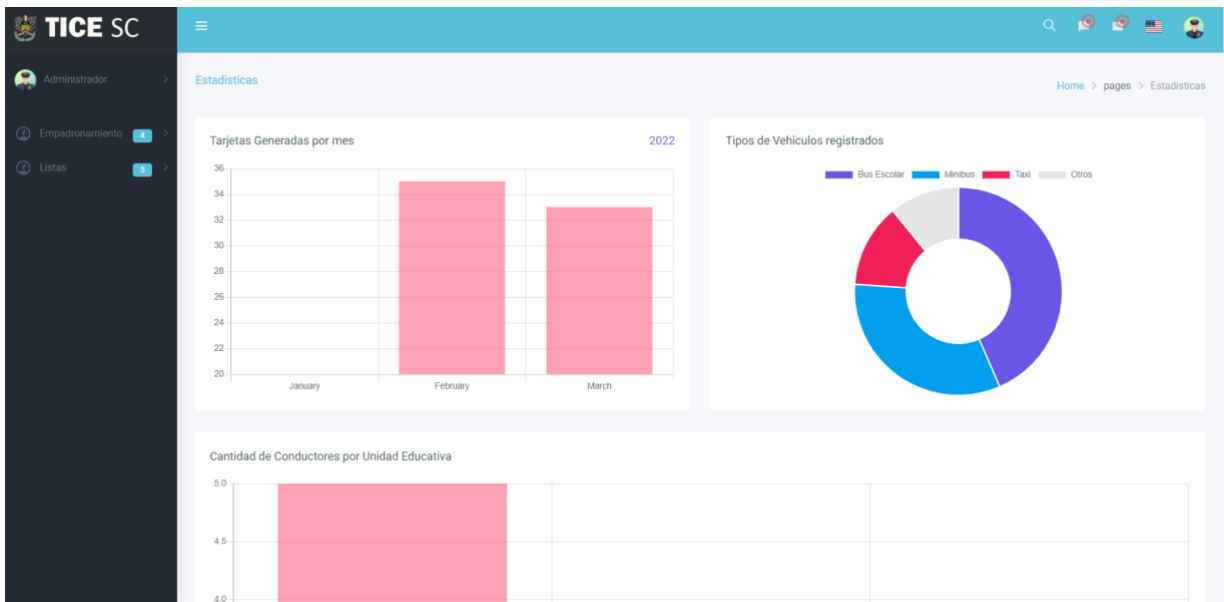


Figura 7 Estadísticas
Fuente: Elaboración propia

Report: conductores (2).pdf

Lista de Conductores

Dirección Departamental de Transporte Santa Cruz

#	Conductor	Cédula	Unidad Educativa	Mediocre	Placa	Tipo	Fecha de Registro	Acciones
1	234324 234324	324324	Colégio Jesús Obispo	06 de Octubre	40F	1	3/30/22, 8:00 PM	
2	www.ww	2342432	Colégio Jesús Obispo	Angela del infierno	sdhdf		1/30/22, 8:00 PM	
3	Rosario Quique Alarcia	10009622	Pedro Domingo Murillo	Calamitas	12ABC	2	1/30/22, 8:00 PM	
4	and and	76785	Colégio Jesús Obispo	test 3	54aadf	1	2/19/22, 8:00 PM	
5	Pepeito Perez	123456	Colégio Jesús Obispo	14 de Septiembre	123ABC	Taxi	2/19/22, 8:00 PM	

Figura 8 Reportes
Fuente: Elaboración propia