

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**PROPUESTA DE UNA METODOLOGÍA BASADA EN SCRUM
PARA EL DESARROLLO DE SISTEMAS DE SOFTWARE Y DEL
PROCESO DE ASEGURAMIENTO DE LA CALIDAD PARA EL
DEPARTAMENTO DE TIC's DE EMASEO EP**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS INFORMATICOS Y DE COMPUTACIÓN**

YESSENIA SOLEDAD ERRAEZ ENRRIQUEZ
yessenia.erraez@epn.edu.ec

DIRECTOR: Msc. Marcos Raúl Córdova Bayas
raul.cordova@epn.edu.ec

Quito, noviembre, 2020

DECLARACIÓN

Yo Yessenia Soledad Erraez Enrriquez, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

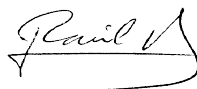
A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Yessenia Soledad Erraez Enrriquez

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Yessenia Soledad Erraez Enriquez, bajo mi supervisión.



Msc. Raúl Córdova
DIRECTOR DE PROYECTO

AGRADECIMIENTO

A Dios y a mi Virgencita del Cisne por guiarme de su mano para poder culminar mi carrera.

A mi madre, por su apoyo incondicional cuando más lo he necesitado, gracias por darme tranquilidad y animarme cuando he estado a punto de darme por vencida. Gracias a sus sacrificios y oraciones ahora estoy cumpliendo este gran sueño.

A mi hermanita Anabel, por su cariño y por sacarme siempre una sonrisa con sus ocurrencias, definitivamente mi inspiración para salir adelante y ser mejor persona.

A mi tía Rosario, por el gran amor que me ha dado desde que era una niña hasta ahora a pesar de la distancia, por apoyarme en todos mis sueños siempre, gracias por creer en mí cuando ni yo mismo lo hacía y por todos sus sacrificios para que yo pueda salir adelante.

A mis tíos: Manuel, José, Cristóbal, Vicente y Bolívar gracias por quererme como a una hija y hacer de mi infancia la etapa más hermosa de mi vida, por sus consejos, por cuidarme siempre y por apoyarme durante toda mi vida.

A Jona, por compartir muchos momentos felices conmigo, brindarme su apoyo y darme ánimos siempre.

A mis amigos por todos los momentos grandiosos que hemos vivido, por sus consejos, por las risas, por los llantos y por compartir esta maravillosa etapa de mi vida, ellos saben quiénes son y son los mejores.

A mi tutor el Msc. Raúl Córdova, muchas gracias por confiar en mí, por su infinita paciencia y por guiarme a lo largo de este proyecto, ha sido un camino lleno de aprendizaje.

Al equipo del Área de Desarrollo de EMASEO en especial al Ing. Eduardo Loachamin, muchas gracias por creer en mí y brindarme su apoyo para poder llevar a cabo este proyecto, gracias por compartir su conocimiento y experiencias conmigo.

A mi querida universidad, la “Escuela Politécnica Nacional” por abrirme las puertas y permitirme ser una profesional.

DEDICATORIA

A mis abuelitos Angelito y Zoilita que ya no están conmigo, pero que me dieron su amor en vida, gracias por darme la hermosa familia que tengo.

A mi familia quienes han estado conmigo siempre tanto en los momentos buenos como en los malos, sin su amor, apoyo y sacrificio, yo no estaría cumpliendo este sueño, por eso este trabajo está dedicado para ustedes, son mi orgullo y mi mayor motivación para salir adelante.

A mis pequeños Donking, Osito, Chester y Jake por darme su amor puro e incondicional y por desvelarse conmigo durante largas jornadas de estudio.

CONTENIDO

1. INTRODUCCIÓN	1
1.1. Análisis de la situación actual del Área de Desarrollo en EMASEO EP	1
1.2. Desarrollo de Software	3
1.3. Lenguaje Unificado de Modelado (UML).....	6
1.4. Scrum.....	9
1.4.1. Definición de Scrum.....	9
1.4.2. Roles	9
1.4.3. Artefactos de Scrum [10]	10
1.4.4. Eventos de Scrum	10
1.4.5. Puntos de historia	11
1.4.6. Ciclo de vida de Scrum	12
1.5. Proceso de Aseguramiento de la Calidad de Software	15
2. METODOLOGÍA BASADA EN SCRUM Y UML PARA EL DESARROLLO DE SISTEMAS DE SOFTWARE Y PROCESO DE SQA PARA EL AREA DE DESARROLLO DE EMASEO-EP	17
2.1. Descripción del Proceso actual de Desarrollo de software en EMASEO EP	17
2.2. Justificación del uso de Scrum para la elaboración de la metodología de desarrollo	18
2.3. Propuesta de metodología de Desarrollo basada en Scrum para EMASEO EP	19
2.3.1. Inicio del proyecto.....	19
2.3.2. Planificación	22
2.3.3. Desarrollo del Sprint	24
2.3.4. Revisión del Sprint.....	24
2.3.5. Implementación	25
2.3.6. Cierre del proyecto	26
2.4. Propuesta del proceso de aseguramiento de la calidad.....	27
2.4.1. Evaluación de historias de usuario (Product Backlog).....	27
2.4.2. Evaluación de actividades (Sprint Backlog)	29
2.4.3. Evaluación de criterios de aceptación	31
2.4.4. Evaluación de cumplimiento de la metodología	33
2.4.5. Evaluación del nivel de compromiso de las partes interesadas.....	36
2.4.6. Evaluación del cumplimiento de estándares	38
2.4.7. Pruebas sugeridas.....	40
2.4.8. Evaluación del producto del sprint	48
3. CASO DE ESTUDIO: MÓDULO DE LOGISTICA-SISTEMA DE GESTIÓN DE OPERACIONES	51

3.1.	Aplicación de la metodología híbrida (Scrum + UML) en el desarrollo del Módulo de Logística del SGO	51
3.1.1.	Inicio del proyecto.....	51
3.1.2.	Planificación	79
3.1.3.	Revisión de los sprints.....	87
3.1.4.	Implementación	102
3.1.5.	Cierre del proyecto	102
3.2.	Proceso de Aseguramiento de la calidad aplicado al Módulo de Logística	103
3.2.1.	Evaluación de historias de usuario (Product Backlog).....	103
3.2.2.	Evaluación del Sprint Backlog.....	104
3.2.3.	Evaluación de criterios de aceptación.....	105
3.2.4.	Evaluación del cumplimiento de metodología	111
3.2.5.	Evaluación del nivel de compromiso de las partes interesadas.....	112
3.2.6.	Evaluación del cumplimiento de estándares	112
3.2.7.	Plan de pruebas.....	113
3.2.8.	Evaluación del producto del sprint	120
4.	CONCLUSIONES Y RECOMENDACIONES	123
4.1.	Conclusiones.....	123
4.2.	Recomendaciones.....	124
5.	BIBLIOGRAFÍA	125
6.	ANEXOS.....	127

INDICE DE TABLAS

Tabla 1.1 Puntos de historias de usuario	12
Tabla 2.1 Roles de Scrum.....	20
Tabla 2.2 Product Backlog	20
Tabla 2.3 Criterios de aceptación.....	21
Tabla 2.4 Cronograma del proyecto	22
Tabla 2.5 Descripción de historias de usuario.....	23
Tabla 2.6 Priorización de historias de usuario.....	23
Tabla 2.7 Nivel de dependencia HU.....	24
Tabla 2.8 Revisión del Sprint	25
Tabla 2.9 Escala de evaluación de Historias de usuario	28
Tabla 2.10 Formato de evaluación de Historias de Usuario	29
Tabla 2.11 Escala de evaluación de Sprint Backlog.....	30
Tabla 2.12 Criterios de evaluación de tareas	30
Tabla 2.13 Escala de evaluación para criterios de aceptación	32
Tabla 2.14 Criterios de evaluación de Criterios de Aceptación	32
Tabla 2.15 Criterios de cumplimiento de metodología.....	35
Tabla 2.16 Escala de evaluación del cumplimiento de la metodología	35
Tabla 2.17 Formato de evaluación de cumplimiento de la metodología	36
Tabla 2.18 Evaluados y Evaluadores Cumplimiento de responsabilidades	37
Tabla 2.19 Evaluados y Evaluadores Capacidad de trabajar en equipo	37
Tabla 2.20 Escala de Evaluación Nivel de compromiso partes interesadas	37
Tabla 2.21 Formato de evaluación del Nivel de compromiso de las partes interesadas ...	38
Tabla 2.22 Escala de evaluación Cumplimiento de Estándares	39
Tabla 2.23 Formato de evaluación Cumplimiento de Estándares.....	40
Tabla 2.24 Herramientas para pruebas.....	45
Tabla 2.25 Formato de evaluación Producto del Sprint.....	49
Tabla 3.1 Roles y responsabilidades de Scrum	51
Tabla 3.2 Product Backlog Tareas	52
Tabla 3.3 Descripción caso de uso Registrar frecuencia.....	55
Tabla 3.4 Descripción de caso de uso Editar frecuencia	55
Tabla 3.5 Descripción de caso de uso Consultar frecuencia	55
Tabla 3.6 Descripción caso de uso Registrar horario	56
Tabla 3.7 Descripción caso de uso Editar horario	57
Tabla 3.8 Descripción caso de uso Consultar horario	57
Tabla 3.9 Descripción de caso de uso Registrar tipo de servicio.....	58
Tabla 3.10 Descripción de caso de uso Editar tipo de servicio.....	58
Tabla 3.11 Descripción de caso de uso Consultar tipo de servicio.....	59
Tabla 3.12 Descripción de caso de uso Cargo.....	59
Tabla 3.13 Descripción del caso de uso Editar cargo.....	60
Tabla 3.14 Descripción del caso de uso Consultar cargo.....	60
Tabla 3.15 Descripción del caso de uso Registrar empleado.....	61
Tabla 3.16 Descripción del caso de uso Editar datos empleado	62
Tabla 3.17 Descripción del caso de uso Consultar empleado	62
Tabla 3.18 Descripción del caso de uso Registrar maquinaria.....	63

Tabla 3.19 Descripción del caso de uso Editar maquinaria	64
Tabla 3.20 Descripción del caso de uso Consultar maquinaria	64
Tabla 3.21 Descripción del caso de uso Registrar administración zonal	65
Tabla 3.22 Descripción del caso de uso Editar administración zonal	66
Tabla 3.23 Descripción del caso de uso Consultar administración zonal	66
Tabla 3.24 Descripción del caso de uso Registrar sector.....	67
Tabla 3.25 Descripción del caso de uso Editar sector.....	68
Tabla 3.26 Descripción del caso de uso Consultar sector	68
Tabla 3.27 Descripción del caso de uso Eliminar sector	68
Tabla 3.28 Descripción del caso de uso Registrar ruta ubicación	69
Tabla 3.29 Descripción del caso de uso Editar ruta ubicación	70
Tabla 3.30 Descripción del caso de uso Consultar ruta ubicación.....	70
Tabla 3.31 Descripción del caso de uso Registrar programación de ruta.....	71
Tabla 3.32 Descripción del caso de uso Editar programación de ruta	72
Tabla 3.33 Descripción del caso de uso Consultar programación de ruta	72
Tabla 3.34 Descripción de caso de uso Reasignar ruta	73
Tabla 3.35 Descripción de caso de uso Consultar rutas asignadas a un supervisor	74
Tabla 3.36 Descripción del caso de uso Generar reporte.....	75
Tabla 3.37 Cronograma del proyecto	76
Tabla 3.38 Historias de usuario.....	80
Tabla 3.39 Prioridad de Historias de Usuario	81
Tabla 3.40 Tareas Sprint 1.....	82
Tabla 3.41 Tareas Sprint 2.....	83
Tabla 3.42 Tareas Sprint 3.....	85
Tabla 3.43 Tareas Sprint 4.....	86
Tabla 3.44 Tareas Sprint 5.....	86
Tabla 3.45 PLA-001 Gestionar frecuencia	89
Tabla 3.46 PLA-002 Gestionar horario.....	90
Tabla 3.47 PLA-003 Gestionar tipo de servicio	91
Tabla 3.48 PLA-004 Gestionar talento humano-cargos.....	93
Tabla 3.49 PLA-004 Gestionar talento humano-Empleados.....	94
Tabla 3.50 PLA-005 Gestionar maquinaria	95
Tabla 3.51 PLA-006 Gestionar administración zonal.....	96
Tabla 3.52 PLA-007 Gestionar sectores	98
Tabla 3.53 PLA-008 Gestionar ruta ubicación.....	99
Tabla 3.54 PLA-009 Gestionar programación base de ruta.....	100
Tabla 3.55 PLA-010 Gestionar cambio de supervisor de ruta	101
Tabla 3.56 PLA-011 Reporte de programación diaria	102
Tabla 3.57 Evaluación del Product Backlog	104
Tabla 3.58 Evaluación del Sprint Backlog	105
Tabla 3.59 Evaluación de criterios de aceptación	111
Tabla 3.60 Evaluación del cumplimiento de la metodología	112
Tabla 3.61 Evaluación del nivel de compromiso de las partes interesadas	112
Tabla 3.62 Evaluación del cumplimiento de estándares.....	113
Tabla 3.63 Matriz RACI.....	115
Tabla 3.64 Resultados pruebas rendimiento 1 usuario	115

Tabla 3.65 Resultados pruebas rendimiento 10 usuarios.....	116
Tabla 3.66 Resultados pruebas rendimiento 20 usuarios.....	117
Tabla 3.67 Resultados pruebas rendimiento 50 usuarios.....	117
Tabla 3.68 Resultados pruebas rendimiento 1 usuario	118
Tabla 3.69 Resultados pruebas rendimiento 10 usuarios.....	118
Tabla 3.70 Resultados pruebas rendimiento 20 usuarios.....	118
Tabla 3.71 Resultados pruebas rendimiento 50 usuarios.....	119
Tabla 3.72 Resultados pruebas de humo.....	120
Tabla 3.73 Evaluación incremento sprint 1	120
Tabla 3.74 Evaluación incremento sprint 2	121
Tabla 3.75 Evaluación incremento sprint 3	121
Tabla 3.76 Evaluación incremento sprint 4	121
Tabla 3.77 Evaluación incremento sprint 5	122

INDICE DE FIGURAS

Figura 2.1. Cuadrantes de Agile Testing	42
Figura 3.1 Caso de uso Frecuencia	54
Figura 3.2 Caso de uso Horario	56
Figura 3.3 Caso de uso Tipo de servicio	57
Figura 3.4 Caso de uso Talento humano	59
Figura 3.5 Caso de uso Empleado-Talento humano	61
Figura 3.6 Caso de uso Maquinaria	63
Figura 3.7 Caso de uso Administración zonal	65
Figura 3.8 Caso de uso Sectores.....	66
Figura 3.9 Caso de uso Ruta ubicación	69
Figura 3.10 Caso de uso Programacion base	71
Figura 3.11 Caso de uso Cambio de supervisor de ruta	73
Figura 3.12 Caso de uso Reporte de programación diaria	74
Figura 3.13 ESB (Enterprise Service Bus)	77
Figura 3.14 Arquitectura Java Enterprise Edition	78
Figura 3.15 Tablero Sprint 1-Trello	82
Figura 3.16 Detalles de tarea - Gestionar frecuencia	82
Figura 3.17 Tablero Sprint 2-Trello	84
Figura 3.18 Tablero Sprint 3-Trello	85
Figura 3.19 Tablero Sprint 4-Trello	86
Figura 3.20 Tablero Sprint 5-Trello	87
Figura 3.21 Diagrama de clases Módulo de Logística	87

RESUMEN

El presente proyecto de titulación contiene dos propuestas en el área de Ingeniería de Software para el desarrollo de sistemas en la empresa EMASEO EP. En primer lugar, se desarrolla la propuesta de una Metodología de Desarrollo de Software basada en Scrum y el Lenguaje Unificado de Modelado (UML) y en segundo lugar se desarrolla la propuesta para el proceso Aseguramiento de la Calidad, elaboradas para el Área de Desarrollo del Departamento de TIC's de la Empresa Pública Metropolitana de Aseo EMASEO EP, del Municipio del Distrito Metropolitano de Quito.

En el Capítulo 1 se hace un análisis de la situación actual del Área de Desarrollo de la empresa, en donde se describen los principales inconvenientes encontrados y que limitan a los equipos aplicar sus capacidades al máximo y desarrollar software que cumpla con estándares de calidad y con las necesidades de los usuarios de la empresa. También contiene la descripción del proceso de desarrollo de software, UML, SCRUM y del proceso de Aseguramiento de la Calidad de Software, que son la base para la posterior elaboración de las propuestas.

En el Capítulo 2 se presenta la propuesta de la metodología basada en SCRUM y UML; en este capítulo se explica paso a paso como aplicarla en cada una de sus etapas, se describen las herramientas y los formatos de documentación sugeridos, siempre manteniendo el principio ágil de documentar solamente lo necesario. También se presenta la propuesta del Proceso de Aseguramiento de la Calidad, que contiene las métricas para evaluar el proceso de desarrollo de software con la metodología propuesta.

En el Capítulo 3 se aplican las dos propuestas a un caso de estudio propio de la empresa, que consistió en el desarrollo el Módulo de Logística del Sistema de Gestión de Operaciones para EMASEO EP, lo que permitió validar de manera satisfactoria las propuestas presentadas en este proyecto.

Finalmente, en el Capítulo 4 se presentan las conclusiones y recomendaciones del Proyecto de Titulación elaborado.

Palabras clave: Ingeniería de Software, Procesos Ágiles, SCRUM, Aseguramiento de la Calidad de Software, UML.

ABSTRACT

This degree project contains two proposals in the area of Software Engineering for development systems in the EMASEO EP company. First it presents the proposal for a Software Development Methodology based on Scrum and Unified Modeling Language (UML). Second, it develops the proposal for the Quality Assurance process for the Development Area of “Empresa Pública Metropolitana de Aseo” EMASEO-EP TIC’s Department, part of the Metropolitan District of Quito Municipality.

In Chapter 1, an analysis of the current situation of the company's Development Area is made, describing the main problems that restrict the teams apply their full capacities and that don't allow them developing software that accomplish company users quality standards and requirements. It also contains the description of the software development process, UML, SCRUM and the Software Quality Assurance process, which are the basis for developing the proposals.

Chapter 2 presents the methodology proposal based on SCRUM and UML; this chapter explains step by step how to apply it in each of their phases, describing the tools and documentation formats suggested, always maintaining the agile principle of documenting only the necessary. It also presents the Quality Assurance Process proposal, that contains the metrics to evaluate the software development process based in the proposed methodology.

In Chapter 3, both proposals are applied to developing a case study in the company, related to the development of the Operations Management System Logistics Module for EMASEO EP, showing a satisfactory validation of the proposals given in the project.

Finally, Chapter 4 contains the degree project conclusions and recommendations.

Keywords: Software Engineering, Agile Process, SCRUM, Software Quality Assurance, UML

1. INTRODUCCIÓN

1.1. Análisis de la situación actual del Área de Desarrollo en EMASEO EP

El departamento de Tecnología de la Información y Comunicación (TIC's) de la Empresa Pública Metropolitana de Aseo (EMASEO EP), actualmente cuenta con un Área de Desarrollo creada hace tres años, que tiene como objetivo atender las necesidades de desarrollo de software de los procesos principales y de apoyo de la empresa.

Sin embargo, hace algunos años el desarrollo de sistemas de software se hacía dentro de cada área de la empresa. Es decir, se tenía un programador distinto en cada área, por ende, se utilizaban distintos procesos de desarrollo y lenguajes de programación, sin hacer un análisis previo de los sistemas a desarrollarse y de su complejidad, lo que resultaba en un gasto excesivo. Todo esto traía consecuencias que afectaban la calidad de los sistemas creados, ya que, al no tener una metodología definida, el mantenimiento o integración de los sistemas de software se volvió una tarea complicada.

Todo lo anterior motivó a que surgiera la necesidad de crear un Área de Desarrollo de Software cuyo objetivo era atender todas las necesidades de desarrollo de las demás áreas de la empresa, utilizando una metodología adecuada.

El Área de Desarrollo de Software de EMASEO EP se encuentra conformada por un grupo pequeño de cinco personas, cada una trabajando en distintos proyectos según la demanda de sistemas requeridos en la empresa. Por lo general el tipo de sistemas de software que desarrollan son de pequeña y mediana complejidad; por otro lado, también se presentan problemas en el mantenimiento de sistemas antiguos, puesto que carecen de documentación y no fueron desarrollados siguiendo un proceso o metodología de desarrollo de software bien definida.

En la actualidad, el desarrollo de sistemas de software en la empresa se lleva a cabo utilizando el modelo en espiral de seis regiones de Bohem. En este modelo, cada fase del desarrollo de software se representa como una espiral, si existe un nuevo requerimiento factible de implementar únicamente se añade una nueva vuelta a la espiral. En cada iteración de la espiral se deben realizar las siguientes tareas: comunicación con el cliente, planificación, análisis de riesgos, ingeniería, construcción y adaptación del software y evaluación del cliente. Además, también se trabaja con documentación que considera: resumen ejecutivo, documento de alcance, matriz de riesgos, informa de avance, actas de entrega y cierre del proyecto y un documento de gestión de cambios [1].

Al utilizar como guía el modelo de seis regiones de Bohem, no se hizo un análisis previo del tipo de sistemas de software que se desarrollan y del ambiente y condiciones de trabajo dentro del área. Además, este modelo no se está aplicando correctamente, ya que no se siguen las actividades propuestas en el mismo y en el desarrollo de algunos sistemas de software ni siquiera se lo ha tomado en cuenta.

Existen algunos inconvenientes dentro del Área de Desarrollo que afectan la productividad y el desarrollo de sistemas de software. Uno de los principales, es que cada integrante del equipo de desarrollo trabaja de manera independiente, es decir, cada uno desarrolla un sistema de software distinto, de manera que la misma persona se encarga del front-end, back-end, documentación y pruebas. En muchos de los sistemas de software desarrollados no se trabaja en la documentación. Además, cada programador utiliza diferentes estándares de programación, estándares base de datos, lenguajes de programación y arquitectura.

También se presentan problemas en el proceso de desarrollo de software y tiempos de entrega del producto final. En muchos casos los sistemas de software propuestos en el POA (Plan Operativo Anual) no se desarrollan completamente, lo cual afecta al equipo de desarrollo y a las áreas que solicitan los sistemas. Es importante que se trabaje correctamente en cada etapa del desarrollo de software, desde el análisis de requerimientos, hasta el proceso de pruebas y postproducción del sistema.

En la entrevista realizada con cada uno de los integrantes del equipo de desarrollo de software, uno de ellos mencionaba que todos los problemas de planificación de actividades y tiempos de entrega surgían desde que no se hacía un análisis de requerimientos adecuado.

Esto traía problemas en la planificación de tiempos, ya que no se analizaba conscientemente el tiempo que se emplearía en desarrollar un sistema o en cada etapa del mismo; también mencionaban que cuando el sistema de software estaba en etapa avanzada surgían cambios de última hora por parte del cliente o, a su vez, este solicitaba una gran cantidad de nuevos requerimientos, lo cual hacía que los tiempos definidos ya no se cumplan o se dé menos importancia al proceso de pruebas y postproducción.

“Otro problema frecuente es la comunicación con el usuario”, mencionaba otro miembro del equipo de desarrollo, ya que cuando el usuario no acude a las reuniones planificadas y no expresa claramente sus necesidades respecto al sistema de software a desarrollarse, es probable que no se definan todos los requerimientos que necesitaba tener el sistema para cumplir con el alcance y objetivo definidos inicialmente.

A pesar de haberse definido un modelo de desarrollo de software, existen actividades muy importantes dentro del mismo a las cuales no se les ha dado la debida importancia, por ejemplo, la documentación, el análisis y definición de los requerimientos, fases de pruebas y control de calidad en etapas de pre y post producción del software.

Los problemas mencionados afectan tanto a la calidad del producto final y tiempos de entrega como a la productividad del equipo de desarrollo, ya que los integrantes no trabajan en equipo y por ende no se aprovecha al máximo las capacidades más destacadas de cada uno de ellos.

Tomando en cuenta la situación actual y los principales problemas que se presentan en el desarrollo de los sistemas de software, se presentó la necesidad de definir una metodología de desarrollo adecuada para el tipo de sistemas que se desarrollan dentro del Área de

Desarrollo, junto a un proceso de Aseguramiento de la Calidad que permita brindar más seguridad a los usuarios sobre la calidad de los sistemas desarrollados y a ayudar al equipo de desarrollo a trabajar de manera organizada y eficiente, obteniendo así un aumento en la productividad para llevar a cabo todos los proyectos definidos en su Plan Operativo Anual (POA).

1.2. Desarrollo de Software

En los últimos años el software se ha convertido en una pieza fundamental en nuestra sociedad, ya que permite gestionar un recurso muy importante: la información. Para desarrollar software no basta con seguir un proceso ordenadamente, también hay que encontrar una solución adecuada acorde al tipo de sistema que se desee implementar.

La Ingeniería de software es una disciplina que comprende todos los aspectos para producir software, desde la especificación de requerimientos hasta la etapa de mantenimiento cuando esté se empieza a utilizar [2].

El desarrollo de software se lleva a cabo siguiendo un proceso específico acorde al tipo de software a desarrollar y a los resultados que se desea obtener. Básicamente existen dos tipos de procesos que se pueden aplicar al desarrollar software: procesos tradicionales y procesos ágiles.

Los procesos tradicionales tienen un enfoque predictivo, donde se sigue un proceso secuencial, en una sola dirección y sin marcha atrás. La estimación y levantamiento de requerimientos se realiza únicamente al inicio del proyecto y no se esperan cambios durante el desarrollo, por tal razón se dice que es un proceso predictivo [2].

Por otro lado, los procesos ágiles se caracterizan por ser más flexibles y adaptativos, esto quiere decir que no se cierran al cambio, más bien lo aceptan con normalidad. Para aplicar este tipo de procesos debe existir una excelente comunicación con el cliente, debido a que los requerimientos o características del proyecto pueden cambiar en cualquier momento [2].

Teniendo en cuenta estos conceptos se puede elegir el proceso que mejor se adapte al tipo de sistema a desarrollar y al equipo de trabajo, de esta manera se asegurará parte de la calidad del producto final.

Procesos de Desarrollo de Software

Un proceso de desarrollo de software es un conjunto de actividades que se relacionan y conducen a la elaboración de un producto de software. Estas actividades pueden incluir el desarrollo de software desde cero en un lenguaje de programación como Java o C. Sin embargo, el software a nivel profesional se desarrolla extendiendo o modificando sistemas existentes, configurando o integrando el software comercial o componentes del sistema [3].

Existen diferentes tipos de procesos de software, pero todos incluyen las cuatro actividades claves en la ingeniería de software: especificación de software, diseño e implementación, validación y evolución. En la práctica estas actividades se vuelven más complejas e

incluyen otras actividades como validación de requerimientos, diseño arquitectónico, entre otras.

Los procesos de software son complejos y se apoyan en personas con capacidad de juzgar y tomar decisiones. La mayor parte de organizaciones han adaptado y diseñado sus propios procesos de desarrollo de software. Para sistemas críticos, se requiere un proceso de desarrollo muy estructurado. En cambio, para los sistemas empresariales, con requerimientos que cambian rápidamente, es probable que sea más efectivo utilizar un proceso menos formal y más flexible.

Los procesos de software se clasifican como dirigidos por un plan o como procesos ágiles. Los procesos dirigidos por un plan son aquellos donde todas las actividades se planean con anticipación y el avance se mide contra dicho plan. En los procesos ágiles la planeación es incremental y más fácil de modificar el proceso para reflejar los requerimientos cambiantes del cliente [3]. Estos procesos son conocidos como tradicionales, para diferenciarlos de los ágiles.

Modelos de Procesos de Desarrollo de Software

Un modelo de proceso de desarrollo de software es una representación simplificada de un proceso. Cada modelo representa a otro desde una perspectiva distinta, por ende, ofrece solo información parcial sobre dicho proceso. Por ejemplo, un modelo de actividad del proceso muestra únicamente las actividades y su secuencia, pero deja de lado los roles de las personas que intervienen en estas actividades. Existen algunos modelos genéricos y las descripciones que se muestran a continuación no son definitivas. Más bien, son abstracciones del proceso que se utilizan para explicar distintos enfoques del desarrollo de software.

Entre los modelos de proceso tradicionales más conocidos tenemos:

Modelo en cascada: toma las actividades principales del proceso de especificación, desarrollo, validación y evolución y, luego, los representa como fases separadas del proceso, tales como definición de requerimientos, diseño de software, implementación, pruebas, etc.

Modelo incremental: vincula las actividades de especificación, desarrollo y validación. El sistema se desarrolla como una serie de versiones llamadas incrementos, y cada versión añade una nueva funcionalidad a la versión anterior.

Modelos evolutivos: el software, al igual que los sistemas complejos evoluciona con el tiempo. Es normal que los requerimientos del negocio o del producto cambien conforme avanza el desarrollo, lo cual hace que se necesite un modelo de proceso diseñado explícitamente para adaptarse a un producto que evoluciona. Los modelos evolutivos son iterativos. Se caracterizan porque se desarrollan cada vez versiones más completas del software. A continuación, se describen dos modelos representativos del proceso evolutivo:

- **Modelo de prototipos:** la clave de este modelo es la comunicación. Los participantes deben reunirse para definir los objetivos generales del software e identificar la mayor parte de requerimientos, de esta manera podrán asignar la mayor prioridad a quienes lo requieran. Luego se planea una iteración para hacer el prototipo y se lleva a cabo el modelado. Si va a construirse un prototipo se puede utilizar fragmentos de sistemas existentes o aplicar herramientas que permitan generar rápidamente sistemas que funcionen satisfactoriamente.
- **Modelo espiral:** se acopla con naturaleza iterativa de hacer prototipos con aspectos controlados y sistemáticos del modelo en cascada. Tiene dos características distintivas principales. La primera es el enfoque cíclico para el crecimiento incremental en la evolución de la definición del sistema, y la segunda es el compromiso y colaboración de los participantes para encontrar una solución que satisfaga todas las partes interesadas.

Los modelos descritos anteriormente no son excluyentes entre ellos y con frecuencia se usan en conjunto, sobre todo para el desarrollo de sistemas grandes y complejos. [3].

Procesos ágiles

La principal característica de un proceso ágil es la adaptabilidad. Es decir, debe responder eficientemente a los cambios que puedan presentarse en los requerimientos durante el desarrollo de software. Pero la adaptación continua no llega muy lejos si no hay avance. Por tal razón, un proceso ágil debe adaptarse incrementalmente. Para lograr esto el equipo ágil necesita tener retroalimentación continua por parte del cliente.

El resultado de aplicar un proceso ágil se conoce como incrementos, que son las partes de un sistema que se entregan en cortos periodos de tiempo, de modo que la adaptación vaya al mismo ritmo del cambio. El enfoque iterativo permite que el cliente evalúe en forma regular el incremento de software y que de la retroalimentación necesaria al equipo ágil para que realice las adaptaciones al proceso de manera que se aproveche la retroalimentación obtenida [2].

La filosofía detrás de los procesos ágiles se refleja en el manifiesto ágil [4], que acordaron muchos de los desarrolladores líderes de estos procesos. Este manifiesto afirma:

Se están descubriendo mejores formas para desarrollar software, al hacerlo y al ayudar a otros a hacerlo. Gracias a este trabajo se ha llegado a valorar:

- *A los individuos y las interacciones sobre los procesos y las herramientas.*
- *Al software operativo sobre la documentación exhaustiva.*
- *La colaboración con el cliente sobre la negociación del contrato.*
- *La respuesta al cambio sobre el seguimiento de un plan.*

Esto significa que, aunque exista valor en los objetos de la derecha, se valorara más a los de la izquierda.

Entre los procesos ágiles más conocidos esta XP, también conocido como programación extrema. Entre otros procesos ágiles están Scrum, Kanban, Agile Inception, etc.

Aunque todos los procesos ágiles se basan en la noción del desarrollo y la entrega de incrementos, proponen diferentes procesos para lograrlo. Sin embargo, comparten una serie de principios, según el manifiesto ágil y por ende tienen mucho en común [3].

Método y Metodología

Método: es un proceso disciplinado para la generación de un conjunto de modelos que describen diferentes aspectos de un sistema usando una notación bien definida. Por lo general, los métodos se aplican a las primeras fases del ciclo de vida de desarrollo de software: análisis y diseño. También existen métodos de diseño solamente, o métodos de análisis y diseño, por ejemplo, el método de análisis y diseño de Booch [5].

Metodología: es un conjunto de métodos aplicados a lo largo del ciclo de vida de desarrollo de software, unificados por un enfoque general o filosófico. Existen metodologías de desarrollo de software estructuradas, orientadas a objetos, etc.

Ya que los métodos están compuestos de dos partes: un proceso de desarrollo y un lenguaje de modelamiento, entonces se puede concluir que al ser una metodología un conjunto de métodos estará compuesta de un proceso de desarrollo y de un lenguaje de modelamiento, que cubren todas las etapas del ciclo de vida de desarrollo de software. El ejemplo más conocido de metodología orientadas a objetos es el de la unión del Proceso Unificado (Unified Process) y UML (Unified Modeling Language) [6].

1.3. Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado (UML) es un lenguaje estándar para escribir planos de software. Puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucre una gran cantidad de software. UML es un lenguaje muy expresivo, que cubre todas las vistas necesarias para desarrollar y luego desplegar tales sistemas.

UML es un lenguaje y por tanto es solo una parte de un método de desarrollo de software. Este lenguaje es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que sea dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental [7].

Para aplicar UML es necesario conocer tres elementos fundamentales: *bloques* básicos de construcción de UML, *reglas* sobre cómo pueden combinarse con dichos bloques y algunos *mecanismos* comunes que se aplican a lo largo de todo el lenguaje. Estos elementos se describen a continuación:

Bloques básicos de UML: el vocabulario de UML incluye tres clases de bloques básicos:

- **Elementos:** son bloques básicos de construcción orientados a objetos de UML. Se utilizan para escribir modelos bien formados. En UML existen cuatro tipos de elementos: elementos estructurales, elementos de comportamiento, elementos de agrupación y elementos de anotación.
- **Relaciones:** son bloques básicos de construcción para relaciones en UML. Se utilizan para escribir modelos bien formados. En UML existen cuatro tipos de relaciones: dependencia, asociación, generalización y realización.
- **Diagramas:** es una representación gráfica de un conjunto de elementos, la mayoría de las veces se visualiza como un grafo conexo de nodos (elementos) y arcos (relaciones). Los diagramas permiten visualizar un sistema desde diferentes perspectivas, por lo tanto, un diagrama es una proyección del sistema. UML versión 2 incluye trece tipos de diagramas, a saber:
 1. Diagrama de clases
 2. Diagrama de objetos
 3. Diagrama de componentes
 4. Diagrama de estructura compuesta
 5. Diagrama de casos de uso
 6. Diagrama de secuencia
 7. Diagrama de comunicación
 8. Diagrama de estados
 9. Diagrama de actividades
 10. Diagrama de despliegue
 11. Diagrama de paquetes
 12. Diagrama de tiempos
 13. Diagrama de visión global de interacciones

A continuación, se describirán con brevedad los diagramas mencionados anteriormente:

Diagrama de clases: muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Son los más comunes en el modelado de sistemas orientados a objetos. Permite ilustrar la vista de diseño estático de un sistema.

Diagrama de componentes: representa la encapsulación de una clase, junto con sus interfaces, puertos y estructura interna, la cual está conformada por otros componentes anidados y conectores. Este diagrama cubre la vista de implementación estática del diseño de un sistema.

Diagrama de estructura compuesta: muestra la estructura interna de una clase o colaboración.

Diagrama de objetos: muestra un conjunto de objetos y sus relaciones. Representan instancias estáticas de los elementos existentes en los diagramas de clases.

Diagrama de despliegue: muestra un conjunto de nodos y sus relaciones. Este tipo de diagrama se usa para ilustrar la vista de implementación estática de una arquitectura.

Diagrama de artefactos: muestra un conjunto de artefactos y sus relaciones con otros artefactos y con las clases que implementan. Se utiliza para mostrar las unidades de implementación física del sistema.

Diagrama de casos de uso: muestra un conjunto de casos de uso, actores y sus relaciones. Estos diagramas cubren la vista de casos de uso estática de un sistema.

Diagrama de secuencia: muestra un conjunto de roles y mensajes enviados y recibidos por las instancias que desempeñan los roles.

Diagrama de comunicación: este tipo de diagrama hace énfasis en la organización estructural de los objetos que envían y reciben mensajes. Muestra un conjunto de roles, conectores entre roles y mensajes enviados y recibidos por las instancias que desempeñan los roles.

Diagrama de estados: muestra una máquina de estados, que está formada por estados, transiciones, eventos y actividades. Este diagrama muestra la vista dinámica de un sistema.

Diagrama de actividades: indica la estructura de un proceso y el flujo de control y datos paso a paso en la computación. Este diagrama cubre la vista dinámica de un sistema.

Diagrama de paquetes: muestra la descomposición del propio modelo en unidades organizativas y sus dependencias.

Diagrama de tiempos: es un diagrama de interacción que muestra los tiempos reales entre diferentes objetos o roles, en oposición a la simple secuencia relativa de mensajes.

Diagrama de visión global de interacciones: es un híbrido entre un diagrama de actividades y un diagrama de secuencia.

Reglas de UML

Como cualquier lenguaje, UML tiene un número de reglas que especifican a que debe asimilarse un modelo bien formado. Un modelo bien formado es aquel que es semánticamente auto consistente y está en armonía con todos sus modelos relacionados. UML tiene reglas semánticas y sintácticas para:

- Nombres → Como llamar a los elementos, relaciones y diagramas.
- Alcance → El contexto que da un significado específico a un nombre.
- Visibilidad → Como se pueden ver y utilizar esos nombres por otros.
- Integridad → Como se relacionan apropiada y consistentemente unos elementos con otros.
- Ejecución → Que significa ejecutar o simular un modelo dinámico.

Las reglas de UML estimulan a considerar las cuestiones más importantes de análisis, diseño e implementación que llevan a los sistemas a convertirse en modelos bien formados con el paso del tiempo [7].

Mecanismos comunes en UML

Este lenguaje se simplifica mediante la presencia de cuatro mecanismos comunes que se aplican de forma consistente a través de todo el lenguaje [7]:

1. **Especificaciones:** proporciona una base semántica que incluye a todas las partes de todos los modelos de un sistema, y cada parte está relacionada con otra de manera consistente.
2. **Adornos:** es la notación gráfica de los elementos de UML. Esta notación es única y clara y proporciona una representación visual de los aspectos más importantes de un elemento.
3. **Divisiones Comunes:** existen diferentes divisiones al modelar sistemas orientados a objetos. Entre ellas tenemos: clase-objeto en donde se modelan tanto clases como objetos y las colaboraciones que los realizan, así como operaciones y los métodos que las implementan, interfaz-implementación y tipo-rol.
4. **Mecanismos de extensibilidad:** UML no es suficiente para expresar todos los matices posibles de todos los modelos en todos los dominios y en todo momento. Por tal razón se dice que UML es un lenguaje abierto-cerrado, para así extender el lenguaje de manera controlada. Los mecanismos de extensión de UML comprenden estereotipos, valores etiquetados y restricciones.

1.4. Scrum

1.4.1. Definición de Scrum

Scrum es un marco de trabajo diseñado para lograr una colaboración eficiente entre los equipos en proyectos, que aplica un conjunto de reglas y artefactos y define roles que generan la estructura adecuada para su correcto funcionamiento.

Scrum utiliza un enfoque iterativo e incremental que está fundamentado en la teoría de control empírico de procesos. Esta teoría se fundamenta en la transparencia, inspección y adaptación[8].

SCRUM se basa en Roles, Artefactos y Eventos, descritos a continuación.

1.4.2. Roles

En Scrum existen dos tipos de roles que están divididos en dos grupos importantes.

Por un lado, están los que están comprometidos en el proyecto y proceso de desarrollo. Por otro lado, aquellos que no forman parte del proceso de Scrum, pero su participación es fundamental para llevar a cabo el proyecto [9].

Roles comprometidos en el proyecto

Scrum Master o Facilitador: es el encargado de velar por el bienestar del equipo de desarrollo y asegurarse de que se está adoptando la metodología, sus prácticas, valores y normas [8].

Product Owner o Dueño del Producto: representa la voz del cliente y es el responsable de establecer el objetivo del proyecto y maximizar el trabajo del equipo de desarrollo; tiene entre sus funciones gestionar la lista de funcionalidades requeridas o Product Backlog [8].

Scrum Team o Equipo de Desarrollo: es un grupo de personas que trabajan de manera continua en el desarrollo de un proyecto. Es importante que el equipo sea multifuncional, es decir, que los miembros del equipo tengan las habilidades necesarias para poder ejecutar las tareas que permitan entregar valor al cliente al final del sprint [10].

Roles involucrados en el proceso

Usuarios: son las personas a quienes va destinado el proyecto.

Stakeholders: son las personas que hacen que el proyecto se haga y quienes reciben un beneficio acorde a lo que se necesita.

Managers: son aquellos que establecen el ambiente adecuado para que se desarrolle el proyecto.

1.4.3. Artefactos de Scrum [10]

Product Backlog o Pila del producto: es una lista priorizada de características, funcionalidades técnicas, requerimientos, bugs y mejoras del proyecto. Esta lista está en constante evolución y es tomada en cuenta durante todo el desarrollo del proyecto. El principal responsable de desarrollarla y gestionarla es el Product Owner.

Sprint Backlog o Pila del Sprint: es la lista de tareas o actividades para desarrollar los requerimientos propuestos para el sprint. También incluye el nivel de dificultad que poseen los requerimientos. El Scrum Team es el responsable de desarrollar esta lista.

Incremento: es el entregable de un sistema desarrollado en un sprint. Este entregable implica que el producto está completamente finalizado y se podría desplegar a producción inmediatamente después de revisarlo en el sprint con mínimas tareas o cambios, que bien pueden ser, pruebas o documentación.

1.4.4. Eventos de Scrum

Sprint: es el ciclo de producción dentro de un desarrollo iterativo e incremental en donde se construye el incremento, la duración estándar es de 30 días (puede variar desde una semana a un mes).

Daily Sprint o Scrum Diario: es una reunión de máximo 15 minutos en donde cada miembro del equipo de desarrollo comenta en que avanzó el día anterior, en que avanzará el día actual y que inconvenientes tiene para desarrollar sus tareas; esta reunión se debe llevar a cabo todos los días y a la misma hora.

El Scrum Master debe asegurarse de que se cumpla la regla de que solo los miembros del equipo estén presentes en dicha reunión

Sprint Planning Meeting o Reunión de Planificación del Sprint: en esta reunión se planifica el trabajo que se va a realizar durante el sprint con la finalidad de cumplir con los objetivos planteados.

Esta reunión tiene una duración máxima de 4 horas para un sprint de un mes. Para sprint más cortos, el evento es usualmente más corto. El responsable de llevar esta reunión a cabo es el Scrum Master y debe asegurarse que los asistentes entiendan su propósito.

Sprint Review o Revisión del Sprint: en esta reunión se revisa el incremento y, de ser necesario, se añaden o modifican requerimientos en el Product Backlog.

Durante la reunión se explica todo lo que se hizo para cumplir con el objetivo del sprint y se aceptan sugerencias que optimicen el valor entregado. Debe tener una duración máxima de 4 horas si el sprint es de un mes. El Scrum Master es el encargado de planificar el evento y debe asegurarse que todos los asistentes entiendan el propósito. El resultado de esta reunión será una lista de tareas revisada para el siguiente sprint y con posibles mejoras a implementarse.

Sprint Retrospective o Retrospectiva del Sprint: esta reunión se realiza después de la Reunión de Revisión del Sprint y antes de la Reunión de Planificación del Sprint. Su duración máxima es de tres horas y tiene como objetivo crear un plan de mejoras a implementarse en el siguiente sprint. El Scrum Master es el responsable de organizar esta reunión y explicar a todos los implicados el propósito de la misma.

Al finalizar esta reunión, el Scrum Team debe tener claras las mejoras que se van a implementar y cómo van a mejorar los aspectos en que identificaron algún problema.

1.4.5. Puntos de historia

Este método se utiliza para dimensionar y relacionar la complejidad de una historia de usuario respecto a otra.

El primer paso es seleccionar una historia de usuario para asignarle una complejidad nominal, de manera que sirva de referencia para catalogar al resto de historias de usuario, es decir tener una línea base para comparar entre ellas la complejidad.

Se puede emplear la actividad de planning poker para determinar el valor de los puntos y el tiempo de las historias de usuario, según el cuadro que se muestra a continuación.

Tamaño	Puntos	Tiempo (horas)
XS (Extra pequeño)	1	4-7
S (Pequeño)	2	8-15
M (Mediano)	3	16-23
L (Largo)	5	24-39

XL (Extra Largo)	8	40-48
------------------	---	-------

Tabla 1.1 Puntos de historias de usuario

1.4.6. Ciclo de vida de Scrum

El ciclo de vida de Scrum está compuesto por 6 etapas:

1. Inicio del proyecto
2. Planificación
3. Desarrollo del Sprint
4. Revisión del Sprint
5. Implementación
6. Cierre del proyecto

Cada una de estas etapas está formada por prácticas. A continuación, se describe cada etapa.

1. Inicio del proyecto

En esta etapa se definen cinco prácticas:

- a. **Definir la visión del proyecto:** en esta práctica se estructura la visión del proyecto, explicando las necesidades del mercado que se busca satisfacer, considerando factores importantes como el alcance, tiempo, presupuesto y calidad esperada por el patrocinador del proyecto.
- b. **Formar el equipo Scrum:** se eligen los miembros que formarán parte del equipo Scrum acorde a sus habilidades más destacadas.
- c. **Construir y priorizar el Product Backlog:** en esta etapa se construye el Product Backlog; los elementos que lo conforman son:
 1. Historias de usuario
 2. Errores
 3. Tareas

Para realizar la priorización del Product Backlog se toman en cuenta los siguientes factores:

1. Tiempo
 2. Esfuerzo
 3. Dependencias
 4. Opinión del equipo de desarrollo
- d. **Definir el cronograma de entregas:** en esta práctica el Product Owner en conjunto con los miembros del Scrum Team, determinarán la duración estimada de los distintos Sprints.

- e. **Definir la arquitectura del proyecto:** en esta práctica el Product Owner y Scrum Team establecerán el diseño técnico del producto, que puede irse creando de forma iterativa al desarrollar los componentes en los diferentes Sprints.

2. Planificación

En esta etapa se definen tres prácticas:

- a. **Describir historias de usuario:** en esta práctica se describen las historias de usuario y sus tareas; las historias de usuario permiten definir los requerimientos del sistema.

Las historias de usuario están compuestas por tres elementos:

- **Como:** describe el rol de la persona o grupo que solicita la funcionalidad o requerimiento.
 - **Quiero:** describe la necesidad o requerimiento del usuario en una frase corta.
 - **Para:** describe el beneficio esperado por el usuario una vez que se implemente el requerimiento.
- b. **Priorizar historias de usuario:** esta práctica se realiza en paralelo con la planificación, ya que mientras el equipo termina de desarrollar lo establecido para el último Sprint, el Product Owner realiza la priorización de las historias de usuario que pueden considerarse para el siguiente Sprint y que se pueden tomar como base para guiar el rumbo del proyecto.

Es importante que la priorización se realice antes de que se lleve a cabo la Reunión de Planificación del Sprint con el fin de optimizar tiempo. Los elementos que se deben tomar en cuenta para priorizar las historias de usuario son:

- El valor del negocio
 - La dependencia
- c. **Reunión de planificación del Sprint:** en esta práctica el Product Owner expone el objetivo que se debería alcanzar en el Sprint y los elementos del Product Backlog que, si se completan en el Sprint, lograrían alcanzar el objetivo del Sprint.

El Sprint está conformado por las historias de usuario que fueron priorizadas en la práctica anterior. Además de las historias de usuario priorizadas por el Product Owner, el Product Backlog incluye:

- a. El último incremento del producto.
- b. Fallos encontrados en el producto o incrementos de producto (registro de errores).

3. Desarrollo del Sprint

Esta etapa está conformada por una práctica:

- a. **Desarrollar los entregables:** el objetivo de esta práctica es el desarrollo de los incrementos del sprint, manuales o documentación relacionada, considerando siempre la definición de listo o terminado.

4. Revisión del Sprint

Esta etapa está compuesta por una práctica:

- a. **Reunión de Revisión del Sprint:** en esta práctica se realiza la demostración del incremento del producto al Product Owner y demás partes interesadas, con la finalidad de obtener aprobación sobre lo que se esperaba y lo que fue desarrollado por el equipo.

A continuación, se listan las actividades que se realizan durante la Reunión de Revisión del Sprint:

1. Presentar el incremento del producto
2. Validar el incremento de producto
3. Aprobar el incremento del producto
4. Actualizar el Product Backlog

5. Implementación

Esta etapa está compuesta por dos prácticas:

- a. **Planificación de la implementación:** el objetivo de esta práctica es preparar los elementos necesarios para realizar una implementación exitosa de los incrementos generados en los Sprint, disminuyendo los riesgos que puedan impactar de manera negativa las operaciones de los usuarios o clientes.

La cantidad de incrementos dependerá de lo acordado con el Product Owner o patrocinador del proyecto; sin embargo, por lo general dependen del valor de la organización.

- b. **Implementación de entregables:** esta práctica tiene como objetivo poner a disposición de los usuarios los incrementos finalizados y utilizables desarrollados anteriormente por el Scrum Team.

Cabe recalcar que esta práctica no es posible aplicarla en todos los tipos de proyectos, ni es obligatorio ejecutarla al finalizar cada Sprint, solo se aplica cuando exista un cambio luego de finalizar los Sprints del proyecto.

6. Cierre del proyecto

En esta práctica se dejará registro del cierre y motivo del cierre del proyecto; para este fin podría construirse un acta de cierre del proyecto. Los motivos para el cierre del proyecto pueden ser:

- a. **Cierre exitoso del proyecto:** se cumplió con todos los entregables programados dentro del presupuesto asignado y de acuerdo con los criterios de aceptación.

- b. Cierre parcial del proyecto o aplazamiento:** se cumplió parcialmente con los entregables programados, se está superando el presupuesto asignado o no se ha cumplido completamente con todos los criterios de aceptación.
- c. Cancelación del proyecto:** no se cumplió con ninguno de los entregables programados, se agotó el presupuesto asignado antes de tiempo, no se cumplió con ningún criterio de aceptación.

1.5. Proceso de Aseguramiento de la Calidad de Software

El software es una de las herramientas más utilizadas dentro de una organización. Entre sus principales objetivos están optimizar procesos y satisfacer las necesidades de los usuarios, por tal razón es necesario contar con ciertos criterios para garantizar su calidad.

Según Pressman, la calidad de software es “La concordancia con los requisitos funcionales y de rendimiento establecidos, con estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente, además de cumplir el propósito para el que fue creado”. [2]

La gestión de la calidad del proceso de desarrollo de software influye directamente en la calidad del producto final.

Es importante entender que para garantizar la calidad del producto final es necesario que el proceso se desarrolle en base a lo que se definió inicialmente. Es decir, se debe verificar y validar que cada una de las actividades del ciclo de vida de desarrollo se estén llevando a cabo de manera adecuada, acorde a lo planificado y cumpliendo con lo que se definió inicialmente.

Existen algunas áreas dentro del proceso de gestión de la calidad que son indispensables para garantizar la calidad del proceso de desarrollo y producto de software, estas áreas son: aseguramiento de la calidad del proceso y producto, verificación del proceso, validación del proceso y adicionalmente, revisiones y auditoria que se conocen como métodos de verificación[11].

El propósito del Aseguramiento de la Calidad es entregar a la administración una visibilidad del proceso utilizado y los productos que se desarrollaron mediante las acciones planificadas que aseguren la calidad de dichos procesos y productos.

Por tal razón, el proceso de SQA abarca otros procesos como revisar, auditar e informar a la administración del proyecto sobre el cumplimiento de la planificación realizada y de la aplicación de estándares y procedimientos establecidos tanto en el proceso de desarrollo como del producto final de software.

El proceso de desarrollo incluye todas las actividades que se deben llevar a cabo para desarrollar o mantener el sistema de software. El producto comprende el software y todos los artefactos creados como parte de la definición, mantenimiento y uso del proceso de software, incluyendo especificaciones, descripción del proceso, planes, procedimientos, código y documentación.

Analizando lo anterior, los objetivos principales de SQA son:

- Planificar las actividades del proceso de SQA.
- Verificar y validar que el proceso de desarrollo y producto de software estén alineados a las actividades, estándares, procedimientos y requerimientos establecidos.
- Informar a los individuos afectados sobre las actividades de SQA y sus resultados.
- Comunicar al administrador del proyecto las desviaciones no resueltas dentro del proceso de desarrollo.

Para llevar a cabo estos objetivos es necesario que exista un responsable de SQA que se encargue de planificar las actividades del proceso de SQA, tareas a lo largo del ciclo de vida de un proyecto y la relación con el grupo encargado de desarrollar el software.

El aseguramiento de la calidad debe trabajarse desde etapas tempranas, para establecer planes, estándares y procedimientos que proporcionen valor agregado al proyecto y satisfagan los requerimientos y las políticas organizacionales. La elaboración de un plan, definición de procedimientos y estándares permite facilitar revisiones de cumplimiento y posteriores auditorías.

El principal rol del encargado de SQA es guiar al equipo de desarrollo a obtener un producto de calidad. Pero la implantación de la calidad es responsabilidad del grupo de desarrollo y de la administración. Es más, "la existencia de un proceso de SQA no garantiza que los procesos sean seguidos y que la calidad aparezca mágicamente en el producto. Debe existir un compromiso por parte de toda la organización por orientarse hacia una cultura de calidad." [12]

2. METODOLOGÍA BASADA EN SCRUM Y UML PARA EL DESARROLLO DE SISTEMAS DE SOFTWARE Y PROCESO DE SQA PARA EL AREA DE DESARROLLO DE EMASEO-EP

2.1. Descripción del Proceso actual de Desarrollo de software en EMASEO EP

El Área de Desarrollo de la Empresa Pública Metropolitana de Aseo se encarga de desarrollar software en la institución. Se encuentra conformada por un grupo pequeño de personas que se encargan de atender las necesidades de las diferentes áreas de la empresa. El proceso actual que se sigue para desarrollar software es el siguiente:

1. El usuario o área interesada en solicitar el desarrollo de un sistema de software, envía un correo a la Subdirección de TIC's en donde solicita el sistema y justifica el motivo de dicha necesidad.
2. Se analiza entre la Subdirección de TIC's y el Área de Desarrollo si es factible desarrollar el sistema solicitado y se procede a dar respuesta al usuario o área sobre la decisión tomada.
3. Si se encontró necesario desarrollar el sistema solicitado, se reúnen nuevamente la Subdirección de TIC's y el Área de desarrollo y se analiza el objetivo y alcance del proyecto. También se asignan responsables para el desarrollo del mismo.
4. Se define una nueva reunión entre el usuario que solicitó el sistema y el o los desarrolladores encargados y se definen los principales requerimientos.
5. Se reúnen el jefe del Área de Desarrollo y los desarrolladores responsables de implementar el sistema y se realiza la planificación de actividades y tiempos de entrega.
6. Se desarrolla el sistema acorde a los requerimientos definidos.
7. Se entrega el sistema de software al usuario o área y se trabaja constantemente en la estabilización y mejoras del sistema en etapa postproducción.

Respecto a los tipos de sistemas que se desarrollan dentro del área, en su mayoría corresponden a sistemas web y muy pocos de escritorio. En el desarrollo de estos sistemas se estima un tiempo entre dos y seis meses aproximadamente, dependiendo de la complejidad y cantidad de requerimientos que estos tengan.

Para los desarrolladores, uno de los factores que define la complejidad de un sistema es la cantidad de cálculos matemáticos que se deben aplicar para obtener resultados exactos, por ejemplo, los sistemas que se desarrollan para el Área de Finanzas o Talento Humano suelen requerir de cálculos complejos y exactos, por lo cual se los clasifica como sistemas complejos. También existen sistemas que deben desarrollarse con el fin de poder

integrarlos con sistemas existentes o involucrando a más de dos áreas dentro de la empresa. La cantidad de información que un sistema debe administrar también puede considerarse un factor importante para determinar su complejidad. En sistemas que permiten la gestión de una gran cantidad de información, la complejidad es más elevada que en sistemas en donde se administra un menor volumen de información.

Por otro lado, los sistemas de menor complejidad pueden definirse por tener pocos requerimientos, que en su mayoría están relacionados con la gestión de la información (donde se usan las operaciones básicas de manejo de datos en bases de datos relacionales, denominadas Operaciones CRUD: Create, Read, Update, Delete) o reportes.

Para el desarrollo de las aplicaciones se utilizan herramientas como Eclipse, Ireports, Formularios ASPX, Visual Studio 2015, Deven Express, Ajax ToolKit, ASP .Net, entre otras.

Para el ambiente de pruebas se utilizan herramientas open source compatibles con JEE6 y las bases de datos se encuentran alojadas en servidores como Red Hat y Windows Server 2003. El motor para base de datos de pruebas y producción es Oracle 12.2.

Dentro de los lenguajes de programación que se utilizan para el back-end de aplicaciones están C#, JavaScript y Java, para el front-end se utiliza el lenguaje de etiquetas HTML.

2.2. Justificación del uso de Scrum para la elaboración de la metodología de desarrollo

En el Área de Desarrollo de EMASEO EP los integrantes trabajan individualmente, es decir, cada uno desarrolla un proyecto distinto. A pesar de que existe una buena relación y colaboración dentro del área, no se trabaja en equipo, lo que conlleva al retraso en la entrega del producto final, ya que se sobrecarga de trabajo a los integrantes del equipo. Scrum nos permite solucionar este inconveniente, ya que en esta metodología se fomenta el trabajo en equipo y la buena comunicación, de manera que se pueda aprovechar al máximo las capacidades de cada integrante y mejorar el tiempo de entrega de los proyectos.

Actualmente, la mayoría de los proyectos que se desarrollan dentro del área deben entregarse en cortos plazos de tiempo y sus requerimientos suelen cambiar con frecuencia, ya que inicialmente no se definen correctamente o surgen nuevas necesidades durante el desarrollo. Además, el cliente solo conoce el resultado obtenido cuando se entrega el producto final, por lo que sería adecuado aplicar el enfoque iterativo e incremental de la metodología Scrum. Al aplicar este enfoque, beneficiaríamos tanto al cliente como al equipo de desarrollo, ya que, por un lado, el cliente tendría entregas continuas de valor del proyecto que solicitó y, por otro lado, el equipo de desarrollo no tendría que cambiar todo lo que tenía desarrollado en la etapa final de entrega del proyecto, sino cambiaría acorde a lo que el cliente solicite en cada sprint, de manera que los cambios no sean tan grandes e impliquen demasiado tiempo y esfuerzo.

Como dentro del área se desarrollan proyectos de pequeña y mediana complejidad, utilizar Scrum resulta ideal, ya que este proceso es ideal para proyectos de dicha magnitud.

En el proceso actual el cliente no interactúa más de 3 veces con el equipo de desarrollo, por lo que no se llega a conocer todas sus necesidades. Mediante la aplicación de Scrum el cliente obtiene múltiples beneficios al volverse parte del proceso de desarrollo, ya que de esta manera establece claramente sus expectativas, indica el valor que le aporta cada requerimiento al proyecto y cuando espera que esté desarrollado. De esta forma, el cliente comprobará de manera regular si se cumplen sus expectativas y, podrá dar una retroalimentación al equipo en cada iteración para tomar decisiones acertadas, con la finalidad de obtener un producto de calidad sin gastar demasiados recursos.

Otro problema que suele presentarse con frecuencia es que los integrantes del equipo de desarrollo están encargados de otras tareas aparte del desarrollo de software, por ejemplo, tienen que dar mantenimiento a sistemas antiguos o resolver pequeñas tareas asignadas, por lo cual tienen menos tiempo para dedicarse al desarrollo de nuevos proyectos. En Scrum la comunicación diaria entre los miembros del equipo a través de la reunión Daily Scrum podría ayudar al Scrum Master a ver el estado actual de las tareas y la dificultad que tienen para desarrollarlas, con la finalidad de encontrar una solución entre todos, por ejemplo, planificando adecuadamente la jornada de trabajo y repartición de tareas, de manera que se pueda avanzar en el desarrollo del proyecto.

Por todo lo anterior, se justifica el uso de Scrum como la base para definir la metodología de desarrollo en EMASEO-EP.

2.3. Propuesta de metodología de Desarrollo basada en Scrum para EMASEO EP

En base a lo visto en la sección 1.2 (Método y Metodología), una metodología de desarrollo consta de un proceso más un lenguaje de modelamiento. Para este proyecto, el proceso de desarrollo escogido será el definido por SCRUM, mientras que el lenguaje de modelamiento a usar será UML.

El proceso de desarrollo de software según Scrum está dividido en seis etapas: *inicio del proyecto, planificación, desarrollo del sprint, revisión del sprint, implementación y cierre del proyecto.*

En cada una de estas etapas se agregarán las características de UML que se crean necesarias para reforzar el proceso ágil seleccionado y conformar una metodología de desarrollo que se adapte a las necesidades del Área de Desarrollo de EMASEO EP.

2.3.1. Inicio del proyecto

a) Elaboración del documento Visión

Para empezar con la ejecución de un proyecto se llenará un documento en donde se detallará información general del proyecto a implementar, lo cual permitirá comprender de mejor manera el alcance del mismo.

Este documento se llamará **Visión del Proyecto**[13] y tendrá el siguiente contenido:

1. Introducción
2. Posicionamiento

3. Descripción de Stakeholders y usuarios
4. Descripción de Roles Scrum
5. Descripción global del producto
6. Restricciones
7. Precedencia y prioridad
8. Otros requisitos del producto
9. Requisitos de documentación
10. Atributos de Características

b) Formación del equipo SCRUM

Luego de llenar el documento de Visión del proyecto se formará el equipo Scrum acorde a las habilidades, competencias y carga horaria disponible de los miembros del equipo de desarrollo del Área de Desarrollo y de las partes interesadas en el proyecto dentro de la empresa EMASEO EP.

Los miembros primordiales para formar el equipo Scrum son: Product Owner, Scrum Master y Scrum Team (Recurso Humano que implementará el sistema).

En el documento de **Visión del Proyecto**, numeral 4. **Roles de Scrum**, se agregará la Tabla 2.1 con información de los miembros que participarán en el desarrollo del sistema de software.

Nombre de la Persona	Rol en Scrum	Principales Responsabilidades	Grado de participación en el proyecto

Tabla 2.1 Roles de Scrum

c) Construcción y priorización del Product Backlog

En esta práctica se detallarán los requerimientos funcionales del proyecto, descritos de manera general como tareas, que en conjunto forman el **Product Backlog** el formato a utilizar se muestra en la Tabla 2.2.

ID	Tarea	Prioridad	Tamaño (complejidad)	Esfuerzo (puntos)	Tiempo (días)
Tiempo y esfuerzo totales para la implementación de tareas					

Tabla 2.2 Product Backlog

A continuación, se explica cada columna:

- **ID:** es el identificador que se le asignará a la tarea.
- **Tarea:** es la descripción de la tarea expresada de manera general.
- **Prioridad:** prioridad de la tarea en el negocio. Los valores aceptados son numéricos.
- **Tamaño:** determina la complejidad de la implementación de la tarea; los valores aceptados son: XS, S, M, L, XL.
- **Esfuerzo:** valor en puntos de esfuerzo en función del tamaño de la tarea; los valores aceptados son: 1, 2, 3, 5, 8.
- **Tiempo (días):** tiempo en días para implementar la tarea en función de su tamaño; los valores aceptados son: ½, 1, 2, 3, 5.
- **Tiempo y esfuerzo totales para la implementación de tareas:** sumatoria del tiempo y esfuerzo de implementación de cada tarea.

Además, con la finalidad de realizar las pruebas de aceptación se definirán los **Criterios de Aceptación** el formato a utilizar se muestra en la Tabla 2.3.

ID	Tarea	Nro escenario	Criterio de aceptación	Contexto	Evento	Resultado

Tabla 2.3 Criterios de aceptación

A continuación, se explica cada columna:

- **ID:** es el identificador que se le asignará a la tarea.
- **Tarea:** es la descripción de la tarea expresada de manera general.
- **Nro escenario:** número que identifica el escenario asociado a la tarea.
- **Criterio de aceptación:** descripción del criterio de aceptación.
- **Contexto:** contexto del escenario que define un comportamiento.
- **Evento:** representación de la acción que el usuario ejecuta, en la descripción definida para el escenario.
- **Resultado:** dado el contexto y la acción ejecutada por el usuario, el resultado es el comportamiento del sistema en esa situación.

d) Elaboración de Diagramas de Casos de Uso

Para comprender de mejor manera los requerimientos funcionales del sistema, se modelarán los requerimientos mediante diagramas de casos de uso, que se elaborarán con la herramienta StartUML, en su versión free.

e) Elaboración del cronograma del proyecto

Se ordenan las tareas por prioridad desde alta a baja, y se procede a marcar el número de días (plazo) de implementación de cada tarea en el cronograma, el formato a utilizar se muestra en la Tabla 2.4 Cronograma del proyecto.

Actividad o Tarea	Duración en horas	Duración en días	Fecha de inicio	Fecha de fin	Entregable

Tabla 2.4 Cronograma del proyecto

La tabla tiene las siguientes columnas

- **Actividad o Tarea:** descripción de la actividad o tarea a llevar a cabo.
 - **Duración en horas:** tiempo que tomará implementar la actividad o tarea en horas.
 - **Duración en días:** tiempo que tomará implementar la actividad o tarea en días.
- Ojo:** Considerar que 8 horas equivalen a un día en la jornada laboral de la EMASEO EP.
- **Fecha de inicio:** fecha en la que iniciará la implementación de la actividad o tarea.
 - **Fecha de fin:** fecha en la que terminará la implementación de la actividad o tarea.
 - **Entregable:** nombre del documento, código fuente, manuales, entre otros, a entregar al finalizar la actividad.

f) Definir la arquitectura del proyecto

En esta práctica se debe definir la arquitectura con la que se trabajará en el proyecto. La arquitectura por elegir queda a criterio del Scrum Team considerando los factores de tecnología a utilizar y tipo de sistema a desarrollar.

2.3.2. Planificación

a) Descripción de historias de usuario

Una vez definidas las tareas a implementar en el Product Backlog, el siguiente paso es describirlas como historias de usuario.

En la Tabla 2.5 se muestra el formato a utilizar para la descripción de historias de usuario.

Tarea	Historia de usuario			Descripción
	Como	quiero	para	

Tabla 2.5 Descripción de historias de usuario

La Tabla 2.5 contiene las siguientes columnas:

- **Tarea:** es la descripción de la tarea expresada de manera general.
- **Como:** rol dentro del sistema del beneficiario de la implementación de la historia de usuario.
- **Quiero:** descripción de la característica que apoya al beneficio que obtendrá el usuario.
- **Para:** beneficio que se obtiene con la implementación de la historia de usuario.
- **Descripción:** descripción de la historia de usuario de manera más detallada.

b) Priorización de historias de usuario

Después de realizar la descripción de las historias se procede a realizar la respectiva priorización en base al valor de la historia de usuario para el negocio y su dependencia entre ellas.

En la Tabla 2.6 se muestra el formato a utilizar para realizar la priorización de historias de usuario.

ID	Prioridad

Tabla 2.6 Priorización de historias de usuario

La Tabla 2.6 tiene las siguientes columnas:

- **ID:** identificador de la historia de usuario.
- **Prioridad:** prioridad de la historia de usuario en el sprint.

Se determinará la prioridad de la historia de usuario en base a la Tabla 2.7 Nivel de dependencia de las Historias de usuario.

Prioridad	Nivel de dependencia	Puntuación
Muy alta	No depende de otras HU, pero otras HU dependen de su implementación.	1

Alta	Dependen de la implementación de otra HU y otras HU dependen de su implementación.	2
Media	Depende de la implementación de otras HU, pero no existen otras HU que dependan de su implementación.	3
Baja	No depende de la implementación de ninguna HU y ninguna otra HU depende de su implementación.	4

Tabla 2.7 Nivel de dependencia HU

c) Reunión de Planificación del Sprint

En esta reunión se planificarán las tareas (Sprint Backlog) necesarias que se llevarán a cabo para implementar las historias de usuario asignadas al sprint que se va a ejecutar.

Para realizar el Sprint Backlog se utilizará la herramienta Trello, en donde se podrá evaluar el progreso de la implementación de las historias de usuario durante el desarrollo del sprint.

2.3.3. Desarrollo del Sprint

a) Definición de terminado

El Scrum Team, Scrum Master y Product Owner, en la reunión de Planificación del Sprint, definirán el concepto de terminado, de manera que todas las partes interesadas lleguen a un acuerdo común.

b) Desarrollo de entregables

En esta etapa se especifican las historias de usuario que se van a implementar en el sprint y sus respectivas actividades. Además, al finalizar el sprint se evalúa si cada una de las actividades se ha completado o, por el contrario, si no se han completado, se implementarán en el siguiente sprint.

Para conocer el estado de las actividades al final del sprint se trabajará con la herramienta Trello, en donde se llevará el seguimiento de las mismas durante todo el sprint. Las tareas completadas se moverán a la columna **Completado** y las que no se han completado a la columna **Por implementar**.

2.3.4. Revisión del Sprint

a) Reunión de Revisión del Sprint

En esta etapa se presentará el incremento del producto y se comprobará que cada historia de usuario cumpla con los criterios de aceptación definidos. Si la historia de usuario cumple con los criterios de aceptación se aprobará, caso contrario, se implementará en el siguiente sprint.

En la Tabla 2.8, Revisión del Sprint, se muestran los campos que se deberán llenar una vez implementadas cada una de las historias de usuario asignadas al sprint.

ID Historia de usuario	
Evidencias del incremento:	
Cumple criterios de aceptación	
Estado	
Observaciones	

Tabla 2.8 Revisión del Sprint

A continuación, se describen los campos:

- **ID Historia de usuario:** identificador de la historia de usuario.
- **Incremento:** resultado de la historia de usuario implementada.
- **Cumple criterios de aceptación:** si la historia de usuario cumple con los criterios de aceptación, colocar SI, caso contrario colocar la palabra NO y el criterio de aceptación que incumple.
- **Estado:** estado de la historia de usuario después de la reunión de revisión del sprint. Los valores pueden ser: **Aprobado** o **por implementar siguiente sprint**.
- **Observaciones:** detallar las observaciones encontradas en la implementación de la historia de usuario.

Al final de cada revisión del sprint se detallarán los siguientes datos:

- **Total de historias de usuario por implementar:** número total de historias de usuario planificadas para implementar en el sprint.
- **Total de historias de usuario implementadas y aprobadas:** número de historias de usuario implementadas y aprobadas exitosamente en el sprint.
- **Total de historias de usuario pendientes:** número de historias de usuario que no se implementaron y que se deben implementar en el próximo sprint.

2.3.5. Implementación

a) Planificación de la implementación

La implementación de las historias de usuario definidas para cada sprint se llevará a cabo siguiendo los estándares de código fuente y base de datos, también se representará la estructura del sistema mediante un diagrama de clases.

Estándar de codificación

La codificación sistemas de software en el Área de Desarrollo de EMASEO EP se lleva a cabo utilizando los siguientes lenguajes de programación para back-end: Java, JavaScript, y C#. Con la finalidad de seguir buenas prácticas y llevar un código más legible y ordenado se utiliza el estándar lower camelcase.

Estándar de base de datos

Para el acceso a datos a los sistemas de software desarrollados en el Área de Desarrollo de EMASEO EP, se utiliza el gestor de base de datos Oracle, con la finalidad de seguir buenas prácticas se utilizará un estándar basado en nombres y siglas para objetos de base de datos propios de la empresa.

Diagrama de clases

Se implementará un **Diagrama de clases** para tener un mejor entendimiento de la base de datos, en donde se representará la estructura del sistema mediante el modelamiento de sus clases, atributos y relaciones. Este diagrama se modelará en la herramienta StartUML en su versión gratuita.

b) Planificación de implementación de entregables

El proceso de implementación de entregables solamente se llevará a cabo si existen requerimientos que no fueron implementados dentro del plazo de tiempo previsto o si no se cumplieron con los criterios de aceptación definidos inicialmente. Se planificará la implementación de estos requerimientos en el siguiente sprint, asignándoles una prioridad alta.

2.3.6. Cierre del proyecto

En esta última etapa del proyecto se llenará el **Acta de Cierre del Proyecto[14]**, en donde se especificarán los motivos de la culminación del proyecto, ya sea que se haya cerrado exitosamente, parcialmente o se suspenda por alguna razón de fuerza mayor.

Además, se describirán los entregables aprobados durante el proceso de implementación del proyecto. Para constancia de ello, deberán firmar todos los involucrados, es decir, Product Owner, Scrum Master, Scrum Team y demás partes interesadas en caso de existir.

2.4. Propuesta del proceso de aseguramiento de la calidad

El aseguramiento de la calidad (Quality Assurance - QA) es un proceso que se debe llevar a cabo a lo largo de todo el proyecto de desarrollo de software. Los beneficios que se obtendrán al aplicar este proceso son múltiples, entre ellos:

- Obtener un producto que satisfaga las necesidades del cliente y, por ende, genere confianza en los productos desarrollados por el equipo de desarrollo.
- Facilitar las tareas de mantenimiento a corto y largo plazo.
- Detectar errores en etapas tempranas y evitar costos adicionales para su eliminación a largo plazo.

Para este proyecto, el proceso de QA se aplicará durante todo el proceso de desarrollo de software.

A continuación, se detallarán los criterios y métricas a considerar para evaluar la calidad de los entregables en cada etapa de Scrum:

2.4.1. Evaluación de historias de usuario (Product Backlog)

La adecuada definición de historias de usuario es un punto clave en el desarrollo de un sistema software.

Una historia de usuario correctamente definida deberá describir la tarea que el cliente o usuario desea realizar y su finalidad, para alcanzar el nivel de detalle adecuado; las historias de usuario deberán cumplir el modelo INVEST, desarrollado por Bill Wake[15].

Este modelo permite garantizar que las historias de usuario ofrecen un valor al negocio, aunque se desarrollen en una sola iteración. Las siglas INVEST hacen referencia a que las historias de usuario deberán cumplir con las siguientes características:

- **I-Independent (Independiente):** cada historia de usuario podrá ser planificada e implementada en cualquier orden. No dependen unas de otras, si esto ocurre, se deben dividir o combinar.
- **N-Negotiable (Negociable):** las historias de usuario deben ser negociables ya que sus detalles deberán ser acordados por el cliente o usuario y el equipo de desarrollo durante la fase de acuerdo.
- **V-Valuable(Valorable):** la historia de usuario debe agregar un valor para el cliente.
- **E-Estimable (Estimable):** una buena historia de usuario debe ser estimada con la precisión suficiente para ayudar al cliente o usuario a priorizar y planificar su implementación.
- **S- Small (Pequeña):** debe ser lo más pequeña posible de tal modo que ocupe como máximo un sprint.

- **T-Testable (Comprobable):** la historia de usuario debe poderse probar. Tanto el usuario como el equipo de desarrollo deberán probarla para saber cuándo está finalizada.

Evaluadores

La evaluación deberá ser realizada por el Departamento de Aseguramiento de la Calidad o persona encargada de este proceso, en caso de que la empresa cuente con los recursos mencionados. Caso contrario, la evaluación se realizará por parte del Scrum Master, quien representará al Scrum Team y Product Owner, que representará a los usuarios y stakeholders, ya que se necesita tener dos puntos de vista para mantener la objetividad y balancear los resultados en caso de subjetividad.

Escala de evaluación de calidad

La escala de evaluación a considerar para las historias de usuario es la siguiente:

Nro. de criterios INVEST que cumple la historia de usuario	Nivel de calidad
1-2	Baja
3-4	Media
5	Alta

Tabla 2.9 Escala de evaluación de Historias de usuario

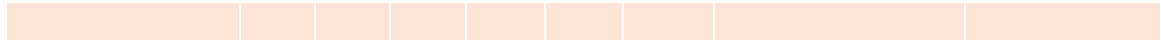
La cantidad de criterios INVEST que cumple la historia de usuario permitirán estimar su calidad, la que será evaluada por los evaluadores correspondientes.

Así mismo, se podrá evaluar la calidad del conjunto de historias de usuario definidas para cada sprint haciendo promedio de la sumatoria del nivel de calidad de todas las historias de usuario para el total de historias de usuario asignadas al sprint.

Formato de evaluación

La Tabla 2.10 permitirá evaluar si las historias de usuario cumplen con el modelo mencionado anteriormente.

Historia de usuario	Independiente	Negociable	Valorable	Estimable	Pequeña	Testeable	Nro criterios que cumple la historia de usuario	Nivel de calidad



A continuación, se describirá *Tabla 2.10 Formato de evaluación de Historias de Usuario* cada columna:

- **Historia de usuario:** descripción de la historia de usuario en términos de COMO, QUIERO y PARA.
- Marcar con una **X** sobre el o los criterios que cumple la historia de usuario.
- **Nro criterios que cumple:** número de criterios INVEST que cumple la historia de usuario según el evaluador.
- **Nivel de calidad:** nivel de calidad de la historia de usuario acorde al número de criterios que esta cumple. Los valores pueden ser: Baja, Media, Alta.

2.4.2. Evaluación de actividades (Sprint Backlog)

El conjunto de actividades que se deben llevar a cabo para implementar las historias de usuario asignadas a un sprint están descritos en el Sprint Backlog; para evaluarlo y garantizar su calidad, se utilizará el método **SMART**, que consiste en evaluar si las actividades cumplen con las siguientes características:

- **S-Specific (Específica):** una actividad debe ser lo suficientemente específica para que todos puedan entenderla.
- **M-Measurable (Medible):** ¿Hace lo que pretende? Es el equipo quien debe ponerse de acuerdo sobre lo que esto significa.
- **A-Achievable (Alcanzable):** ¿Es razonable la meta? Es bueno establecer metas que representen un desafío, pero puede ser contraproducente no alcanzarlas.
- **R-Relevant (Relevante):** cada actividad debe ser relevante, contribuyendo a la historia en cuestión.
- **T-Time Boxed (Encasillada):** una actividad debe estar limitada a una duración específica. No necesariamente debe ser una estimación formal en horas o días, pero debe haber una expectativa para que los desarrolladores sepan cuando deben buscar ayuda.

Evaluadores

La evaluación deberá ser realizada por el Departamento de Aseguramiento de la Calidad o persona encargada de este proceso, en caso de que la empresa cuente con los recursos mencionados. Caso contrario, la evaluación se realizará por parte del Scrum Master, quien evaluará si las actividades creadas por el Scrum Team cumplen con los criterios a evaluar según el método SMART.

Escala de evaluación

La escala de evaluación a considerar para las actividades definidas en el Sprint Backlog consta en la Tabla 2.11.

Nro. de criterios SMART que cumple la actividad	Nivel de calidad
1-2	Baja
3-4	Media
5	Alta

Tabla 2.11 Escala de evaluación de Sprint Backlog

La cantidad de criterios SMART que cumple una actividad define la calidad de la misma y, será evaluada por los evaluadores mencionados anteriormente.

También se podrá conocer la calidad del Sprint Backlog o conjunto de actividades, haciendo la sumatoria del número de criterios que cumplieron cada una de ellas, dividido para el total de actividades definidas.

$$\sum_{\text{CRITERIOS CUMPLEN ACTIVIDADES S.B}} = \text{N}^{\circ} \text{ CRITERIOS ACTIV 1} + \text{N}^{\circ} \text{ CRITERIOS ACTIV 2} + \dots + \text{N}^{\circ} \text{ CRITERIOS ACTIV n}$$

$$\text{Nivel de calidad}_{S.B} = \sum_{\text{CRITERIOS CUMPLEN ACTIVIDADES S.B}} / \text{N}^{\circ} \text{ total de actividades}_{S.B}$$

(3.1)

Por ejemplo, si en el Sprint Backlog hay 10 actividades se sumará el número de criterios que cumplió cada una de estas actividades y se dividirá la sumatoria para 10.

Formato de evaluación

La Tabla 2.12 permitirá evaluar si las actividades cumplen con las características del método SMART.

ID historia de usuario	Actividades	Específico	Medible	Alcanzable	Relevante	A tiempo	Nro. criterios cumple	Nivel de calidad
ID-001								
ID-002								

Tabla 2.12 Criterios de evaluación de tareas

A continuación, se describirá cada columna:

- **ID historia de usuario:** identificador de la historia de usuario.
- **Actividades:** actividades que se deberán llevar a cabo para implementar la historia de usuario.
- Marcar con una **X** sobre el o los criterios que cumple la actividad.
- **Nro. criterios cumple:** número de criterios SMART con los que cumple la actividad según el criterio del evaluador.
- **Nivel de calidad:** nivel de calidad de la actividad acorde al número de criterios con los que cumple.

2.4.3. Evaluación de criterios de aceptación

Los criterios de aceptación son una serie de preceptos que validan la implementación de las historias de usuario y ayudan a entender el producto de software, por lo cual es esencial que se escriban correctamente. Un criterio de aceptación de calidad deberá cumplir con las siguientes características [16]:

- **Atomicidad:** debe tener dos resultados únicamente: ÉXITO o FRACASO.
- **No ambiguo:** debe ser interpretado de una única manera por n personas.
- **Verificable:** debe estar escrito de forma que el cliente o usuario lo pueda verificar rápidamente.
- **Completo:** el grupo de criterios de aceptación debe incluir todos los requisitos funcionales.

Evaluadores

La evaluación deberá ser realizada por el Departamento de Aseguramiento de la Calidad o la persona encargada de este proceso en la empresa. En caso de no implementarse todavía los recursos mencionados anteriormente, la evaluación la podrá realizar el Scrum Master, ya que es la persona que se encarga de supervisar que el Scrum Team trabaje aplicando buenas prácticas de manera que se puedan obtener criterios de aceptación de calidad.

Escala de evaluación

La escala de evaluación a considerar para los criterios de aceptación está descrita en la Tabla 2.13.

Nro. de características que cumple el criterio de aceptación	Nivel de calidad
--	------------------

1 - 2	Baja
3	Media
4	Alta

Tabla 2.13 Escala de evaluación para criterios de aceptación

La cantidad de características que cumple el criterio de aceptación define su nivel de calidad.

Para evaluar la calidad de los criterios de aceptación en conjunto se hará la sumatoria del número de características que cumplen cada uno de ellos y se dividirá para el total de los mismos.

$$\sum_{\text{CARACTERÍSTICAS C.A}} = \text{N}^{\circ} \text{ características C.A}_1 + \text{N}^{\circ} \text{ características C.A}_2 + \dots + \text{N}^{\circ} \text{ características C.A}_n$$

$$\text{Nivel calidad Total C.A} = \sum_{\text{CARACTERÍSTICAS C.A}} / \text{N}^{\circ} \text{ total de C.A}$$

(3.2)

Por ejemplo, si se tienen 20 criterios de aceptación, se hará la sumatoria del número de características que cumplen cada uno de ellos y a este resultado se lo dividirá para el número total de criterios de aceptación, es decir, 20.

Formato de evaluación

La Tabla 2.14 permitirá evaluar si los criterios de aceptación se escribieron correctamente.

ID historia de usuario	Criterio de aceptación	Atomidad	No ambiguo	Verificable	Completo	Nro. criterios cumple	Nivel de calidad
ID-001							
ID-002							

Tabla 2.14 Criterios de evaluación de Criterios de Aceptación

A continuación, se describirá cada columna:

- **ID historia de usuario:** identificador de la historia de usuario.
- **Criterio de aceptación:** título del criterio de aceptación.

NOTA: Marcar con una **X** sobre el o los criterios que cumple el criterio de aceptación.

- **Nro. criterios cumple:** número de criterios de calidad que cumplen los criterios de aceptación.
- **Nivel de calidad:** observaciones del criterio de aceptación respecto a la evaluación realizada.

2.4.4. Evaluación de cumplimiento de la metodología

Para verificar si la metodología propuesta se está cumpliendo, se realizará una evaluación en donde se verificará el cumplimiento de los principales criterios que la definen. Por ejemplo, cumplimiento de la reunión diaria, reunión de planificación, reunión de revisión del sprint, definición de historias de usuario, entre otros.

Se verificará el cumplimiento de los siguientes criterios:

ID	Descripción del criterio	No se aplico	Se aplico, pero no exitosamente	Se aplico exitosamente
001	Definición de historias de usuario	se calificará como NO SE APLICO cuando no se definan las historias de usuario.	se calificará como SE APLICO, PERO NO EXITOSAMENTE cuando se definan las historias de usuario, pero estas tengan un nivel de calidad Bajo.	se calificará como SE APLICO EXITOSAMENTE cuando se definan las historias de usuario y estas tengan un nivel de calidad Medio – Alto.
002	Establecimiento de prioridad de las historias de usuario	se calificará como NO SE APLICO , si no se hizo la priorización de historias de usuario en el sprint.		se calificará como SE APLICO EXITOSAMENTE si se realizó la priorización de historias de usuario en el sprint.
003	Elaboración de diagramas de casos de uso	se calificará como NO SE APLICO cuando no se desarrollen los diagramas de casos de uso de los requerimientos definidos.	se calificará como SE APLICO, PERO NO EXITOSAMENTE cuando se realicen los diagramas de caso de uso, pero no estén correctamente definidos.	se calificará como SE APLICO EXITOSAMENTE cuando se realicen los diagramas de caso de uso y estos sean claros y bien entendidos por todos los involucrados en el desarrollo del proyecto.
004	Cumplimiento de Reunión de planificación del sprint	se calificará como NO SE APLICO , si no se realiza la reunión de	se calificará como SE APLICO, PERO NO EXITOSAMENTE si se realiza la reunión de	se calificará como SE APLICO EXITOSAMENTE , si se realizó la reunión de

		planificación del sprint al comienzo del mismo.	planificación del sprint, pero no se llega a un acuerdo común entre las partes interesadas.	planificación del sprint y todos los involucrados se sienten satisfechos con el resultado de la reunión.
005	Definición de Sprint Backlog	se calificará como NO SE APLICO cuando no se defina el sprint backlog en el sprint.	se calificará como SE APLICO, PERO NO EXITOSAMENTE , cuando las actividades definidas en el sprint backlog tengan un nivel de calidad Bajo.	se calificará como SE APLICO EXITOSAMENTE cuando las actividades definidas en el sprint backlog tengan un nivel de calidad Medio – Alto.
006	Elaboración de diagramas de clases	se calificará como NO SE APLICO , si no se realizó el diagrama de clases.	se calificará como SE APLICO, PERO NO EXITOSAMENTE cuando se realicen los diagramas de clases, pero no estén correctamente definidos.	se calificará como SE APLICO EXITOSAMENTE cuando se realicen los diagramas de clases y sean entendidos por todos los involucrados en el proyecto.
007	Cumplimiento de Reuniones Diarias	se calificará como NO SE APLICO cuando no se hayan realizado las reuniones diarias o se haya realizado solo un 20% de las mismas en todo el sprint.	se calificará como SE APLICO, PERO NO EXITOSAMENTE cuando se hayan realizado hasta el 80 % de las reuniones diarias en el sprint o el límite de tiempo de 15 minutos se ha extendido.	se calificará como SE APLICO EXITOSAMENTE cuando se hayan realizado todos los días las reuniones diarias de máximo 15 minutos y se haya sacado provecho de las mismas.
008	Entrega de incremento	se calificará como NO SE APLICO cuando no se haya realizado la entrega del incremento al final del sprint al Product Owner.	se calificará como SE APLICO, PERO NO EXITOSAMENTE cuando se haya realizado la entrega del incremento al Product Owner pero la calidad del mismo sea Baja.	se calificará como SE APLICO EXITOSAMENTE cuando se haya entregado el incremento al Product Owner y la calidad del mismo sea Media o Alta.

009	Cumplimiento de Reunión de revisión del sprint	se calificará como NO SE APLICO cuando no se haya realizado la reunión de revisión del sprint.	se calificará como SE APLICO, PERO NO EXITOSAMENTE cuando se haya realizado la reunión de revisión del sprint, pero no se haya llegado a un acuerdo común de resolución del sprint entre las partes involucradas en el proyecto.	se calificará como SE APLICO EXITOSAMENTE cuando se haya realizado la reunión de revisión del sprint y todas las partes involucradas se den por satisfechas.
-----	--	---	---	---

Tabla 2.15 Criterios de cumplimiento de metodología

Evaluadores

La evaluación deberá ser realizada por el Departamento de Aseguramiento de la Calidad o persona encargada de este proceso, en caso de que la empresa cuente con los recursos mencionados. Caso contrario, la evaluación se realizará por parte del Scrum Master y Product Owner, quienes representaran al Scrum Team y, usuarios y stakeholders respectivamente, ya que se necesita tener dos puntos de vista para mantener la objetividad y balancear los resultados en caso de subjetividad.

Escala de evaluación

La escala de evaluación a considerar para evaluar el cumplimiento de los criterios que definen la metodología propuesta se describe en la Tabla 2.16.

Puntuación	Significado	Nivel de calidad
1 - 1.9	No se aplico	Bajo
2 - 2.5	Se aplico, pero no exitosamente	Medio
2.5 - 3	Se aplico exitosamente	Alto

Tabla 2.16 Escala de evaluación del cumplimiento de la metodología

Formato de evaluación

La Tabla 2.17 permitirá evaluar si se implementó la metodología adecuadamente en el sprint.

CRITERIOS DE EVALUACION DE CUMPLIMIENTO DE METODOLOGIA	PUNTUACIÓN EVALUADOR 1	PUNTUACIÓN EVALUADOR 2
CRI-001 Definición de historias de usuario		
CRI-002 Establecer prioridad a historias de usuario		
CRI-003 Elaboración de diagramas de casos de uso		

CRI-004 Cumplimiento de reunión de planificación		
CRI-005 Definición de Sprint Backlog		
CRI-006 Elaboración de diagramas de clases		
CRI-007 Reuniones diarias		
CRI-008 Entrega de incremento		
CRI-009 Cumplimiento de reunión de revisión		
Sumatoria total de la puntuación criterios		
Promedio entre la sumatoria de la puntuación de los evaluadores		
Nivel de calidad obtenido acorde al promedio entre la sumatoria de los puntajes de los evaluadores		

Tabla 2.17 Formato de evaluación de cumplimiento de la metodología

2.4.5. Evaluación del nivel de compromiso de las partes interesadas

Uno de los puntos más importantes a tener en cuenta al aplicar el proceso Scrum es la relación entre las partes interesadas y la predisposición que tengan para trabajar con la metodología propuesta. Por ende, es importante evaluar el desempeño de las personas involucradas en el desarrollo del proyecto, no con la finalidad de juzgarlos, sino que a partir de los errores cometidos se pueda mejorar las técnicas aplicadas y el trabajo en equipo, con la finalidad de crear un equipo autoorganizado e independiente. Esta evaluación se podría hacer en los dos primeros y dos últimos sprints para comparar el nivel de evolución de los integrantes del proyecto de desarrollo de software.

Los criterios a evaluar son:

- **Nivel de compromiso en el desarrollo del proyecto:** se trata del compromiso de las partes involucradas en el proyecto (Product Owner, Scrum Master y Scrum Team), por ejemplo, asistencia a las reuniones para definición de requerimientos, apertura al dialogo para llegar a acuerdos comunes, entre otros.
- **Nivel de conocimiento del negocio:** este criterio únicamente se evaluará al Producto Owner y Scrum Master, quienes representarán a los clientes o usuarios y al Scrum Team respectivamente. El Product Owner y Scrum Team deberán comprender el negocio para dar a conocer las necesidades de clientes y explicar estas necesidades a los miembros del Scrum Team, respectivamente.
- **Nivel de cumplimiento de responsabilidades asignadas:** se evaluará el nivel de cumplimiento de las responsabilidades asignadas. En este punto los evaluadores se repartirán de acuerdo a la Tabla 2.18.

Evaluado	Evaluador
Product Owner	Scrum Master
Scrum Master	Scrum Team

Scrum Team	Scrum Master y Product Owner
Scrum Master, Scrum Team y Product Owner	Departamento de SQA o Personal encargado de SQA (en caso de existir en la empresa)

Tabla 2.18 Evaluados y Evaluadores Cumplimiento de responsabilidades

- **Nivel de conocimiento técnico en las tecnologías a utilizar:** este criterio se evaluará al Scrum Team en base a las tecnologías asignadas a cada miembro del equipo. Los encargados de evaluar serán el Scrum Master y el miembro con más experiencia del Scrum Team, todo esto con la finalidad de evaluar el conocimiento al inicio y al final del proyecto.
- **Capacidad de trabajar en equipo:** se evaluará la capacidad de trabajar en equipo de todos los involucrados en el desarrollo del proyecto.

En este punto los evaluadores se repartirán de acuerdo con la Tabla 2.19.

Evaluado	Evaluador
Product Owner	Scrum Master
Scrum Master	Scrum Team
Scrum Team	Scrum Master y Product Owner
Scrum Master, Scrum Team y Product Owner	Departamento de SQA o Personal encargado de SQA (en caso de existir en la empresa)

Tabla 2.19 Evaluados y Evaluadores Capacidad de trabajar en equipo

Escala de evaluación

La escala de evaluación a considerar para evaluar el nivel de compromiso de las partes interesadas se describe en la Tabla 2.20.

Puntuación	Nivel de calidad
1-1.9	Regular
2-2.5	Bueno
2.5-3	Excelente

Tabla 2.20 Escala de Evaluación Nivel de compromiso partes interesadas

Además, se podrá evaluar el nivel de compromiso de todo el equipo involucrado en el desarrollo del proyecto, dividiendo el promedio de la sumatoria total de la puntuación de

los criterios por evaluar de todos los miembros del equipo para el número total de participantes en el proyecto.

Formulas aplicadas para la evaluación por cada miembro de Scrum

$$\sum_{\text{CRITERIOS A EVALUAR}} = \text{Criterio}_1 + \text{Criterio}_2 + \dots + \text{Criterio}_n$$

$$\text{Promedio}_{\text{CRIT A EVAL/TOT.CRITERIOS EVAL}} = \frac{\sum_{\text{CRIT A EVAL}}}{\text{Total}_{\text{CRIT.EVALUADOS}}}$$

(3.3)

Formulas aplicadas para la evaluación de criterios de todo el equipo Scrum

$$\sum_{\text{PROM C/MIEMBRO EQUIPO}} = \text{Promedio}_{\text{CRIT.EVAL MIEMBRO 1}} + \text{Promedio}_{\text{CRIT.EVAL MIEMBRO 2}} + \dots + \text{Promedio}_{\text{CRIT.EVAL MIEMBRO n}}$$

$$\text{Promedio}_{\frac{\sum_{\text{PROM C/MIEMBRO EQUIPO}}}{\text{TOT MIEMBROS EQUIPO}}} = \frac{\sum_{\text{PROM MIEMBRO 1}} + \sum_{\text{PROM MIEMBRO 2}} + \dots + \sum_{\text{PROM MIEMBRO n}}}{\text{Total}_{\text{MIEMBROS EQUIPO}}}$$

(3.4)

Formato de evaluación

La Tabla 2.21 permitirá evaluar el nivel de compromiso de las partes interesadas en el proyecto.

Criterios por evaluar	Product Owner	Scrum Master	Scrum Team		
			Miembro 1	Miembro 2	Miembro n
Nivel de compromiso en el desarrollo del proyecto (PO, SM y ST)					
Nivel de conocimiento del negocio (SM y PO)					
Nivel de cumplimiento de responsabilidades asignadas (PO, SM y ST)					
Nivel de conocimiento técnico en las tecnologías a utilizar (Miembros del ST)	N/A				
Capacidad de trabajar en equipo (PO, SM y ST)					
Suma total de la puntuación de cada criterio por evaluar					
Promedio de la suma anterior dividido para el numero de criterios por evaluar					

Tabla 2.21 Formato de evaluación del Nivel de compromiso de las partes interesadas

2.4.6. Evaluación del cumplimiento de estándares

Utilizar estándares de codificación y base de datos en el desarrollo de un sistema de software traen múltiples beneficios, entre los principales a destacar están:

- Facilidad de compresión del código fuente del sistema de software.
- Código fuente más ordenado, estructurado y limpio.

- Facilidad de mantenimiento o actualizaciones a corto, medio o largo plazo, en el sistema de software.
- La producción de código fuente es más eficiente y permite al equipo de desarrollo trabajar en paralelo, ya que todos comprenden lo que hacen al utilizar un estándar definido.

Por los múltiples beneficios que se obtienen al utilizar estándares de codificación y base de datos, es necesario evaluar que se estén aplicando los estándares definidos en el **Capítulo 2 literal 2.3.5 Etapa de Implementación**. Para esto se propone que se definan los Evaluadores, los Evaluados, la Escala de Evaluación y el Formato de Evaluación, de la siguiente manera:

Evaluadores

La evaluación deberá ser realizada por el Departamento de Aseguramiento de la Calidad o persona encargada de este proceso en la empresa. En caso de no implementarse todavía los recursos mencionados anteriormente, la evaluación la realizara el Scrum Master con la ayuda del miembro con más experiencia del Scrum Team, con la finalidad de evaluar si se están aplicando los estándares de codificación y base de datos especificados inicialmente.

Evaluados

Se evaluará la aplicación de los estándares definidos a los miembros del Scrum Team que codifiquen y estructuren la base de datos del sistema de software.

Escala de evaluación

En la Tabla 2.22 se describe la valoración para evaluar si se cumplen con los estándares especificados en el capítulo 2.

Valoración	Significado
Cumple	Cumple con la aplicación del estándar especificado en el trabajo desarrollado.
No cumple	No cumple con la aplicación del estándar especificado en el trabajo desarrollado.
No aplica o N/A	No aplica la evaluación sobre el miembro del Scrum Team porque en su trabajo no implica el uso del estándar especificado.

Tabla 2.22 Escala de evaluación Cumplimiento de Estándares

Formato de evaluación

La Tabla 2.23 permitirá evaluar el cumplimiento de estándares en el desarrollo del proyecto.

Estándar a evaluar	Miembros del Scrum Team		
	Miembro 1	Miembro 2	Miembro n
Codificación			
Base de datos			

Tabla 2.23 Formato de evaluación Cumplimiento de Estándares

A continuación, se explican los campos de la Tabla 2.23:

- **Estándar a evaluar:**
Codificación: indica que el estándar que se va a evaluar en esa fila es el de codificación.
Base de datos: indica que el estándar que se va a evaluar en esa fila es el de base de datos.
- **Miembros del Scrum Team:** en cada columna se colocará el nombre de cada miembro con la valoración correspondiente: Cumple, No cumple o No Aplica o N/A

2.4.7. Pruebas sugeridas

Esta etapa inicia con la definición del plan de pruebas, el que sirve como guía para la realización de pruebas y permite verificar que el sistema de software cumpla con los requerimientos definidos por el usuario, con las debidas garantías de calidad.

El plan de pruebas define los objetivos de las pruebas en un sistema, establece y coordina una estrategia de trabajo y provee un marco adecuado para elaborar una planificación paso a paso de las actividades de las pruebas[17].

Las pruebas ágiles o Agile Testing son técnicas esenciales en el desarrollo de sistemas de software que implementan los principios y prácticas de las metodologías ágiles como: iteraciones cortas, refactorización e integraciones continuas.

Las pruebas ágiles mejoran la agilidad porque permiten crear un producto de calidad, esta calidad promueve la adaptabilidad al cambio en un ambiente cambiante[18].

Características de las pruebas ágiles

Existen muchos principios ágiles que se relacionan con las pruebas ágiles; a continuación se muestran las principales características de agile testing[18]:

- Las pruebas forman parte del desarrollo, formando un todo integral, no es algo que esté separado.
- Mayor productividad y calidad del sistema de software al tener un equipo menos estresado y más feliz.
- Acelera la retroalimentación para poder realizar las adaptaciones necesarias lo más pronto posible y evitar grandes cambios en la entrega del producto final.

- Todo el equipo se involucra en las pruebas para fomentar el entendimiento global desde un enfoque centrado en el cliente, lo cual aumenta la creatividad en la propuesta de soluciones.
- Permite incorporar técnicas de pruebas para aumentar la agilidad y productividad.

Además, al usar Agile Testing se debe[19]:

- Asegurar que las funcionalidades entregadas se ajustan a los requisitos establecidos por el usuario.
- Identificar escenarios de ejecución no previstos por el usuario
- Asegurar que la aplicación presta servicio bajo parámetros aceptables de desempeño, que soporta carga y no colapsa.
- Asegurar que el código desarrollado es confiable, mantenible y puede crecer con la aplicación.

Matriz de pruebas ágiles

Las pruebas ágiles maximizan el valor de la solución y minimizan los riesgos, ya que resulta mucho mejor fallar y descubrir errores en las primeras etapas de desarrollo del proyecto, corregirlos y de esa manera tener un sistema estable y sin fallos al final del proyecto[18].

Con esto se considerarán los cuadrantes del Agile Testing definidos originalmente por Brian Marick y luego aceptados por Lista Crispin y Janet Gregory en su obra "Agile Testing"[19].

- Perspectiva tecnológica
- Perspectiva de negocio
- Soporte y guía de desarrollo
- Crítica al producto

Con estos cuadrantes se representa de la siguiente manera la matriz de pruebas ágiles[18]:

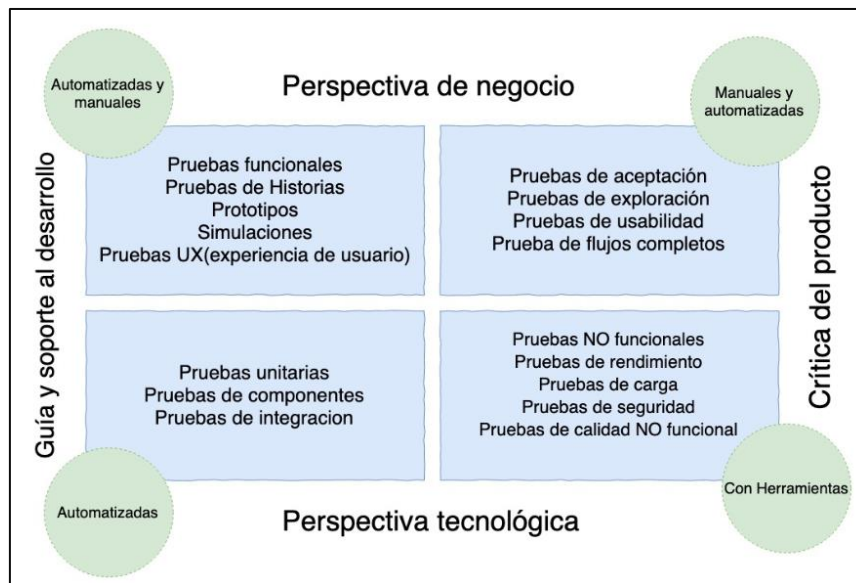


Figura 2.1. Cuadrantes de Agile Testing

Cuadrante 1: proporciona una guía y soporte al programador desde la perspectiva tecnológica, se crean pruebas unitarias y de componentes, las cuales se construyen antes y después de escribir código, esto nos indica que las partes más pequeñas funcionan como deberían, para esto se utiliza TDD (Test Driven Deveploment).

- **Pruebas unitarias:** tienen como objetivo verificar la funcionalidad y estructura de cada componente individualmente una vez que ha sido codificado[20].

Existen dos enfoques principales para el diseño de casos de prueba:

a) Pruebas de caja blanca: se verifica la estructura interna del componente con independencia de la funcionalidad establecida para el mismo. Por tanto, no se comprueba la corrección de los resultados si estos se producen.

b) Pruebas de caja negra: se comprueba el correcto funcionamiento de los componentes del sistema de software, analizando las entradas y salidas y verificando que el resultado sea el esperado.

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

Cuadrante 2: proporciona una guía y soporte al programador, pero desde la perspectiva del negocio, se suelen utilizar pruebas funcionales para historias de usuario y características de tal forma que se validan como el Product Owner las definió y como el usuario final las necesita. Para ello suele utilizarse BDD (Behavior driven development)

para automatizar pruebas y si no existe otra opción se pueden realizar las pruebas manualmente.

- **Pruebas funcionales:** dirigidas a asegurar que el sistema de software realiza correctamente todas las funciones que se han detallado en la especificación de requerimientos dada por el usuario del sistema o Product Owner.
- **Pruebas de regresión:** comprueban esencialmente si la funcionalidad anterior de la aplicación está trabajando coherentemente y que los nuevos cambios ejecutados no han introducido nuevos errores en la aplicación[21].
- **Prototipos:** están destinadas a validar la interfaz de usuario propuestas (prototipos). En estas pruebas los encargados de diseñar la interfaz de usuario elaboran pantallas de ejemplos o wireframes, los cuales validan con el cliente o Product Owner antes de que comience el desarrollo.
- **Pruebas UX (Experiencia de usuario):** es el proceso de evaluar diferentes aspectos de la usabilidad del software para determinar la mejor manera de que un sistema de software y sus elementos interactúen con su audiencia.

Cuadrante 3: proporciona una crítica del cliente desde la perspectiva del negocio, son pruebas a nivel del sistema de software completo, se valida que el sistema cumpla con las expectativas de funcionalidad y usabilidad. Se realizan manualmente en la mayoría de los casos ya que involucran al usuario y a los testers dentro de un ambiente real o simulado. En estas pruebas predominan las UAT (User Acceptance Test) a nivel del sistema.

- **Pruebas de aceptación:** el objetivo de las pruebas de aceptación es validar que un sistema de software cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de funcionalidad y rendimiento.

Estas pruebas van dirigidas a comprobar que el sistema cumple con los requisitos de funcionamiento esperado, recogidos en el catálogo de requisitos y en los criterios de aceptación del sistema de software, y conseguir así la aceptación final del sistema por parte del usuario.

- **Pruebas de exploración:** es un enfoque de pruebas en el que simultáneamente se aprende sobre la aplicación, se diseñan casos de prueba y se ejecutan esos casos de prueba.

Para realizar este tipo de pruebas se debe diseñar y ejecutar las mismas con un objetivo concreto. Con estas conclusiones se va aprendiendo sobre la aplicación, y se utiliza esta información para diseñar y ejecutar nuevas pruebas[22].

Cuadrante 4: proporciona una crítica del cliente desde la perspectiva tecnológica; son pruebas para asegurar la calidad completa del sistema de software en términos de

velocidad de carga, rendimiento, seguridad y escalabilidad. Por lo general estas pruebas se realizan con la ayuda de herramientas automáticas.

- **Pruebas de carga:** consiste en simular demanda sobre una aplicación de software y medir su resultado. Estas pruebas se realizan bajo demanda esperada y también en condiciones de sobrecarga.

Para ejecutar estas pruebas se requiere el uso de herramientas de testing que simulen carga como por ejemplo SoapUI¹.

- **Pruebas de rendimiento:** consisten en determinar que los tiempos de respuesta están dentro de los intervalos establecidos en las especificaciones del sistema.
- **Pruebas de estrés:** son pruebas de carga que se realizan con demandas mayores a la capacidad operativa, con frecuencia hasta llegar al punto de ruptura. Este tipo de prueba de software se utiliza para determinar la estabilidad de un sistema o aplicación, con especial atención en la disponibilidad y manejo de errores cuando se enfrenta a la sobrecarga.

Al igual que las pruebas de carga, se requieren de herramientas que simulen la demanda, SoapUI es una de las herramientas que permite simular peticiones para servidores de aplicaciones web.

- **Pruebas de seguridad:** consiste en probar los atributos o características de seguridad del sistema, si es un sistema seguro o no, si puede ser vulnerado, si existe control de acceso por medio de cuentas de usuario, si pueden ser vulnerados estos accesos.
- **Pruebas de resistencia:** estas pruebas implican someter al sistema a una carga determinada durante un periodo de tiempo, para determinar cómo se comporta luego de un uso prolongado.
- **Pruebas de escalabilidad:** consisten en verificar la capacidad de una aplicación de escalar cualquiera de sus características no funcionales, como, por ejemplo, la carga que soporta, número de transacciones, volúmenes de datos, entre otros.
- **Pruebas de recuperación:** se realizan para verificar que tan rápido y que tan bien se recupera una aplicación luego de experimentar un fallo de hardware o software. Por lo tanto, para realizar pruebas de recuperación se requiere forzar la falla y luego verificar si la recuperación ocurre adecuadamente.

Herramientas para pruebas

En la Tabla 2.24 se muestran algunas herramientas para realizar las pruebas descritas anteriormente:

¹ Soap UI: es una herramienta de gran alcance diseñada para realizar pruebas funcionales, de carga, estrés, seguridad y rendimiento. Tiene una interfaz simple e intuitiva y permite la utilización de métodos de captura y repetición.

Herramientas	Unitarias	Integración	Funcionales	Prototipos	Usabilidad	Aceptación	Carga	Rendimiento	Regresión	Estrés	Seguridad	Plataforma que soporta	Gratuita/Pago
Junit	x								x			Java/multiplataforma	Open Source
TestNG	x	x	x									Java	Open Source
SoapUI			x				x	x		x	x	Java	Open Source /PRO
Selenium			x			x			x			Java	Open Source
Balsamiq Mockups				x								Windows, Linux y Mac OS	Open Source
Ninja Mock				x								Android, iOS, Windows	Open Source
Five Second Test					x								Pago
First Click Testing					x								Pago
FitNesse						x						Java, C#, PHP, Ruby, .NET	Open Source
Cucumber						x						Ruby, Java y .NET	Open Source
k6							x	x					Open Source
Jtest	x		x				x		x			Java	Open Source
Jmeter							x	x		x		Servicios web SOAP/REST, Sitios web http y https, Bases de datos, FTP y servidores de correo PHP, ASP .NET, Java	Open Source

Tabla 2.24 Herramientas para pruebas

Estructura del plan de pruebas

En el desarrollo de un sistema de software, la fase de pruebas es crítica para asegurar que el producto sea enviado a un ambiente de producción con la calidad esperada por el cliente. Por tal razón, es necesario contar con un plan de pruebas para especificar minuciosamente las funciones a probar, cómo serán ejecutadas esas pruebas, quiénes serán los responsables y el cronograma para su ejecución[23].

A continuación, se detallan las secciones del Plan de Pruebas[24] a utilizar para la definición de pruebas en el Área de Desarrollo de EMASEO EP:

1. **Historial de versiones:** se muestra el historial de las versiones del documento de Plan de Pruebas, indicando quién elabora la versión y la fecha.
2. **Información del proyecto:** información general del proyecto, departamento o área organizacional, líder o gerente del proyecto, patrocinador y el gerente o líder de pruebas.
3. **Aprobaciones:** lista de personas que deben aprobar el plan de pruebas de software indicando su nombre, cargo, departamento y espacio para su firma.
4. **Resumen ejecutivo:** contiene el resumen de todo el contenido del plan de pruebas de software, establece cuál es el propósito, si es un plan maestro o detallado, restricciones (por ejemplo, de recursos o presupuesto) entre otros aspectos.
5. **Alcance de pruebas**
 - a) **Elementos de pruebas:** listado de todos los módulos, componentes o elementos que se van a probar.
 - b) **Nuevas funcionalidades a probar:** listado de lo que se va a probar desde el punto de vista del usuario. No es una descripción técnica del software sino de sus características y funcionalidades.
 - c) **Pruebas de regresión:** listado de funcionalidades no involucradas directamente en el desarrollo, cuyos componentes están siendo afectados y, por ende, deben probarse para asegurar que continúen funcionando adecuadamente.
 - d) **Funcionalidades a no probar:** lista de funcionalidades que no se van a probar. Se debe incluir información sobre por qué no se van a probar y los riesgos que se asumen al no probarlas.
 - e) **Enfoque de pruebas (Estrategia):** se definen los tipos de pruebas a realizar (funcionales, rendimiento, usabilidad, no funcionales, etc.), requerimientos especiales de las pruebas, nivel de pruebas de regresión, entre otros aspectos.
6. **Criterios de aceptación o rechazo:**
 - a) **Criterios de aceptación o rechazo:** son los criterios de aceptación que serán considerados para dar por completo el plan de pruebas de software; por ejemplo, completar 100% de las pruebas unitarias, cierto porcentaje de casos exitosos, cobertura de todos los componentes, porcentaje de defectos corregidos, entre otros.
 - b) **Criterios de suspensión:** establece bajo qué condiciones se detiene un conjunto de casos de pruebas.

- c) Criterios de reanudación:** luego de haber suspendido las pruebas, se establece bajo qué criterios se reanudarán.
- 7. Entregables:** establece qué se entregará como parte de la ejecución del plan, por ejemplo, documento de plan de pruebas, casos de pruebas, logs de errores, reporte de errores, reportes emitidos por las herramientas utilizadas en las pruebas, entre otros.
- 8. Recursos:**
- a) Requerimientos de entorno-Hardware:** lista de requerimientos de equipos de hardware y red necesarios para completar las actividades del plan de pruebas. Se incluyen servidores de bases de datos, aplicación, equipos de PC que necesiten los testers, conectividad a la red, entre otros.
 - b) Requerimiento de entorno-Software:** lista de requerimientos de software necesarios para completar las actividades de prueba, incluir accesos al sistema (entornos de pruebas) y base de datos, así como la instalación de software en el equipo de los testers.
 - c) Herramientas de pruebas requeridas:** especifica las herramientas de software, metodologías y técnicas empleadas para las pruebas.
 - d) Personal:** lista del personal necesario para completar las pruebas.
- 9. Planificación y organización:**
- a) Procedimiento para las pruebas:** especifica los procedimientos o metodologías de pruebas de software a emplear durante la ejecución del plan de pruebas.
 - b) Matriz de responsabilidades:** lista de las personas integrantes del equipo de QA (Quality Assurance – Aseguramiento de la Calidad) y sus responsabilidades.
 - c) Cronograma:** debe estar basado en la estimación de las actividades realizadas por el equipo de prueba. En él se identifican los hitos relevantes de las pruebas de software, se establecen dependencias y demás aspectos de un cronograma.
 - d) Premisas:** se describen las premisas relacionadas con las tareas de pruebas de software incluyendo limitaciones de tiempo, disponibilidad de recursos que se asumen, uso de metodología de pruebas, uso de herramientas, entre otras.
 - e) Dependencias y riesgos:** se listan los riesgos identificados en el proceso de pruebas de software, por ejemplo:
 - ✓ Dependencias con desarrollos
 - ✓ Dependencias con otros proyectos
 - ✓ Disponibilidad de recursos

- ✓ Limitaciones de tiempo

10. Referencias: lista de documentos que pueden citarse como apoyo o para ampliar el contenido del plan de pruebas.

11. Glosario: definiciones de los términos usados en la documentación.

2.4.8. Evaluación del producto del sprint

El producto o también llamado incremento para entregar al final de cada sprint, es de los entregables más importantes en el desarrollo de un sprint. Para asegurar que fue desarrollado bajo un proceso que cumpla con los estándares definidos y los requerimientos funcionales, se evaluará que el producto de software cumpla con los siguientes atributos de calidad:

- **Funcionalidad:** es la capacidad del producto de software para proveer funciones que cumplan las necesidades expresadas o implícitas cuando el software se utiliza en determinadas condiciones.

La funcionalidad se prueba mediante las pruebas funcionales, este tipo de pruebas tiene como objetivo verificar que los requerimientos especificados inicialmente se cumplan y arrojen el resultado esperado.

- **Usabilidad:** es la medida de la calidad de la experiencia que tiene un usuario cuando interactúa con un producto o sistema.

La usabilidad se prueba mediante las pruebas de UX (User Experience) su objetivo es medir que tan fácil es de usar un sistema de software.

- **Fiabilidad:** es la capacidad de recuperación ante un fallo y grado de previsión.

Este atributo se prueba mediante pruebas de recuperación las cuales consisten en demostrar que el sistema puede recuperarse ante fallos, sin comprometer la integridad de los datos.

- **Rendimiento:** está relacionado con los tiempos de respuesta, productividad, precisión, disponibilidad y uso de recursos.

Este atributo se prueba con la ayuda de pruebas de rendimiento las cuales tienen como objetivo determinar que los tiempos de respuesta están dentro de los intervalos establecidos en las especificaciones del sistema.

- **Mantenibilidad:** representa la capacidad del sistema de software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas[25].

Para evaluar si un sistema de software es mantenible se debe verificar el cumplimiento de las siguientes características[26]:

- Verificación de la aplicación de estándares de desarrollo, base de datos, nomenclatura reconocible y estándares para las interfaces de usuario.
- Verificar que la documentación sobre el sistema de software sea clara.
- Verificación de código parametrizado para promover la reutilización de código.
- Verificar si los algoritmos utilizados están optimizados.
- Verificar la facilidad para realizar pruebas.

Si el incremento a probar cumple con las características mencionadas anteriormente se dirá que es MANTENIBLE, caso contrario se debería verificar que se cumplan con las características mencionadas para declararlo como MANTENIBLE.

El Departamento de Aseguramiento de la Calidad o el equipo de pruebas será el encargado de verificar que el incremento cumpla con las características definidas.

Formato de evaluación

La Tabla 2.25 permitirá evaluar la cantidad de criterios de calidad con los que cumple el producto o incremento de software presentado al final de cada sprint.

NRO Sprint		
ID's HISTORIAS DE USUARIO IMPLEMENTADAS		
Atributos de calidad	Seleccione con una x los atributos que cumple el producto	Observaciones
Funcionalidad		
Usabilidad		
Fiabilidad		
Rendimiento		
Mantenibilidad		
Total atributos calidad cumplidos		

Tabla 2.25 Formato de evaluación Producto del Sprint

A continuación, se explican los campos de la Tabla 3.17:

- **Nro. Sprint:** número del sprint que se va a evaluar.
- **ID's historias de usuario implementadas:** identificadores de las historias de usuario que se implementaron en el sprint.
- **Evidencia de funcionamiento:** material que evidencie que se implementaron las historias de usuario. Por ejemplo, capturas de pantalla, código, etc.
- Selección de los atributos de calidad con los que cumple el producto:
 - Funcionalidad
 - Usabilidad
 - Fiabilidad
 - Rendimiento
 - Mantenibilidad

- **Observaciones:** observaciones respecto a la prueba que se realizó, se pueden anexar resultados, novedades presentadas durante la aplicación de la prueba, etc.
- **Total de atributos de calidad cumplidos:** número de atributos de calidad con los que cumple el incremento.

3. CASO DE ESTUDIO: MÓDULO DE LOGÍSTICA-SISTEMA DE GESTIÓN DE OPERACIONES

3.1. Aplicación de la metodología híbrida (Scrum + UML) en el desarrollo del Módulo de Logística del SGO

3.1.1. Inicio del proyecto

a) Visión del proyecto

La visión del proyecto a implementar llamado “Sistema de Gestión de Operaciones” para EMASEO EP se describe en el **ANEXO 1** Documento de Visión.

b) Formación del equipo Scrum

En la Tabla 3.1 se describen los roles y principales responsabilidades de los miembros que conforman el equipo Scrum para la implementación de este proyecto.

Nombre de la Persona	Rol en Scrum	Principales Responsabilidades	Grado de participación en el proyecto
Eduardo Loachamin	Product Owner	*Levantar los requerimientos del Sistema de Gestión de Operaciones con la colaboración del Scrum Team *Validar que los requerimientos definidos se implementen y cumplan con la función definida anteriormente.	80 %
Samuel Villareal	Scrum Team	*Implementar los requerimientos definidos por el Product Owner.	100 %
Yessenia Erraez	Scrum Team	*Realizar la documentación requerida en la implementación del proyecto.	80 %
Frans Naranjo	Scrum Team	*Implementar los requerimientos definidos por el Product Owner.	80 %

Tabla 3.1 Roles y responsabilidades de Scrum

c) Product Backlog

En la Tabla 3.2 se describen las tareas que en conjunto conforman el Product Backlog con su respectivo identificador, prioridad, tamaño, esfuerzo y tiempo estimado de implementación.

ID	Tarea	Prioridad	Tamaño	Esfuerzo	Tiempo(horas)
PLA-001	Gestionar frecuencia	1	S	2	8
PLA-002	Gestionar horario	1	S	2	8
PLA-003	Gestionar tipos de servicios	1	S	2	8
PLA-004	Gestionar talento humano	1	L	5	24
PLA-005	Gestionar maquinaria	1	L	5	24
PLA-006	Gestionar administraciones zonales	1	S	2	8
PLA-007	Gestionar ruta ubicación	1	L	5	24
PLA-008	Gestionar sectores	1	L	5	24
PLA-009	Gestionar programación diaria de rutas asignando recursos	1	XL	8	40
PLA-010	Gestionar cambio de supervisor de ruta	1	M	3	16
PLA-011	Generar reporte de programación	1	S	2	8
Esfuerzo y tiempo totales para la implementación de tareas				41	192

Tabla 3.2 Product Backlog Tareas

Para validar las pruebas de aceptación se han escrito criterios de aceptación descritos en el **ANEXO 2**.

d) Diagramas de caso de uso

A continuación, se representarán los diagramas de casos de uso de los requerimientos definidos para el Módulo de Logística.

El Módulo de Logística está conformado por los siguientes catálogos, los mismos que se conforman por submódulos:

Catálogo de Insumos: en este catálogo se mostrarán todos los elementos y recursos necesarios para planificar una ruta.

- **Frecuencia:** es la cantidad de días en los que se ejecutara un servicio seleccionado para la ruta que se desea planificar, por ejemplo, diariamente, dos veces por semana, cuatro veces por semana, etc. En este submódulo se mostrará la lista de frecuencias registradas por el usuario Analista de Logística quien más adelante deberá seleccionar una frecuencia para la ruta que desea programar.

- **Horario:** es el horario en el que se ejecutará el servicio seleccionado para la ruta que se desea planificar, por ejemplo, matutino 9:00 am – 11 am, vespertino 2:00pm – 4 pm, etc. En este submódulo se mostrará la lista de horarios registrados por el usuario Analista de Logística quien más adelante deberá seleccionar una frecuencia para la ruta que desea programar.
- **Tipo de servicio:** en este submódulo se registrarán los tipos de servicios que ofrece la empresa a la ciudadanía, por ejemplo, recolección a pie de vereda, hidrolavado, recolección contenerizada, etc. Entre estos servicios el usuario Analista de Logística deberá seleccionar uno para la ruta que planificará más adelante.
- **Talento Humano:** este submódulo tendrá dos pestañas: cargo y empleado.

En la pestaña **Cargo** se registrarán los tipos de cargo que existen en la empresa, por ejemplo, servidor público 1, servidor público 2, supervisor, etc. El usuario Analista de Logística seleccionara en la pestaña **Empleado** el cargo que le corresponde a persona.

En la pestaña **Empleado** se mostrará la lista de empleados de la empresa que se consumió desde el sistema CGWEB de Talento Humano y también se podrá registrar nuevos empleados, el usuario Analista de Logística seleccionara más adelante la cantidad de empleados que se necesitan para llevar a cabo la ejecución de una ruta.

- **Maquinaria** en el submódulo **Maquinaria** se mostrará el inventario de maquinaria que existe en la empresa que se consumió desde el sistema **D-SIES WEB** y también se podrá registrar nueva maquinaria, el usuario Analista de Logística deberá seleccionar más adelante el tipo de maquinaria que necesitará acorde al servicio que se prestará en la programación de una ruta.

Catálogo de Inventario de ruta: en este catálogo se registrará información para armar las rutas que se van a programar en el catálogo de Logística.

- **Administración zonal:** en este submódulo se mostrarán las administraciones zonales registradas por el usuario Analista de Logística para más adelante asignar la administración zonal correspondiente a los sectores que se registrarán.
- **Sectores:** en este submódulo se registrarán los sectores que más adelante se asignarán a las rutas que el usuario Analista de Logística programará.
- **Ruta ubicación:** en este submódulo se registrarán las rutas a las cuales más adelante el usuario Analista de Logística deberá asignar los elementos y recursos necesarios (frecuencia, horario, tipo de servicio, rutas, personal y maquinaria) para poder ejecutarlas.

Catálogo de Logística: en este catálogo se gestionarán las rutas creadas anteriormente, se asignarán los recursos necesarios para ejecutarlas y en caso de ser necesario se podrá hacer una reasignación de rutas a un supervisor distinto al asignado inicialmente. También se podrá descargar el reporte de programación de rutas.

- **Programacion base:** en este submódulo el usuario Analista de Logística deberá seleccionar los elementos y recursos para la ejecución de la ruta, por ejemplo, seleccionar frecuencia, horario, tipo de servicio, además deberá asignar la ruta, personal y maquinaria.
- **Cambio de supervisor de ruta:** en este submódulo el usuario Analista de Logista podrá asignar una ruta programada previamente a un nuevo supervisor de ruta.
- **Reporte de programación diaria:** en este submódulo el usuario Analista de logística podrá descargar un reporte de programación diaria escogiendo una fecha de inicio y seleccionando el supervisor del cual desea descargar esta información.

1. DIAGRAMAS DE CASOS DE USO DEL CATÁLOGO DE INSUMOS

a) Frecuencia de ruta

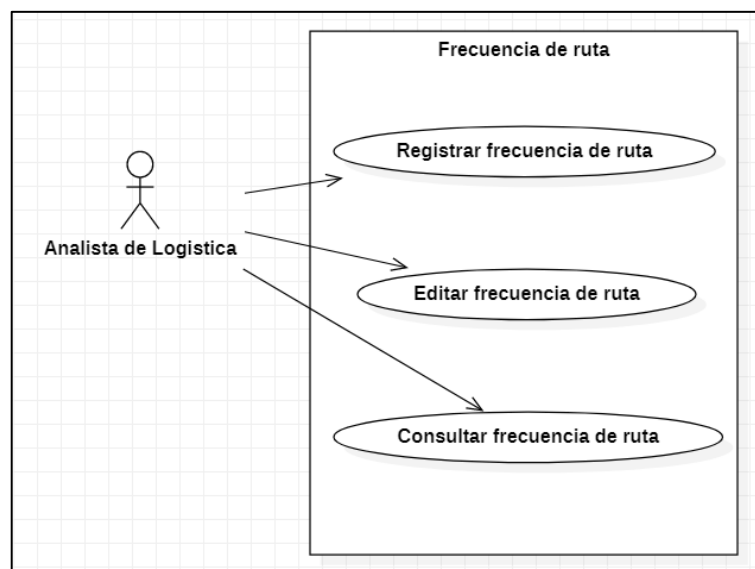


Figura 3.1 Caso de uso Frecuencia

Descripción de caso de uso: Registrar frecuencia de ruta

Caso de Uso	Registrar frecuencia de ruta	Identificador: FRE-001
Actores	Analista de Logística	
Descripción	Registrar frecuencia con los datos correspondientes.	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Frecuencia.	
Flujo normal		
Paso	Actividades	
1.	Dar clic en el botón Nuevo.	
2.	Ingresar los datos correspondientes de la frecuencia de ruta.	
3.	Dar clic en el botón Guardar.	
Flujo alternativo		
Paso	Actividades	

2a.	El usuario Analista de Logística no llena todos los campos de la frecuencia. El sistema muestra un mensaje en donde pida al usuario llenar todos los campos de la frecuencia para que esta se pueda guardar en la base de datos.
Postcondición	El registro de frecuencia se guardó en la base de datos exitosamente.
Dependencias	N/A

Tabla 3.3 Descripción caso de uso Registrar frecuencia

Descripción de caso de uso: Editar frecuencia de ruta

Caso de Uso	Editar frecuencia de ruta	Identificador: FRE-002
Actores	Analista de Logística	
Descripción	Editar dato o datos de frecuencia de ruta	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Frecuencia. La frecuencia debe estar registrada en la base de datos.	
Flujo normal		
Paso	Actividades	
1.	Buscar el registro de frecuencia a editar ingresando las siglas o días en la barra de búsqueda.	
2.	Dar clic en el botón Editar.	
3.	Editar el dato o datos de la frecuencia	
4.	Dar clic en el botón Guardar.	
Flujo alternativo		
Paso	Actividades	
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.	
Postcondición	El dato o los datos de la frecuencia se actualizaron en la base de datos exitosamente.	
Dependencias	FRE-001	

Tabla 3.4 Descripción de caso de uso Editar frecuencia

Descripción de caso de uso: Consultar frecuencia de ruta

Caso de Uso	Consultar frecuencia de ruta	Identificador: FRE-003
Actores	Analista de Logística	
Descripción	Consultar datos de frecuencia de ruta	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Frecuencia. La frecuencia debe estar registrada en la base de datos.	
Flujo normal		
Paso	Actividades	
1.	Buscar el registro de frecuencia ingresando las siglas o días en la barra de búsqueda.	
2.	Dar clic en el botón Buscar	
3.	El registro buscado se muestra en la tabla	
Flujo alternativo		
Paso	Actividades	
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.	
Postcondición	El registro buscado se muestra en la tabla.	
Dependencias	FRE-001	

Tabla 3.5 Descripción de caso de uso Consultar frecuencia

b) Horario de ruta

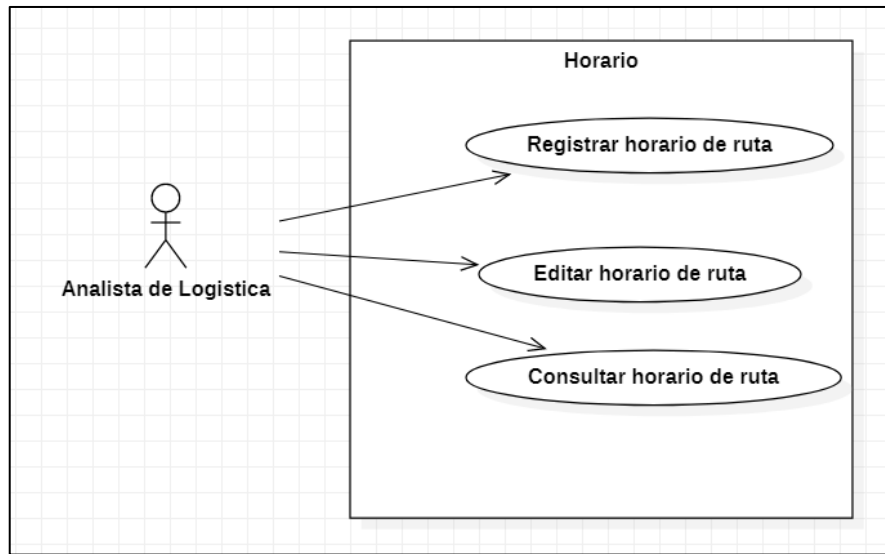


Figura 3.2 Caso de uso Horario

Descripción de caso de uso: Registrar horario de ruta

Caso de Uso	Registrar horario de ruta	Identificador: HOR-001
Actores	Analista de Logística	
Descripción	Registrar horario con los datos correspondientes.	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Horario.	
Flujo normal		
Paso	Actividades	
1.	Dar clic en el botón Nuevo.	
2.	Ingresar los datos correspondientes al horario de ruta.	
3.	Dar clic en el botón Guardar.	
Flujo alternativo		
Paso	Actividades	
2a.	El usuario Analista de Logística no llena todos los campos del horario. El sistema muestra un mensaje en donde pida al usuario llenar todos los campos del horario para que este se pueda guardar en la base de datos.	
Postcondición	El registro se ha guardado correctamente en la base de datos.	
Dependencias	N/A	

Tabla 3.6 Descripción caso de uso Registrar horario

Descripción de caso de uso: Editar horario de ruta

Caso de Uso	Editar horario de ruta	Identificador: HOR-002
Actores	Analista de Logística	
Descripción	Editar dato o datos de horario de ruta	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Horario. El horario debe estar registrado en la base de datos.	
Flujo normal		
Paso	Actividades	
1.	Buscar el registro de horario a editar ingresando las siglas o descripción en la barra de búsqueda.	
2.	Dar clic en el botón Editar.	
3.	Editar el dato o datos del horario	

4.	Dar clic en el botón Guardar.
Flujo alternativo	
Paso	Actividades
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.
Postcondición	El dato o los datos del horario se actualizaron en la base de datos exitosamente.
Dependencias	HOR-001

Tabla 3.7 Descripción caso de uso Editar horario

Descripción de caso de uso: Consultar horario de ruta

Caso de Uso	Consultar horario de ruta	Identificador: HOR-003
Actores	Analista de Logística	
Descripción	Consultar datos de horario de ruta	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Horario. El horario debe estar registrada en la base de datos.	
Flujo normal		
Paso	Actividades	
1.	Buscar el registro de horario ingresando las siglas o descripción en la barra de búsqueda.	
2.	Dar clic en el botón Buscar	
3.	El registro buscado se muestra en la tabla	
Flujo alternativo		
Paso	Actividades	
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.	
Postcondición	El horario buscado se muestra en la tabla.	
Dependencias	HOR-001	

Tabla 3.8 Descripción caso de uso Consultar horario

c) Tipo de servicio

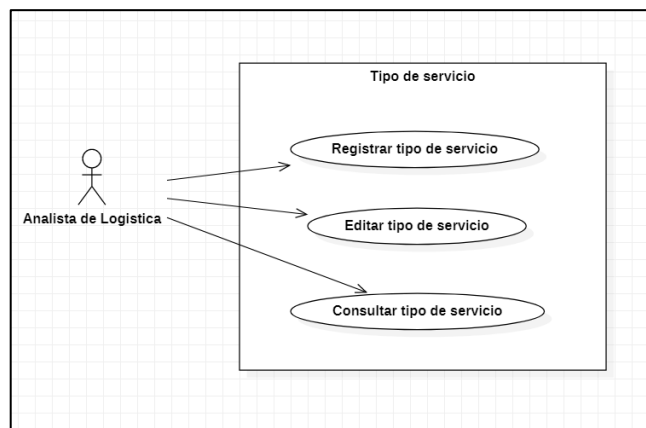


Figura 3.3 Caso de uso Tipo de servicio

Descripción de caso de uso: Registrar tipo de servicio

Caso de Uso	Registrar tipo de servicio	Identificador: SER-001
Actores	Analista de Logística	
Descripción	Registrar tipo de servicio con los datos correspondientes.	

Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Tipo de servicio.
Flujo normal	
Paso	Actividades
1.	Dar clic en el botón Nuevo.
2.	Ingresar los datos correspondientes al tipo de servicio
3.	Dar clic en el botón Guardar.
Flujo alternativo	
Paso	Actividades
2a.	El usuario Analista de Logística no llena todos los campos del tipo de servicio. El sistema muestra un mensaje en donde pide al usuario llenar todos los campos del tipo de servicio para que este se pueda guardar en la base de datos.
Postcondición	El registro se ha guardado correctamente en la base de datos.
Dependencias	N/A

Tabla 3.9 Descripción de caso de uso Registrar tipo de servicio

Descripción de caso de uso: Editar tipo de servicio

Caso de Uso	Editar tipo de servicio	Identificador: SER-002
Actores	Analista de Logística	
Descripción	Editar dato o datos de tipo de servicio	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Tipo de servicio. El tipo de servicio debe estar registrado en la base de datos.	
Flujo normal		
Paso	Actividades	
1.	Buscar el registro de tipo de servicio a editar ingresando las siglas, denominación o descripción en la barra de búsqueda.	
2.	Dar clic en el botón Editar.	
3.	Editar el dato o datos del horario	
4.	Dar clic en el botón Guardar.	
Flujo alternativo		
Paso	Actividades	
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.	
Postcondición	El dato o los datos del tipo de servicio se actualizaron en la base de datos exitosamente.	
Dependencias	SER-001	

Tabla 3.10 Descripción de caso de uso Editar tipo de servicio

Descripción de caso de uso: Consultar tipo de servicio

Caso de Uso	Consultar tipo de servicio	Identificador: SER-003
Actores	Analista de Logística	
Descripción	Consultar datos del tipo de servicio	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Tipo de servicio. El tipo de servicio debe estar registrada en la base de datos.	
Flujo normal		
Paso	Actividades	

1.	Buscar el tipo de servicio ingresando las siglas, denominación o descripción en la barra de búsqueda.
2.	Dar clic en el botón Buscar
3.	El registro buscado se muestra en la tabla
Flujo alternativo	
Paso	Actividades
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arrojo resultados.
Postcondición	El tipo de servicio buscado se muestra en la tabla.
Dependencias	SER-001

Tabla 3.11 Descripción de caso de uso Consultar tipo de servicio

d) Talento humano – Cargo

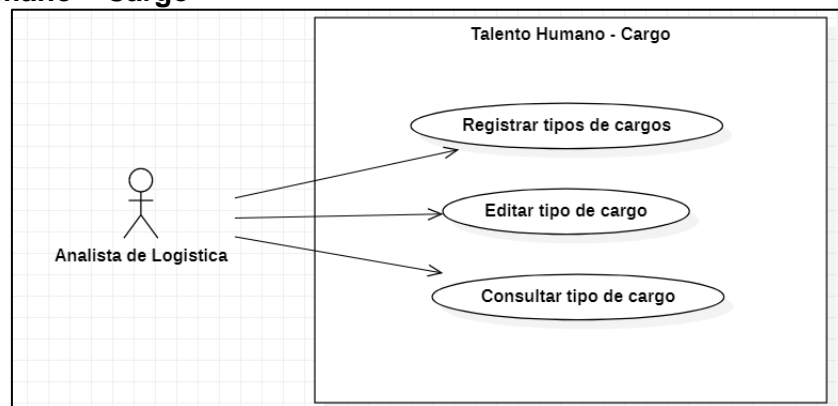


Figura 3.4 Caso de uso Talento humano

Descripción de caso de uso: Registrar tipo de cargo

Caso de Uso	Registrar tipo de cargo	Identificador: CAR-001
Actores	Analista de Logística	
Descripción	Registrar tipo de cargo con los datos correspondientes.	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Cargos.	
Flujo normal		
Paso	Actividades	
1.	Dar clic en el botón Nuevo.	
2.	Ingresar los datos correspondientes al tipo de cargo.	
3.	Dar clic en el botón Guardar.	
Flujo alternativo		
Paso	Actividades	
2a.	El usuario Analista de Logística no llena todos los campos del tipo de cargo. El sistema muestra un mensaje en donde pida al usuario llenar todos los campos del tipo de cargo para que este se pueda guardar en la base de datos.	
Postcondición	El registro se ha guardado correctamente en la base de datos.	
Dependencias	N/A	

Tabla 3.12 Descripción de caso de uso Cargo

Descripción de caso de uso: Editar tipo de cargo

Caso de Uso	Editar tipo de cargo	Identificador: CAR-002
Actores	Analista de Logística	
Descripción	Editar dato o datos de tipo de cargo	

Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Cargos. El tipo de cargo debe estar registrado en la base de datos.
Flujo normal	
Paso	Actividades
1.	Buscar el registro de tipo de cargo a editar ingresando las siglas, denominación o descripción en la barra de búsqueda.
2.	Dar clic en el botón Editar.
3.	Editar el dato o datos del cargo.
4.	Dar clic en el botón Guardar.
Flujo alternativo	
Paso	Actividades
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.
Postcondición	El dato o los datos del tipo de cargo se actualizaron en la base de datos exitosamente.
Dependencias	CAR-001

Tabla 3.13 Descripción del caso de uso Editar cargo

Descripción de caso de uso: Consultar tipo de cargo

Caso de Uso	Consultar tipo de cargo	Identificador: CAR-003
Actores	Analista de Logística	
Descripción	Consultar datos del tipo de cargo	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo Cargos. El tipo de cargo debe estar registrada en la base de datos.	
Flujo normal		
Paso	Actividades	
1.	Buscar el tipo de cargo ingresando las siglas, denominación o descripción en la barra de búsqueda.	
2.	Dar clic en el botón Buscar	
3.	El registro buscado se muestra en la tabla	
Flujo alternativo		
Paso	Actividades	
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.	
Postcondición	El tipo de cargo buscado se muestra en la tabla.	
Dependencias	CAR-001	

Tabla 3.14 Descripción del caso de uso Consultar cargo

e) Empleados - Talento Humano

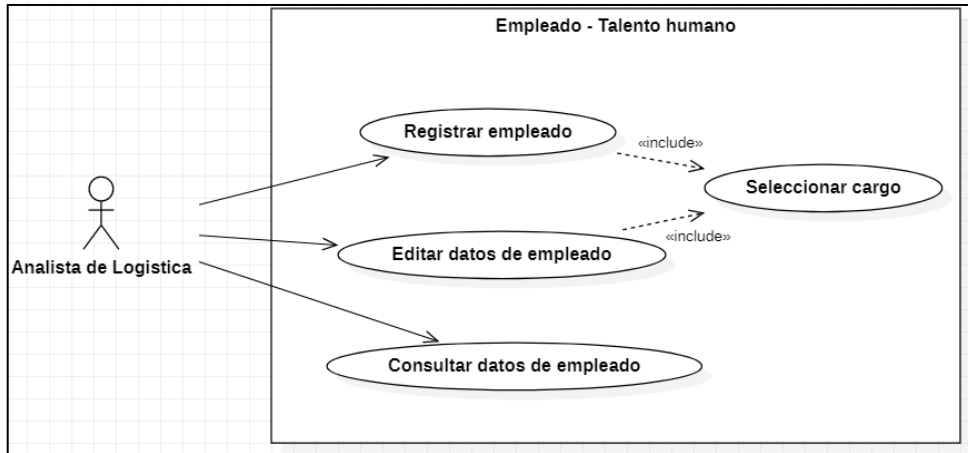


Figura 3.5 Caso de uso Empleado-Talento humano

Descripción de caso de uso: Registrar empleado

Caso de Uso	Registrar empleado	Identificador: EMP-001
Actores	Analista de Logística	
Descripción	Registrar empleado con los datos correspondientes.	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Empleados. El cargo que se va a asignar al empleado tiene que estar registrado en el submódulo de cargos.	
Flujo normal		
Paso	Actividades	
1.	Dar clic en el botón Nuevo.	
2.	Ingresar los datos correspondientes del empleado y seleccionar su cargo.	
3.	Dar clic en el botón Guardar.	
Flujo alternativo		
Paso	Actividades	
2a.	El usuario Analista de Logística no llena todos los campos del empleado. El sistema muestra un mensaje en donde pide al usuario llenar todos los campos del empleado para que este se pueda guardar en la base de datos.	
2b.	El cargo que se desea asignar al empleado no está registrado en la base de datos por lo tanto no se muestra en la lista de cargos. El usuario Analista de Logística debe regresar al submódulo de Cargos y registrar el cargo que asignara al empleado, luego puede volver a repetir el proceso de creación del empleado y asignar el cargo correspondiente.	
Postcondición	El registro se ha guardado correctamente en la base de datos.	
Dependencias	CAR-001	

Tabla 3.15 Descripción del caso de uso Registrar empleado

Descripción de caso de uso: Editar datos de empleado

Caso de Uso	Editar datos del empleado	Identificador: EMP-002
Actores	Analista de Logística	
Descripción	Editar dato o datos del empleado	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Empleados. El empleado debe estar registrado en la base de datos. El tipo de cargo que se va a asignar debe estar registrado en la base de datos.	
Flujo normal		

Paso	Actividades
1.	Buscar el empleado a editar ingresando la identificación, nombre, apellido, cargo o rol en la barra de búsqueda.
2.	Dar clic en el botón Editar.
3.	Editar el dato o datos del empleado.
4.	Dar clic en el botón Guardar.
Flujo alternativo	
Paso	Actividades
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.
Postcondición	El dato o los datos del empleado se actualizaron en la base de datos exitosamente.
Dependencias	EMP-001 CAR-001

Tabla 3.16 Descripción del caso de uso Editar datos empleado

Descripción de caso de uso: Consultar datos de empleado

Caso de Uso	Consultar datos del empleado	Identificador: EMP-003
Actores	Analista de Logística	
Descripción	Consultar datos del empleado	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo Empleados. El empleado debe estar registrado en la base de datos.	
Flujo normal		
Paso	Actividades	
1.	Buscar el tipo de cargo ingresando la identificación, nombre, apellido, cargo o rol en la barra de búsqueda.	
2.	Dar clic en el botón Buscar	
3.	El registro buscado se muestra en la tabla	
Flujo alternativo		
Paso	Actividades	
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.	
Postcondición	El empleado buscado se muestra en la tabla.	
Dependencias	EMP-001	

Tabla 3.17 Descripción del caso de uso Consultar empleado

f) Maquinaria

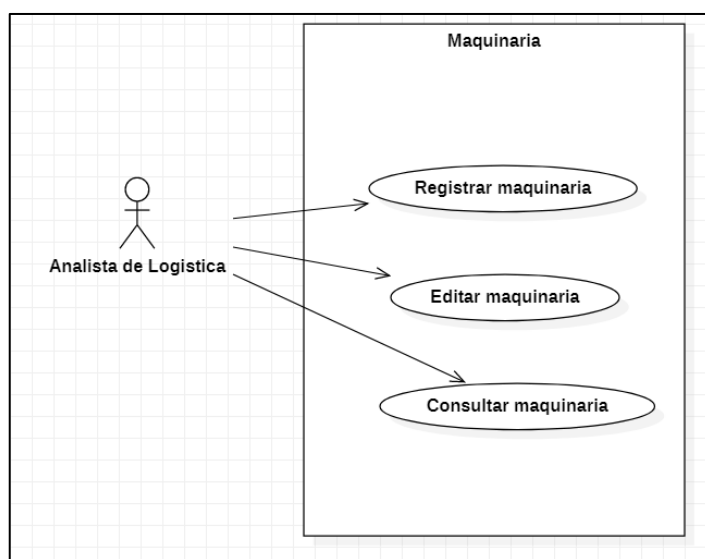


Figura 3.6 Caso de uso Maquinaria

Descripción de caso de uso: Registrar maquinaria

Caso de Uso	Registrar maquinaria	Identificador: MAQ-001
Actores	Analista de Logística	
Descripción	Registrar maquinaria con los datos correspondientes.	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Maquinaria.	
Flujo normal		
Paso	Actividades	
1.	Dar clic en el botón Nuevo.	
2.	Ingresar los datos correspondientes de la maquinaria.	
3.	Dar clic en el botón Guardar.	
Flujo alternativo		
Paso	Actividades	
2a.	El usuario Analista de Logística no llena todos los campos de la maquinaria. El sistema muestra un mensaje en donde pida al usuario llenar todos los campos de la maquinaria para que esta se pueda guardar en la base de datos.	
Postcondición	El registro se ha guardado correctamente en la base de datos.	
Dependencias	N/A	

Tabla 3.18 Descripción del caso de uso Registrar maquinaria

Descripción de caso de uso: Editar datos de maquinaria

Caso de Uso	Editar datos de la maquinaria	Identificador: MAQ-002
Actores	Analista de Logística	
Descripción	Editar dato o datos de la maquinaria	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Maquinaria La maquinaria debe estar registrada en la base de datos.	
Flujo normal		
Paso	Actividades	

1.	Buscar la maquinaria a editar ingresando disco, placa, marca, clase, modelo o tipo en la barra de búsqueda.
2.	Dar clic en el botón Editar.
3.	Editar el dato o datos de la maquinaria.
4.	Dar clic en el botón Guardar.
Flujo alternativo	
Paso	Actividades
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.
Postcondición	El dato o los datos del empleado se actualizaron en la base de datos exitosamente.
Dependencias	MAQ-001

Tabla 3.19 Descripción del caso de uso Editar maquinaria

Descripción de caso de uso: Consultar maquinaria

Caso de Uso	Consultar datos de la maquinaria	Identificador: MAQ-003
Actores	Analista de Logística	
Descripción	Consultar datos de la maquinaria	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo Maquinaria. La maquinaria debe estar registrada en la base de datos.	
Flujo normal		
Paso	Actividades	
1.	Buscar la maquinaria a editar ingresando disco, placa, marca, clase, modelo o tipo en la barra de búsqueda.	
2.	Dar clic en el botón Buscar	
3.	El registro buscado se muestra en la tabla	
Flujo alternativo		
Paso	Actividades	
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.	
Postcondición	La maquinaria buscada se muestra en la tabla.	
Dependencias	MAQ-001	

Tabla 3.20 Descripción del caso de uso Consultar maquinaria

2. DIAGRAMAS DE CASO DE USO DEL CATÁLOGO DE INVENTARIO DE RUTA

g) Administración zonal

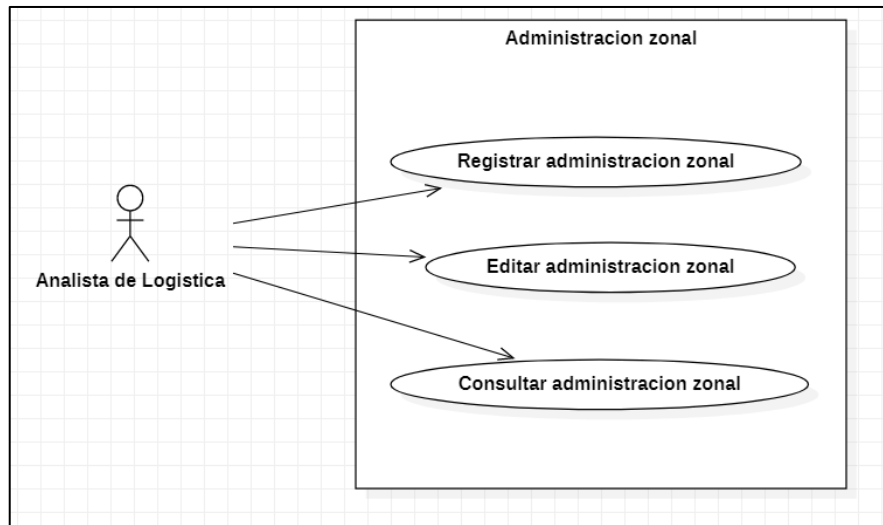


Figura 3.7 Caso de uso Administración zonal

Descripción de caso de uso: Registrar administración zonal

Caso de Uso	Registrar administración zonal	Identificador: ADM-001
Actores	Analista de Logística	
Descripción	Registrar administración zonal con los datos correspondientes.	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Administración zonal.	
Flujo normal		
Paso	Actividades	
1.	Dar clic en el botón Nuevo.	
2.	Ingresar los datos correspondientes de la administración zonal.	
3.	Dar clic en el botón Guardar.	
Flujo alternativo		
Paso	Actividades	
2a.	El usuario Analista de Logística no llena todos los campos de la administración zonal. El sistema muestra un mensaje en donde pida al usuario llenar todos los campos de la administración zonal para que esta se pueda guardar en la base de datos.	
Postcondición	El registro se ha guardado correctamente en la base de datos.	
Dependencias	N/A	

Tabla 3.21 Descripción del caso de uso Registrar administración zonal

Descripción de caso de uso: Editar administración zonal

Caso de Uso	Editar datos de administración zonal	Identificador: ADM-002
Actores	Analista de Logística	
Descripción	Editar dato o datos de la administración zonal	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Administración zonal. La administración zonal debe estar registrada en la base de datos.	
Flujo normal		
Paso	Actividades	

1.	Buscar la administración zonal ingresando siglas, denominación o tipo en la barra de búsqueda.
2.	Dar clic en el botón Editar.
3.	Editar el dato o datos de la administración zonal.
4.	Dar clic en el botón Guardar.
Flujo alternativo	
Paso	Actividades
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.
Postcondición	El dato o los datos de la administración zonal se actualizaron en la base de datos exitosamente.
Dependencias	ADM-001

Tabla 3.22 Descripción del caso de uso Editar administración zonal

Descripción de caso de uso: Consultar administración zonal

Caso de Uso	Consultar datos de la administración zonal	Identificador: ADM-003
Actores	Analista de Logística	
Descripción	Consultar datos de la administración zonal	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo Administración zonal. La administración zonal debe estar registrada en la base de datos.	
Flujo normal		
Paso	Actividades	
1.	Buscar la administración zonal a editar ingresando siglas, denominación o tipo en la barra de búsqueda.	
2.	Dar clic en el botón Buscar	
3.	El registro buscado se muestra en la tabla	
Flujo alternativo		
Paso	Actividades	
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.	
Postcondición	La administración zonal buscada se muestra en la tabla.	
Dependencias	ADM-001	

Tabla 3.23 Descripción del caso de uso Consultar administración zonal

h) Sectores

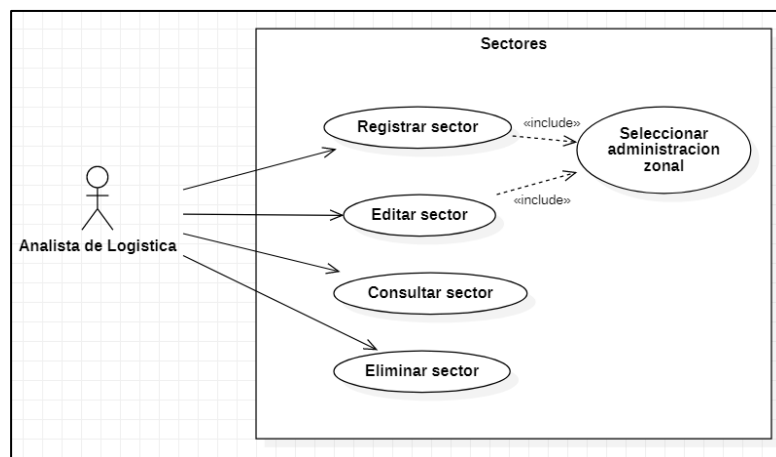


Figura 3.8 Caso de uso Sectores

Descripción de caso de uso: Registrar sector

Caso de Uso	Registrar sector	Identificador: SEC-001
Actores	Analista de Logística	
Descripción	Registrar sector con los datos correspondientes.	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Sectores. Debe estar registrada previamente la administración zonal a seleccionar.	
Flujo normal		
Paso	Actividades	
1.	Dar clic en el botón Nuevo.	
2.	Ingresar los datos correspondientes del sector y seleccionar la administración zonal.	
3.	Dar clic en el botón Guardar.	
Flujo alternativo		
Paso	Actividades	
2a.	El usuario Analista de Logística no llena todos los campos del sector. El sistema muestra un mensaje en donde pida al usuario llenar todos los campos del sector para que esta se pueda guardar en la base de datos.	
2b.	La administración zonal a la que el sector pertenece no se muestra entre las administraciones zonales registradas. El usuario Analista de logística deberá volver al submódulo de Administración zonal, registrar la administración zonal correspondiente y luego repetir el proceso de registro del sector con la administración zonal correspondiente.	
Postcondición	El registro se ha guardado correctamente en la base de datos.	
Dependencias	ADM-001	

Tabla 3.24 Descripción del caso de uso Registrar sector

Descripción de caso de uso: Editar sector

Caso de Uso	Editar datos del sector	Identificador: SEC-002
Actores	Analista de Logística	
Descripción	Editar dato o datos del sector	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Sectores El sector debe estar registrado en la base de datos.	
Flujo normal		
Paso	Actividades	
1.	Buscar el sector ingresando siglas, denominación, tipo, siglas de región, denominación de región o estado en la barra de búsqueda.	
2.	Dar clic en el botón Editar.	
3.	Editar el dato o datos del sector.	
4.	Dar clic en el botón Guardar.	
Flujo alternativo		
Paso	Actividades	
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.	
Postcondición	El dato o los datos del sector se actualizaron en la base de datos exitosamente.	
Dependencias	SEC-001 ADM-001	

Tabla 3.25 Descripción del caso de uso Editar sector

Descripción de caso de uso: Consultar sector

Caso de Uso	Consultar datos del sector	Identificador: SEC-003
Actores	Analista de Logística	
Descripción	Consultar datos de la administración zonal	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo Sectores. El sector deberá estar registrada en la base de datos.	
Flujo normal		
Paso	Actividades	
1.	Buscar el sector ingresando siglas, denominación, tipo, siglas de región, denominación de región o estado en la barra de búsqueda.	
2.	Dar clic en el botón Buscar	
3.	El registro buscado se muestra en la tabla	
Flujo alternativo		
Paso	Actividades	
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.	
Postcondición	El sector buscado se muestra en la tabla.	
Dependencias	SEC-001	

Tabla 3.26 Descripción del caso de uso Consultar sector

Descripción de caso de uso: Eliminar sector

Caso de Uso	Eliminar datos del sector	Identificador: SEC-004
Actores	Analista de Logística	
Descripción	Eliminar sector	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo Sectores. El sector deberá estar registrada en la base de datos.	
Flujo normal		
Paso	Actividades	
1.	Buscar el sector ingresando siglas, denominación, tipo, siglas de región, denominación de región o estado en la barra de búsqueda.	
2.	Dar clic en el botón Eliminar.	
3.	Dar clic en el botón Aceptar para confirmar la eliminación del sector.	
4.	El sector se eliminó.	
Flujo alternativo		
Paso	Actividades	
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.	
3a.	Si no está seguro de eliminar el sector el usuario deberá dar clic en el botón Cancelar.	
Postcondición	El sector se eliminó correctamente de la base de datos.	
Dependencias	SEC-001	

Tabla 3.27 Descripción del caso de uso Eliminar sector

i) Ruta ubicación

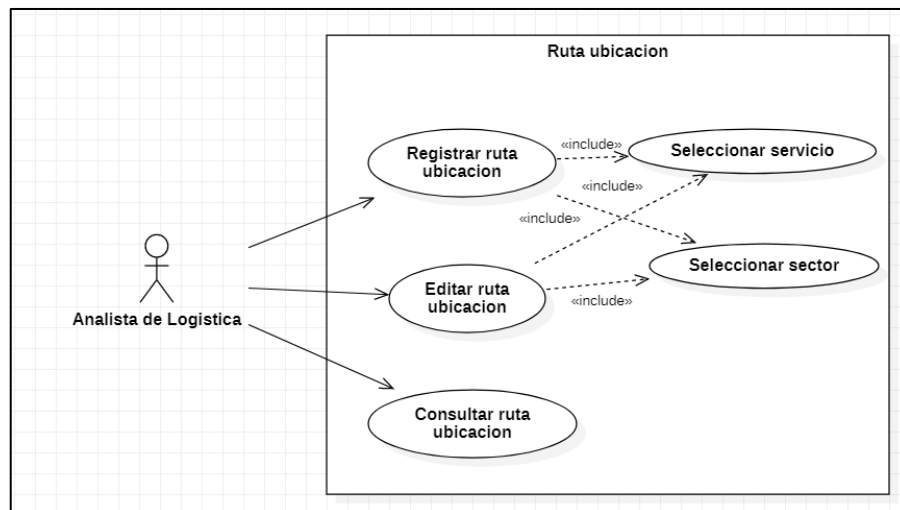


Figura 3.9 Caso de uso Ruta ubicación

Descripción de caso de uso: Registrar ruta ubicación

Caso de Uso	Registrar ruta ubicación	Identificador: RUT-001
Actores	Analista de Logística	
Descripción	Registrar ruta con los datos correspondientes.	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Ruta ubicación. Debe estar registrado el servicio a seleccionar. Debe estar registrado el sector a seleccionar.	
Flujo normal		
Paso	Actividades	
1.	Dar clic en el botón Nuevo.	
2.	Ingresar los datos correspondientes a la ruta y seleccionar el servicio y sectores correspondientes a la ruta.	
3.	Dar clic en el botón Guardar.	
Flujo alternativo		
Paso	Actividades	
2a.	El usuario Analista de Logística no llena todos los campos de la ruta. El sistema muestra un mensaje en donde pida al usuario llenar todos los campos de la ruta para que esta se pueda guardar en la base de datos.	
Postcondición	El registro se ha guardado correctamente en la base de datos.	
Dependencias	SER-001 SEC-001	

Tabla 3.28 Descripción del caso de uso Registrar ruta ubicación

Descripción de caso de uso: Editar ruta ubicación

Caso de Uso	Editar datos de la ruta	Identificador: RUT-002
Actores	Analista de Logística	
Descripción	Editar dato o datos de la ruta	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Ruta ubicación. La ruta debe estar registrada en la base de datos.	
Flujo normal		

Paso	Actividades
1.	Buscar la ruta ingresando fecha de registro, siglas, denominación, clase o estado en la barra de búsqueda.
2.	Dar clic en el botón Editar.
3.	Editar el dato o datos de la ruta.
4.	Dar clic en el botón Guardar.
Flujo alternativo	
Paso	Actividades
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.
Postcondición	El dato o los datos de la ruta se actualizaron en la base de datos exitosamente.
Dependencias	RUT-001, SEC-001, SER-001

Tabla 3.29 Descripción del caso de uso Editar ruta ubicación

Descripción de caso de uso: Consultar ruta ubicación

Caso de Uso	Consultar datos de la ruta	Identificador: RUT-003
Actores	Analista de Logística	
Descripción	Consultar datos de la ruta	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Ruta ubicación. La ruta deberá estar registrada en la base de datos.	
Flujo normal		
Paso	Actividades	
1.	Buscar la ruta ingresando fecha de registro, siglas, denominación, clase o estado en la barra de búsqueda.	
2.	Dar clic en el botón Buscar	
3.	El registro buscado se muestra en la tabla	
Flujo alternativo		
Paso	Actividades	
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.	
Postcondición	La ruta buscada se muestra en la tabla.	
Dependencias	RUT-001	

Tabla 3.30 Descripción del caso de uso Consultar ruta ubicación

3. DIAGRAMAS DE CASOS DE USO: CATÁLOGO DE LOGISTICA

j) Programación base

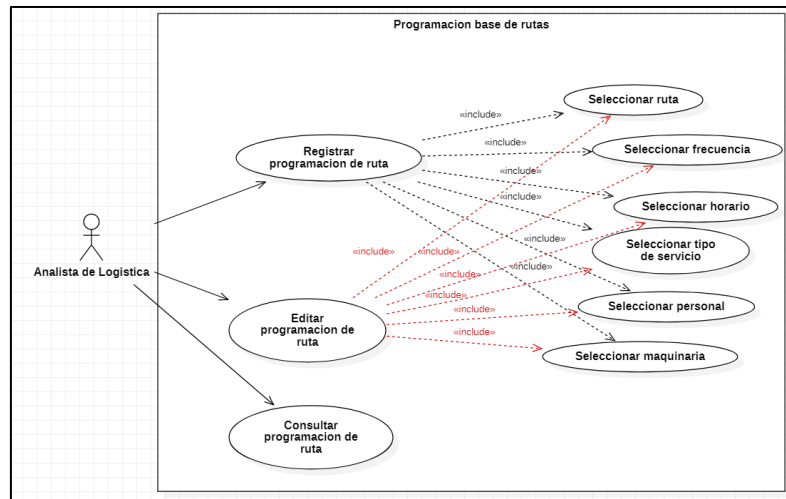


Figura 3.10 Caso de uso Programacion base

Descripción de caso de uso: Registrar programación de ruta

Caso de Uso	Registrar programación de ruta	Identificador: PRO-001
Actores	Analista de Logística	
Descripción	Registrar programación de ruta con los datos correspondientes.	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Programación de ruta. Debe estar registrada la frecuencia, horario y tipo de servicio a seleccionar. Debe estar registrada la ruta a seleccionar. Debe estar registrado el personal a seleccionar. Debe estar registrada la maquinaria a seleccionar.	
Flujo normal		
Paso	Actividades	
1.	Dar clic en el botón Nuevo.	
2.	Ingresar los datos correspondientes a la programación de ruta: ruta, frecuencia, horario, tipo de servicio, personal, maquinaria	
3.	Dar clic en el botón Guardar.	
Flujo alternativo		
Paso	Actividades	
2a.	El usuario Analista de Logística no llena todos los campos de la ruta. El sistema muestra un mensaje en donde pida al usuario llenar todos los campos de la ruta para que esta se pueda guardar en la base de datos.	
Postcondición	El registro se ha guardado correctamente en la base de datos.	
Dependencias	FRE-001, HOR-001, SER-001, RUT-001, PER-001, MAQ-001	

Tabla 3.31 Descripción del caso de uso Registrar programación de ruta

Descripción de caso de uso: Editar programación de ruta

Caso de Uso	Editar datos de programación de ruta	Identificador: PRO-002
Actores	Analista de Logística	
Descripción	Editar dato o datos de la programación de ruta	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Programación de ruta. La programación de ruta debe estar registrada en la base de datos.	
Flujo normal		

Paso	Actividades
1.	Buscar la programación de ruta ingresando código, siglas, descripción, servicio, frecuencia o estado en la barra de búsqueda.
2.	Dar clic en el botón Editar.
3.	Editar el dato o datos de la programación de ruta.
4.	Dar clic en el botón Guardar.
Flujo alternativo	
Paso	Actividades
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.
Postcondición	El dato o los datos de la programación de ruta se actualizaron en la base de datos exitosamente.
Dependencias	FRE-001, HOR-001, SER-001, RUT-001, PER-001, MAQ-001

Tabla 3.32 Descripción del caso de uso Editar programación de ruta

Descripción de caso de uso: Consultar programación de ruta

Caso de Uso	Consultar datos de la programación de ruta	Identificador: PRO-003
Actores	Analista de Logística	
Descripción	Consultar datos de la programación de ruta	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Programación de ruta. La programación de ruta deberá estar registrada en la base de datos.	
Flujo normal		
Paso	Actividades	
1.	Buscar la programación de ruta ingresando código, siglas, descripción, servicio, frecuencia o estado en la barra de búsqueda.	
2.	Dar clic en el botón Buscar	
3.	El registro buscado se muestra en la tabla	
Flujo alternativo		
Paso	Actividades	
1a.	El registro no existe en la base de datos. El sistema no muestra nada porque la búsqueda no arroja resultados.	
Postcondición	La programación de ruta buscada se muestra en la tabla.	
Dependencias	PRO-001	

Tabla 3.33 Descripción del caso de uso Consultar programación de ruta

k) Cambio de supervisor de ruta

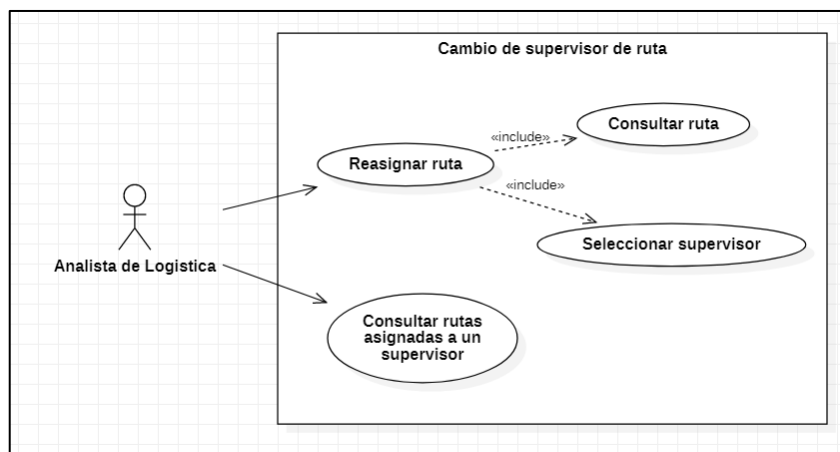


Figura 3.11 Caso de uso Cambio de supervisor de ruta

Descripción de caso de uso: Reasignar ruta

Caso de Uso	Reasignar ruta	Identificador: REA-001
Actores	Analista de Logística	
Descripción	Reasignar una ruta asignada previamente a un supervisor a otro supervisor diferente.	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Cambio de supervisor de ruta. Debe estar registrada la programación de ruta. Debe estar registrado el supervisor al que se le asignara la ruta.	
Flujo normal		
Paso	Actividades	
1.	Consultar el supervisor ingresando código, apellidos o nombres en la barra de búsqueda.	
2.	Dar clic en el botón Ver rutas del supervisor que se desea reasignar las rutas.	
3.	Se mostrarán todas las rutas asignadas a dicho supervisor.	
4.	Consultar la ruta a reasignar ingresando siglas de la ruta, estado, denominación, frecuencia o jornada en la barra de búsqueda.	
5.	Seleccionar el nombre del supervisor al que se le reasignara la ruta.	
6.	Dar clic en el botón Guardar.	
Flujo alternativo		
Paso	Actividades	
1a.	El supervisor no tiene ninguna ruta asignada por lo cual no se mostrará ningún resultado al realizar la búsqueda.	
4a.	La ruta que se desea reasignar aún no se ha programado. Debe programar primero la ruta en el submódulo de Programación base.	
4b.	La ruta que desea reasignar se ha programado a un supervisor diferente, por lo cual deberá regresar al submódulo de Programación base, buscar la ruta programada y ver el nombre del supervisor al cual se le asigno inicialmente.	
5a.	El nombre del supervisor al que desea asignar la ruta no aparece entre los supervisores registrados. Deberá registrar al supervisor a reasignar la ruta y después realizar el proceso de reasignación de ruta.	
Postcondición	El registro se ha actualizado en la base de datos.	
Dependencias	PRO-001, EMP-001	

Tabla 3.34 Descripción de caso de uso Reasignar ruta

Descripción de caso de uso: Consultar rutas asignadas a un supervisor

Caso de Uso	Consultar rutas asignadas a un supervisor	Identificador: REA-003
Actores	Analista de Logística	
Descripción	Consultar rutas asignadas a un supervisor.	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Cambio de supervisor de ruta.	
Flujo normal		
Paso	Actividades	
1.	Buscar el supervisor ingresando código, nombre o apellido en la barra de búsqueda.	
2.	Dar clic en el botón Ver rutas	
3.	Se mostrarán en una nueva ventana las rutas asignadas a dicho supervisor.	
Flujo alternativo		
Paso	Actividades	
3a.	No se mostrará ninguna ruta porque el supervisor aún no tiene rutas asignadas.	
Postcondición	Se muestran las rutas asignadas a un supervisor.	
Dependencias	PRO-001	

Tabla 3.35 Descripción de caso de uso Consultar rutas asignadas a un supervisor

l) Reporte de programación diaria

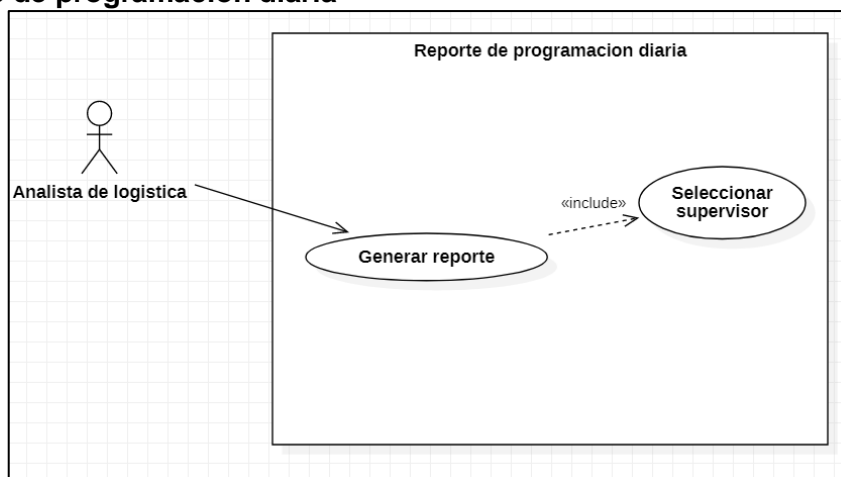


Figura 3.12 Caso de uso Reporte de programación diaria

Descripción de caso de uso: Generar reporte

Caso de Uso	Generar reporte	Identificador: REP-003
Actores	Analista de Logística	
Descripción	Generar reporte de programación de rutas por supervisor.	
Precondición	El usuario Analista de Logística debe estar registrado en el sistema y tener los permisos necesarios para ingresar al submódulo de Reporte de programación diaria.	
Flujo normal		
Paso	Actividades	
1.	Seleccionar la fecha desde la que se desea generar el reporte.	
2.	Seleccionar el supervisor del cual se desea generar el reporte	
3.	Dar clic en el botón Buscar.	
4.	Seleccionar el archivo y dar clic en el botón Descargar.	
Flujo alternativo		
Paso	Actividades	

2a.	El supervisor no está registrado en el sistema o no tiene ninguna ruta programada por lo cual no se muestra.
Postcondición	Se genero el reporte en formato pdf con la información solicitada.
Dependencias	PRO-001

Tabla 3.36 Descripción del caso de uso Generar reporte

e) Cronograma del proyecto

La Tabla 3.37 muestra el cronograma del proyecto, en donde se describen las actividades que se realizaran para la implementación del mismo, tomando en cuenta desde la definición y levantamiento de requerimientos, hasta el cierre del proyecto.

Con la estimación de horas realizada en el literal c) Product Backlog se ha dividido las tareas en diferentes sprints de similar duración. Cabe recalcar que un día equivale a ocho horas (duración de la jornada laboral en EMASEO EP). La estimación de tiempo y fechas de implementación del proyecto se muestra a continuación:

Actividad o Tarea	Duración horas	Duración días	Fecha de inicio	Fecha de fin	Entregable
Definición y levantamiento de requerimientos	32	4	04/05/2020	07/05/2020	Product Backlog, Documento de Visión, Casos de uso
Análisis de datos	24	3	08/05/2020	12/05/2020	
Software base	36	4 1/2	14/05/2020	20/05/2020	Prototipo de front end
Sprint 1	32	4	25/05/2020	28/05/2020	Incremento 1
Reunión de planificación del sprint 1	4	1/2			Sprint Backlog
Gestionar frecuencia	8	1			
Gestionar horario	8	1			
Gestionar tipo de servicios	8	1			
Reunión de revisión del sprint 1	4	1/2			
Sprint 2	56	7	01/06/2020	12/06/2020	Incremento 2
Reunión de planificación del sprint 2	4	1/2			Sprint Backlog
Gestionar talento humano	24	3			
Gestionar maquinaria	24	3			
Reunión de revisión del sprint 2	4	1/2			
Sprint 3	63	8	15/06/2020	24/06/2020	Incremento 3
Reunión de planificación el sprint 3	4	1/2			Sprint Backlog
Gestionar administraciones zonales	8	1			

Gestionar sectores	24	3			
Gestionar ruta ubicación	24	3			
Reunión de revisión del sprint 3	3	1/2			
Sprint 4	62	7 3/4	29/06/2020	08/07/2020	Incremento 4
Reunión de planificación del sprint 4	2	1/4			Sprint Backlog
Gestionar programación diaria de rutas asignando recursos	40	5			
Gestionar cambio de supervisor de ruta	16	2			
Reunión de revisión del sprint 4	4	1/2			
Sprint 5	12	1 1/2	13/07/2020	15/07/2020	Incremento 5
Reunión de planificación del sprint 5	2	1/4			Sprint Backlog
Generar reporte de programación	8	1			
Reunión de revisión del sprint 5	2	1/4			
Fase de pruebas	24	3	20/07/2020	22/07/2020	Documento Pruebas
Corrección de errores - versión beta	40	5	23/07/2020	29/07/2020	Incremento correcciones
Fase de pruebas	24	3	30/07/2020	03/08/2020	Documento Pruebas
Corrección de errores - versión reléase	40	5	04/08/2020	11/08/2020	Incremento correcciones
Reunión de presentación del Módulo de Logística	3	3/8	12/08/2020	12/08/2020	Manual de usuario
Cierre de ejecución del Módulo de Logística	3	3/8	14/08/2020	14/08/2020	Código fuente de logística
Total	475	59 1/2			Módulo de Logística

Tabla 3.37 Cronograma del proyecto

f) Arquitectura del proyecto

Para implementar el Sistema de Gestión de Operaciones se tomará como marco de referencia la Arquitectura SOA (Arquitectura orientada a servicios) debido a la gran cantidad de servicios que se deben integrar para desarrollar este sistema y el alto grado de reutilización de componentes que se empleará en el desarrollo del mismo.

A continuación, se describen las arquitecturas a utilizar:

Arquitectura SOA

SOA es un modelo de componentes que interrelaciona las diferentes unidades funcionales de las aplicaciones, denominadas servicios, a través de interfaces y contratos entre esos servicios. La interfaz se define de forma neutral, y debe ser independiente de la plataforma hardware, sistema operativo y del lenguaje de programación a utilizar, ya que esto permite a los servicios construidos sobre sistemas heterogéneos, interactuar entre ellos de manera uniforme y universal.

Esta arquitectura se basa en el concepto productor/consumidor, en ella la lógica de negocios o las funciones individuales son modularizadas y presentadas como servicios para ser consumidas por otras aplicaciones (clientes).

La estrategia de aplicación de la misma debe facilitar su integración y motivar la construcción de servicios, más que aplicaciones. Estos servicios se encargarían de exponer una funcionalidad bien definida a la aplicación que la requiera.

SOA propone una aplicación final que orquesta la ejecución de un conjunto de estos servicios, añade su lógica particular y le presenta una interfaz al usuario final.

Exponer procesos de negocio como servicios es la clave de la flexibilidad de esta arquitectura. Esto permite que otras piezas de funcionalidad (incluso también implementadas como servicios) hagan uso de otros servicios de manera natural, sin importar su ubicación física. Así un sistema evoluciona con la adición de nuevos servicios y con su mejora continua, donde cada servicio evoluciona de una manera independiente[27].

La Arquitectura Orientada a Servicios resultante, define los servicios de los cuales estará compuesto el sistema, sus interacciones, y con qué tecnologías serán implementados.

Tradicionalmente la arquitectura SOA se ha venido implementando a través de ESB (Enterprise Service Bus) de la siguiente forma:

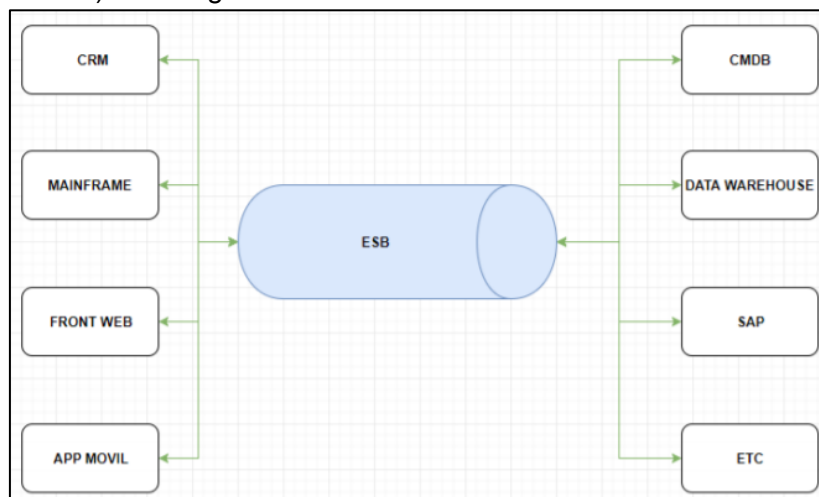


Figura 3.13 ESB (Enterprise Service Bus)

Dentro del marco ya creado, se ha visto la conveniencia de continuar utilizando la arquitectura basada en JEE (Java Enterprise Edition) que ha dado buenos resultados en el tipo de negocio de esta institución[27].

Arquitectura JEE

La arquitectura JEE tiene sus especificaciones y está formada de los componentes que se muestran en la Figura 3.14

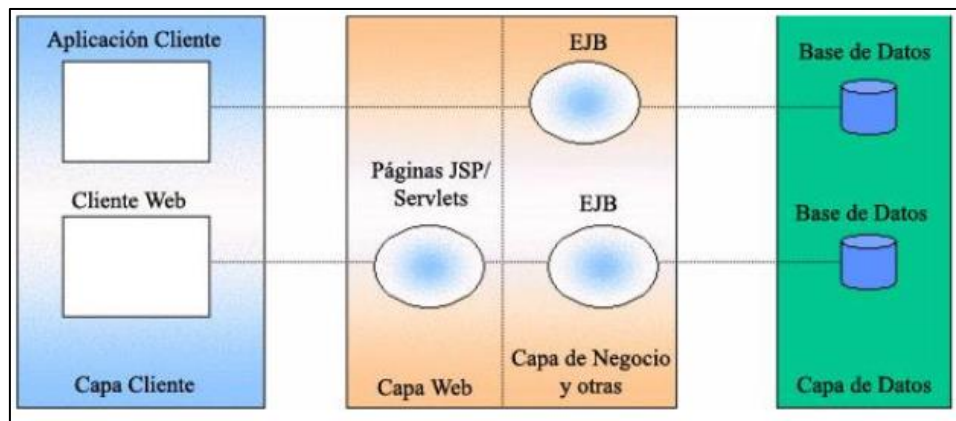


Figura 3.14 Arquitectura Java Enterprise Edition

Capa Cliente

Lo tradicional es utilizar un modelo MVC (Modelo, Vista, Controlador) para el desarrollo de las interfaces de usuario, aunque en algunos casos se puede obviar uno o dos de sus componentes.

- **Vista:** es lo que al usuario le permite e interactuar con las aplicaciones, utiliza el lenguaje HTML5.
- **Modelo:** es la abstracción de los datos con los cuales el controlador intercambia información con las vistas.
- **Controlador:** es el encargado de administrar el flujo de la información. Utiliza los datos del modelo para exponer u obtener información de las vistas, como una buena práctica de programación, desde aquí se deben hacer llamadas para consumir los servicios web.
- **Framework:** para facilitar el desarrollo de las interfaces de usuario, se usará el marco de desarrollo JSF con Primefaces, por ser open source y el estandar de java por defecto.

Capa web: es la capa de presentación y puede incluir:

- **Servlets:** Son clases java que procesan dinámicamente solicitudes (request) y construyen respuestas (responses).
- **Páginas JSP:** Son documentos basados en texto que se ejecutan como los servlets pero procesan de una manera más natural el contenido estático.

Capa de negocio y otras: Maneja la lógica de negocio a través de los componentes Enterprise Java Bean EJB.

- **Beans:** se implementan los beans de identidad, session y de interfaz. La comunicación entre estos puede ser mediante mensajes o protocolo REST. Estos componentes comprenden la lógica de negocio. La especificación JEE, permiten acceder a los datos mediante protocolos como Hibernate o JPA de forma transparente a la base de datos que se utilice.

Capa de datos: Compuesta por el software e infraestructura para manejar persistencia. Incluye sistemas manejadores de bases de datos relacionales.

- **Base de Datos:** la base de datos utilizada para todas las aplicaciones de la EMASEO EP deben ser sobre un motor de base que cumpla ACID (atomicidad, consistencia, aislamiento, durabilidad), La arquitectura deberá soportar Bases de Datos clustered.
- **Servidores de despliegue:** después de realizar un estudio de costo-beneficio, y considerando el desarrollo Open Source, se eligió al servidor Wildfly como servidor de aplicaciones, y es recomendable utilizar un servidor proxy para la liberación de la aplicación.

3.1.2. Planificación

a) Descripción de historias de usuario

En la Tabla 3.38 Descripción de historias de usuario se muestran las tareas expresadas en historias de usuario con su respectiva descripción con la finalidad de comprender mejor cada una de ellas.

Tarea	Historia de usuario			Descripción
	Como	Quiero	Para	
Gestionar frecuencia	Analista de logística	gestionar frecuencias	asignar una frecuencia a la ruta que voy a programar	Se deberá registrar, consultar, modificar frecuencias especificadas en el documento de diseño para asignar una frecuencia a la ruta que voy a programar
Gestionar horario	Analista de logística	gestionar horarios	asignar un horario a la ruta que voy a programar	Se deberá registrar, consultar, modificar horarios especificados en el documento de diseño para asignar un horario a la ruta que voy a programar
Gestionar tipos de servicio	Analista de logística	gestionar los tipos de servicio	asignar un tipo de servicio al sector que voy a crear	Se deberá registrar, consultar, modificar los tipos de servicios especificados en el documento de diseño con la finalidad de seleccionar el tipo de servicio que voy a brindar al sector para el cual voy a planificar la ruta
Gestionar talento humano	Analista de logística	gestionar talento humano (cargos y empleados)	asignar empleados a la ruta que voy a programar	Se deberá registrar, consultar y modificar cargos para asignar a los empleados que voy a registrar. Se deberá registrar, consultar y modificar empleados para asignarlos a la ruta que voy a programar.

Gestionar maquinaria	Analista de logística	gestionar maquinaria	asignar la maquinaria adecuada acorde al servicio que voy a prestar en la ruta que programare	Se deberá registrar, consultar, modificar maquinaria para asignar a la ruta que voy a programar.
Gestionar administración zonal	Analista de logística	gestionar las administraciones zonales de la ciudad de Quito	asignar la administración zonal al sector que voy a crear en caso de que así lo requiera.	Se deberá registrar, consultar, modificar y las administraciones zonales de la ciudad de Quito especificadas en el documento de diseño con la finalidad de seleccionar la administración zonal de un sector en caso de que lo requiera.
Gestionar sectores	Analista de logística	gestionar sectores	asignar un sector a la ruta que voy a programar	Se deberá registrar, consultar, modificar y eliminar sectores para asignarlos en las rutas que voy a programar.
Gestionar ruta ubicación	Analista de logística	gestionar rutas	posteriormente asignar los recursos necesarios para ejecutar las rutas programadas	Se deberá registrar, consultar y modificar rutas con sus datos correspondientes para asignar posteriormente los recursos para su ejecución.
Gestionar programación diaria de rutas asignando recursos	Analista de logística	gestionar programación de rutas	asignar los recursos necesarios para la ejecución de las mismas	Se deberá registrar, modificar y consultar rutas programadas.
Gestionar cambio de supervisor de ruta	Analista de logística	gestionar cambio de supervisor de ruta	reasignar una ruta programada a un supervisor distinto al asignado actualmente	Se mostrará la lista de supervisores con sus respectivas rutas programadas para poder reasignar una ruta a un distinto supervisor.
Generar reporte de programación	Analista de logística	generar un reporte de todas las rutas programadas por supervisor	lanzar la programación de la ruta	Se deberá generar un reporte con todas las rutas planificadas

Tabla 3.38 Historias de usuario

b) Priorización de historias de usuario

A continuación, se muestra la Tabla 3.39 en donde se encuentran las tareas ordenadas por prioridad, la cual se definió tomando en cuenta la dependencia entre las mismas.

ID	Tarea	Prioridad
PLA-001	Gestionar frecuencia	1
PLA-002	Gestionar horario	1
PLA-003	Gestionar tipos de servicio	1
PLA-004	Gestionar talento humano	1
PLA-005	Gestionar maquinaria	1
PLA-006	Gestionar administración zonal	1
PLA-007	Gestionar sectores	2
PLA-008	Gestionar ruta ubicación	2
PLA-009	Gestionar programación diaria de rutas asignando recursos	2
PLA-010	Gestionar cambio de supervisor de ruta	2
PLA-011	Generar reporte de programación	3

Tabla 3.39 Prioridad de Historias de Usuario

c) Reunión de planificación del sprint

Sprint 1

En el sprint 1 se implementarán las siguientes historias de usuario expresadas como tareas con las actividades necesarias para implementarlas:

ID	Tarea	Actividades
PLA -001	Gestionar frecuencia	<ol style="list-style-type: none"> 1. Registrar frecuencia con siglas, descripción, seleccionar días que esta contiene y estado. 2. Editar cualquiera de los datos de la frecuencia. 3. Consultar la frecuencia por siglas y días en las barras de búsqueda de la tabla. 4. Consultar la frecuencia por siglas y días en la barra de búsqueda de la interfaz.
PLA-002	Gestionar horario	<ol style="list-style-type: none"> 1. Registrar horario con siglas, descripción, hora y minuto iniciales, hora y minutos finales y estado. 2. Editar cualquiera de los datos del horario. 3. Consultar el horario por siglas y descripción en la barra de búsqueda de las tablas. 4. Consultar el horario por siglas y descripción en la barra de búsqueda de la interfaz.

PLA-003	Gestionar tipo de servicios	<ol style="list-style-type: none"> 1.Registrar tipo de servicio con siglas, denominación, descripción y check in en caso de necesitar vehículo. 2. Editar cualquiera de los datos del tipo de servicio. 3. Consultar tipo de servicio por siglas, denominación y descripción en la barra de búsqueda de la tabla. 4. Consultar tipo de servicio por siglas, denominación y descripción en la barra de búsqueda de la interfaz.
---------	-----------------------------	--

Tabla 3.40 Tareas Sprint 1

Para realizar un seguimiento de la implementación de las tareas durante el sprint se utilizará la herramienta Trello, el tablero de tareas se muestra en la Figura 3.15.

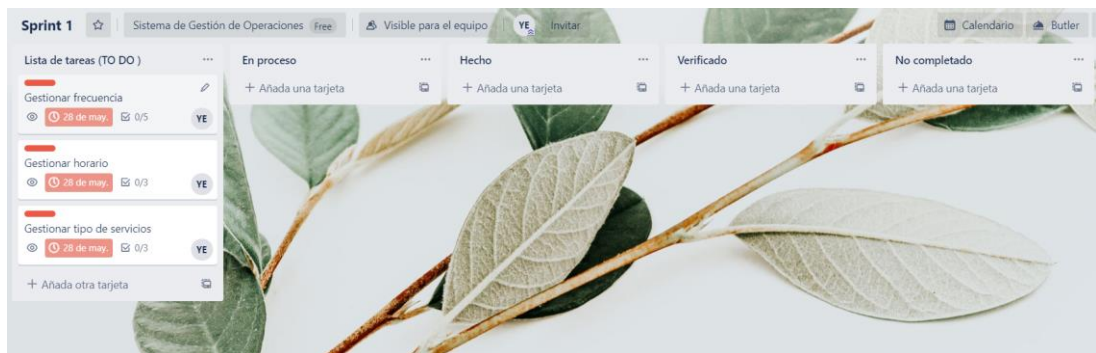


Figura 3.15 Tablero Sprint 1-Trello

Al dar clic sobre la tarea se mostrarán más detalles de la misma, por ejemplo, checklist de actividades, fecha de vencimiento, miembros responsables, archivos adjuntos, entre otras opciones que facilitarán su comprensión.

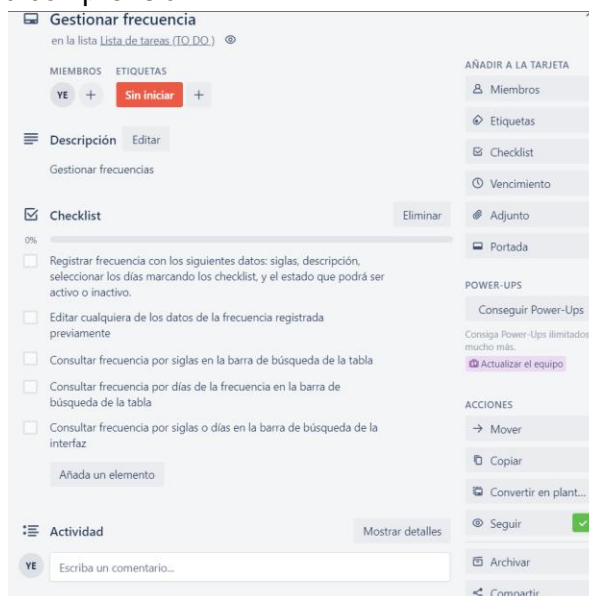


Figura 3.16 Detalles de tarea - Gestionar frecuencia

Sprint 2

En el sprint 2 se implementarán las siguientes historias de usuario expresadas como tareas con las actividades necesarias para implementarlas:

ID	Tarea	Actividades
PLA-004	Gestionar talento humano	<ol style="list-style-type: none"> 1.Registrar cargo con siglas, denominación, descripción. 2.Editar cualquiera de los datos del cargo. 3.Consultar el cargo por siglas, denominación y descripción en la barra de búsqueda de la tabla. 4.Consultar el cargo por siglas, denominación y descripción en la barra de búsqueda de la interfaz. 5.Integrar con CGWEB para obtener datos de los empleados de la empresa. 6.Registrar un empleado con código, identificación, apellido, nombre, relación laboral, actividad y seleccionar el cargo, rol y estado. 7. Editar cualquiera de los datos de un empleado. 8. Consultar datos del empleado ingresando identificación, apellido, nombre, cargo o rol en la barra de búsqueda de la tabla. 9.Consultar datos del empleado ingresando identificación, apellido, nombre, cargo o rol en la barra de búsqueda de la interfaz.
PLA-005	Gestionar maquinaria	<ol style="list-style-type: none"> 1.Integrar con D-SIES WEB para obtener datos de la maquinaria que hay en la empresa. 2. Registrar la maquinaria con disco, placa, marca, clase, modelo y seleccionar el tipo de maquinaria. 3. Editar cualquiera de los datos de la maquinaria. 4. Consultar la maquinaria ingresando disco, placa, marca, clase, modelo o tipo en la barra de búsqueda de la tabla. 5. Consultar maquinaria ingresando disco, placa, marca, clase, modelo o tipo en la barra de búsqueda de la interfaz.

Tabla 3.41 Tareas Sprint 2

Para realizar un seguimiento de la implementación de las tareas durante el sprint se utilizará la herramienta Trello, el tablero de tareas se muestra en la Figura 3.17.

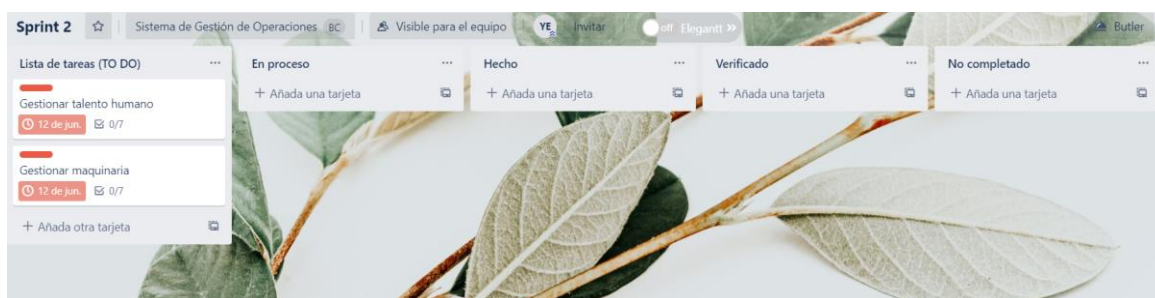


Figura 3.17 Tablero Sprint 2-Trello

Sprint 3

En el sprint 3 se implementarán las siguientes historias de usuario expresadas como tareas con las actividades necesarias para implementarlas:

ID	Tarea	Actividades
PLA-006	Gestionar administración zonal	<ol style="list-style-type: none"> 1. Registrar administración zonal con siglas, tipo y denominación. 2. Editar cualquiera de los datos de la administración zonal. 3. Consultar administración zonal ingresando siglas, denominación o tipo en la barra de búsqueda de la tabla. 4. Consultar administración zonal ingresando siglas, denominación o tipo en la barra de búsqueda de la interfaz.
PLA-007	Gestionar sectores	<ol style="list-style-type: none"> 1. Registrar sector con los siguientes datos: seleccionar administración zonal, tipo de sector e ingresar siglas, denominación y seleccionar estado. 2. Editar cualquiera de los datos del sector. 3. Integrar Open Street Maps en los formularios de registrar y editar sector. 4. Consultar sector ingresando siglas, denominación, tipo, siglas de región, denominación de región y estado en la barra de búsqueda de la tabla. 5. Consultar sector ingresando siglas, denominación, tipo, siglas de región,

		denominación de región y estado en la barra de búsqueda de la interfaz.
		6. Eliminar sector con sus datos correspondientes.
PLA-008	Gestionar ruta ubicación	<ol style="list-style-type: none"> 1. Registrar ruta con los siguientes datos: seleccionar servicio, ingresar siglas, descripción, fecha de registro, seleccionar estado, agregar sectores con su respectiva frecuencia. 2. Editar cualquiera de los datos de una ruta. 3. Integrar Open Street Maps en los formularios de registrar y editar ruta ubicación. 4. Consultar ruta ingresando siglas, fecha de registro, denominación, clase, estado en la barra de búsqueda de la tabla. 5. Consultar ruta ingresando siglas, fecha de registro, denominación, clase, estado en la barra de búsqueda de la interfaz.

Tabla 3.42 Tareas Sprint 3

Para realizar un seguimiento de la implementación de las tareas durante el sprint se utilizará la herramienta Trello, el tablero de tareas se muestra en la Figura 3.18.

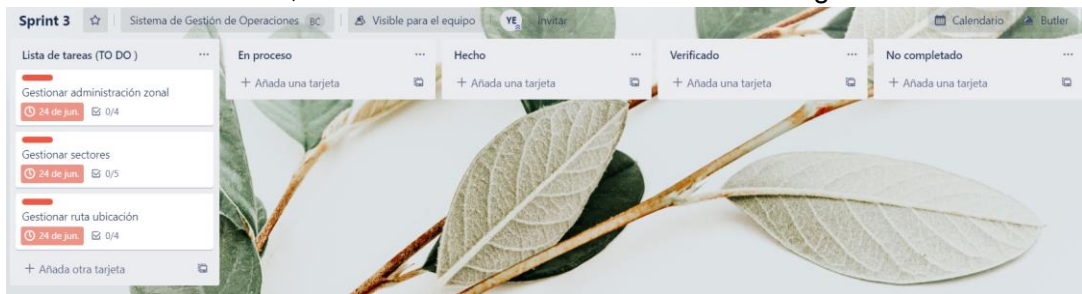


Figura 3.18 Tablero Sprint 3-Trello

Sprint 4

En el sprint 4 se implementarán las siguientes historias de usuario expresadas como tareas con las actividades necesarias para implementarlas:

ID	Tarea	Actividades
PLA-009	Gestionar programación diaria de	1. Registrar asignación de ruta con los siguientes datos: seleccionar ruta, frecuencia, horario, servicio,

	rutas asignando recursos	<p>ingresar fecha de registro, fecha de cierre y descripción y añadir personal y maquinaria.</p> <ol style="list-style-type: none"> 2. Editar cualquiera de los datos de la ruta asignada. 3. Consultar ruta asignada ingresando código, siglas, descripción o servicio en la barra de búsqueda de la tabla. 4. Consultar ruta asignada ingresando código, siglas, descripción, frecuencia o estado en la barra de búsqueda de la interfaz.
PLA-010	Gestionar cambio de supervisor de ruta	<ol style="list-style-type: none"> 1. Ver todas las rutas de un supervisor y reasignar a otro supervisor en caso de ser necesario seleccionando el supervisor al que se le reasignara la ruta. 2. Consultar supervisor por código, apellidos, nombres.

Tabla 3.43 Tareas Sprint 4

Para realizar un seguimiento de la implementación de las tareas durante el sprint se utilizará la herramienta Trello, el tablero de tareas se muestra en la Figura 3.19.

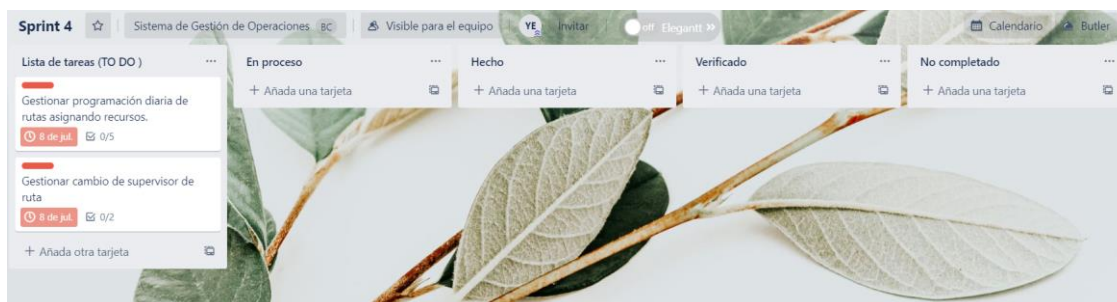


Figura 3.19 Tablero Sprint 4-Trello

Sprint 5

En el sprint 5 se implementarán la siguiente historia de usuario expresada como tarea con las actividades necesarias para implementarlas:

ID	Tarea	Actividades
PLA-011	Gestionar reporte de programación	1. Seleccionar supervisor y generar reporte de rutas programadas.

Tabla 3.44 Tareas Sprint 5

Para realizar un seguimiento de la implementación de las tareas durante el sprint se utilizará la herramienta Trello, el tablero de tareas se muestra en la Figura 3.20.

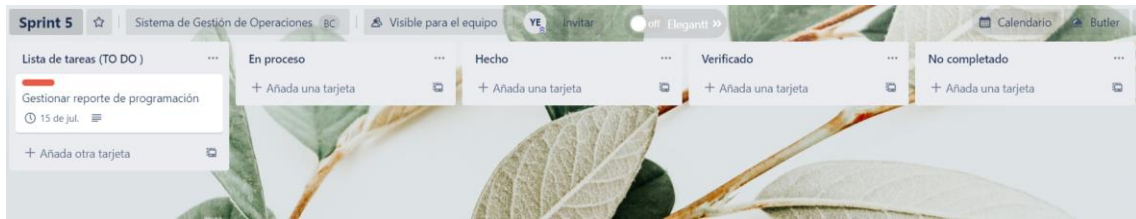


Figura 3.20 Tablero Sprint 5-Trello

Diagrama de clases del Módulo de Logística

A continuación, se muestra el diagrama de clases del Módulo de Logística del Sistema de Gestión de Operaciones con sus respectivas tablas y relaciones entre las mismas:

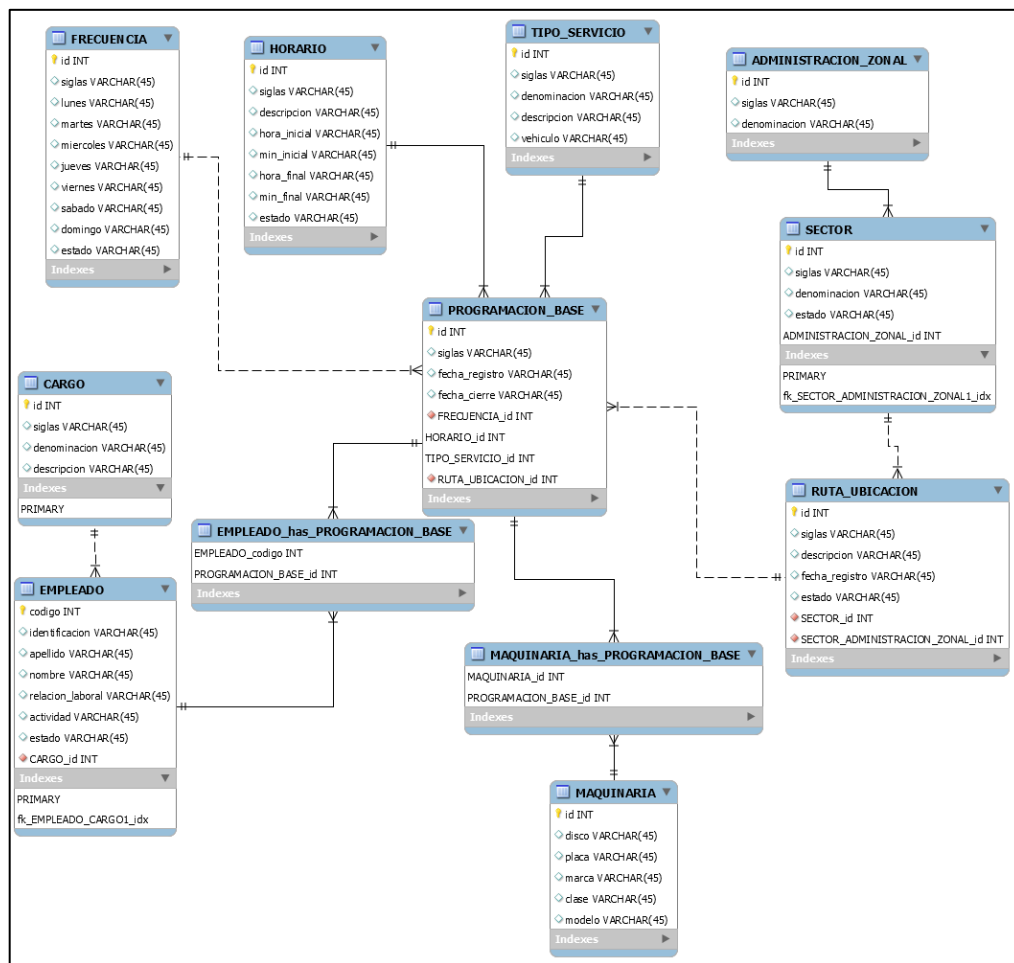


Figura 3.21 Diagrama de clases Módulo de Logística

3.1.3. Revisión de los sprints

a) Reunión de revisión del sprint

Sprint 1

En el sprint 1 se implementaron las siguientes historias de usuario:

ID Historia de usuario: PLA-001 Gestionar frecuencia

Incremento:

Listado de Tiempos

 Buscar por:

H < 1 2 > H 10 ▾

#	HORARIO SIGLAS	DÍAS	ESTADO	ACCIONES
19	L-M-I-J-V-S-D	lunes-martes-miércoles-jueves-viernes-sábado-domingo	Activo	<input type="button" value="Editar"/>
18	S-D	sábado-domingo	Activo	<input type="button" value="Editar"/>
4	D	domingo	Activo	<input type="button" value="Editar"/>
12	M-I-J-V-S	martes-miércoles-jueves-viernes-sábado	Activo	<input type="button" value="Editar"/>
1	L+V	lunes-miércoles-viernes	Activo	<input type="button" value="Editar"/>
2	L-M-I-J-V-S	lunes-martes-miércoles-jueves-viernes-sábado	Activo	<input type="button" value="Editar"/>
3	M-J-S	martes-jueves-sábado	Activo	<input type="button" value="Editar"/>
5	L	lunes	Activo	<input type="button" value="Editar"/>
6	M	martes	Activo	<input type="button" value="Editar"/>
7	I	miércoles	Activo	<input type="button" value="Editar"/>

H < 1 2 > H 10 ▾

Registrar frecuencia

Formulario de Tiempo para la Ruta

Horario Siglas	<input type="text"/>
Horario Descripción	<input type="text"/>
Domingo	<input type="checkbox"/>
Lunes	<input type="checkbox"/>
Martes	<input type="checkbox"/>
Miércoles	<input type="checkbox"/>
Jueves	<input type="checkbox"/>
Viernes	<input type="checkbox"/>
Sábado	<input type="checkbox"/>
Estado	No seleccionado .. ▾
<input type="button" value="Guardar"/>	<input type="button" value="Cancelar"/>

Editar datos de frecuencia

Formulario de Tiempo para la Ruta ✕

#

Horario Siglas

Horario Descripción

Domingo

Lunes

Martes

Miércoles

Jueves

Viernes

Sábado

Estado

Cumple criterios de aceptación	Si
Estado	Aprobado
Observaciones	Se implementaron todas las tareas definidas para la historia de usuario.

Tabla 3.45 PLA-001 Gestionar frecuencia

ID Historia de usuario: PLA-002
Gestionar horario

Incremento:

Nuevo
Buscar por:

#	FRECUENCIA SIGLAS	HORA INICIAL	HORA FINAL	DESCRIPCIÓN	ESTADO	ACCIONES
8	V2	16:00	23:00	Vespertino 2	Activo	Editar
9	N1	22:00	5:00	Nocturno 1	Activo	Editar
12	D4	6:30	11:30	Matutino 4	Activo	Editar
13	V5	15:00	20:00	Vespertino 5	Activo	Editar
16	V6	15:00	23:00	Vespertino 6	Activo	Editar
14	D6	6:00	14:00	Matutino 6	Activo	Editar
20	D8	8:00	16:00	Matutino 8	Activo	Editar
1	D	7:00	15:00	Matutino	Activo	Editar
2	V	14:00	21:00	Vespertino	Activo	Editar
3	N	19:00	3:00	Nocturno	Activo	Editar

Registrar horario

Formulario de Tiempo ✕

Frecuencia Siglas

Frecuencia Horario

Hora Inicial

Minutos

Hora Final

Minutos

Estado

Editar datos de horario

Formulario de Tiempo ✕

#

Frecuencia Siglas

Frecuencia Horario

Hora Inicial

Minutos

Hora Final

Minutos

Estado

Cumple criterios de aceptación	Si
Estado	Aprobado
Observaciones	Se implementaron todas las tareas definidas para la historia de usuario.

Tabla 3.46 PLA-002 Gestionar horario

ID Historia de usuario: PLA-003

Gestionar tipo de servicio

Incremento:

#	SERVICIO SIGLAS	SERVICIO DENOMINACIÓN	SERVICIO DESCRIPCIÓN	ACCIONES
16	RS	RECOLECCION CONTENORIZADA SOTERRADA	Recoleccion Contenerizada Soterrada	✎ Editar
23	SAN	SANITIZACION	Sanitizacion de avenidas	✎ Editar
14	EA	EQUIPO DE APOYO A LAS OPERACIONES	Equipo de Apoyo	✎ Editar
22	TCL	TRANSPORTE DE CONTENEDORES	Intercambio de contenedores en ruta	✎ Editar
18	LM	LIMPIEZA MERCADOS	Limpieza de mercados	✎ Editar
19	LC	LAVADO DE CONTENEDORES	Lavado der Contenedores	✎ Editar
11	RD	RECOLECCION DIFERENCIADA	Recoleccion Diferenciada	✎ Editar
12	BP	BARRIDO DE PARQUES	Barrido de parques emblematicos	✎ Editar
13	BS	BATERIAS SANITARIAS	Lavado de Bateria Sanitarias	✎ Editar
1	RPV	RECOLECCION A PIE DE VEREDA	Circuito dentro de un sector territorial especifico.	✎ Editar

Registrar tipo de servicio

Formulario de Servicio para la Ruta

Servicio Siglas:

Servicio Denominación:

Servicio Descripción:

Vehículo:

Guardar Cancelar

Editar datos de tipo de servicio

Formulario de Servicio para la Ruta

16

Servicio Siglas:

Servicio Denominación:

Servicio Descripción:

Vehículo:

Guardar Cancelar

Cumple criterios de aceptación	Si
Estado	Aprobado
Observaciones	Se implementaron todas las tareas definidas para la historia de usuario.

Tabla 3.47 PLA-003 Gestionar tipo de servicio

Resumen del sprint 1

Total de historias de usuario por implementar: 3

Total de historias de usuario implementadas y aprobadas: 3

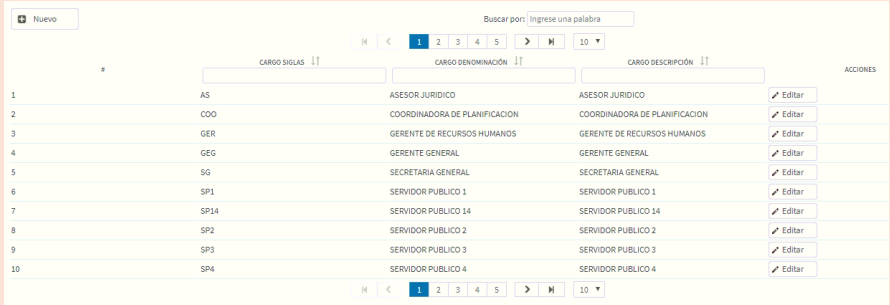
Total de historias de usuario pendientes: 0

Sprint 2

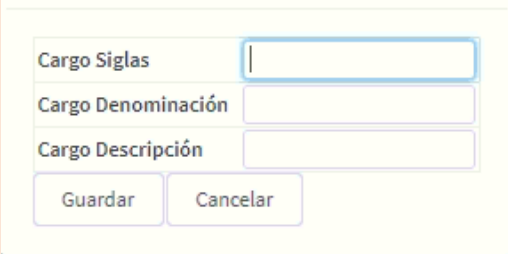
En el sprint 2 se implementaron las siguientes historias de usuario:

ID Historia de usuario: PLA-004 Gestionar talento humano-Cargos

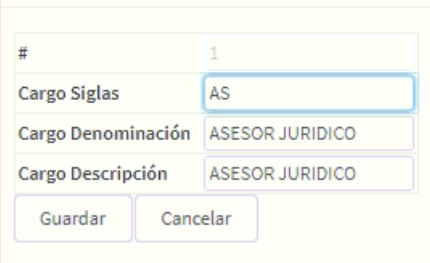
Incremento:



Registrar cargo



Editar datos del cargo



Cumple criterios de aceptación	Si
Estado	Aprobado

Observaciones

Se implementaron todas las tareas definidas para la historia de usuario.

Tabla 3.48 PLA-004 Gestionar talento humano-cargos

ID Historia de usuario: PLA-004 Gestionar talento humano-Empleados

Incremento:

Listado de Empleados

Buscar por:

CÓDIGO	EMPLEADO IDENTIFICACIÓN	APELLIDO	NOMBRE	CARGO	ROL	ACCIONES
711	1726404783	MORENO ENRIQUEZ	YADIRA PATRICIA	SERVIDOR PUBLICO 2	Supervisor	
712	1712230752	NARANJO ESTRELLA	RUTH GEOVANA	SERVIDOR PUBLICO DE APOYO 4	Ninguno	
713	1709301343	OCAÑA OÑATE	WILLAM HOMERO	SERVIDOR PUBLICO 3	Operador	
715	1712091295	OÑATE CARVAJAL	WILMAN ANIBAL	SERVIDOR PUBLICO 2	Supervisor	
729	1712859394	QUIROGA OROZCO	VERONICA LILIANA	SERVIDOR PUBLICO DE APOYO 4	Supervisor	
733	1711001766	ROBALINO GARCIA	RUTH ANGELICA	SERVIDOR PUBLICO 4	Supervisor	
736	1710370733	SIMBA CEVALLOS	JESUS CESAR	SERVIDOR PUBLICO 6	Supervisor	
740	1708738887	VASCO TONATO	MARIA LAURA	SERVIDOR PUBLICO DE APOYO 4	Supervisor	
741	1707392948	VASCONEZ FLORES	JOSE ALBERTO	SERVIDOR PUBLICO 2	Fiscalizador	
748	0913442406	ZABALA GARCIA	PATRICIA ESMERALDA	SERVIDOR PUBLICO 7	Consultas	

Registrar empleado

Formulario del Empleado

Cargo:

Rol:

Estado:

Editar datos del empleado

Formulario del Empleado ✕

#	39
Código	713
Empleado Identificación	1709301343
Apellido	OCAÑA OÑATE
Nombre	WILLAM HOMERO
Relación laboral	ADN
Empleado Actividad	Controlar el Funcionamie
Cargo	SP3 -> SERVIDOR PUBLICO 3
Rol	1 -> Operador
Estado	Activo

Cumple criterios de aceptación	Si
Estado	Aprobado
Observaciones	Se implementaron todas las tareas definidas para la historia de usuario.

Tabla 3.49 PLA-004 Gestionar talento humano-Empleados

ID Historia de usuario: PLA-005 Gestionar maquinaria

Incremento:

Listado de Maquinarias

 Buscar por:

#	DISCO	PLACA	MARCA	CLASE	MODELO	TIPO	ACCIONES
199	30-157	30-157	KENWORTH	RECOLECTOR CARGA POSTERIOR	2015	FORESTAL	<input type="button" value="Editar"/>
200	30-152	30-152	KENWORTH	RECOLECTOR CARGA POSTERIOR	2015	OCCIDENTAL	<input type="button" value="Editar"/>
201	30-153	30-153	KENWORTH	RECOLECTOR CARGA POSTERIOR	2015	OCCIDENTAL	<input type="button" value="Editar"/>
202	30-154	30-154	KENWORTH	RECOLECTOR CARGA POSTERIOR	2015	OCCIDENTAL	<input type="button" value="Editar"/>
203	30-155	30-155	KENWORTH	RECOLECTOR CARGA POSTERIOR	2015	OCCIDENTAL	<input type="button" value="Editar"/>
204	30-156	30-156	KENWORTH	RECOLECTOR CARGA POSTERIOR	2015	OCCIDENTAL	<input type="button" value="Editar"/>
205	30-158	30-158	KENWORTH	RECOLECTOR CARGA POSTERIOR	2015	OCCIDENTAL	<input type="button" value="Editar"/>
206	30-159	30-159	KENWORTH	RECOLECTOR CARGA POSTERIOR	2015	FORESTAL	<input type="button" value="Editar"/>
215	44-09	44-09	DAF	RECOLECTOR CARGA LATERAL	2016	OCCIDENTAL	<input type="button" value="Editar"/>
216	44-10	44-10	DAF	RECOLECTOR CARGA LATERAL	2016	OCCIDENTAL	<input type="button" value="Editar"/>

Registrar maquinaria

Disco	<input type="text"/>
Placa	<input type="text"/>
Marca	<input type="text"/>
Clase	<input type="text"/>
Modelo	<input type="text"/>
Tipo	No seleccionado .. ▾
<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>	

Editar datos de la maquinaria

#	199
Disco	30-157
Placa	30-157
Marca	KENWORTH
Clase	RECOLECTOR CARGA POS
Modelo	2015
Tipo	FOR -> FORESTAL ▾
<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>	

Cumple criterios de aceptación	Si
Estado	Aprobado
Observaciones	Se implementaron todas las tareas definidas para la historia de usuario.

Tabla 3.50 PLA-005 Gestionar maquinaria

Resumen del sprint 2

Total de historias de usuario por implementar: 2

Total de historias de usuario implementadas y aprobadas: 2

Total de historias de usuario pendientes: 0

Sprint 3

En el sprint 3 se implementaron las siguientes historias de usuario:

ID Historia de usuario: PLA-006

Gestionar administración zonal

Incremento:

Listado de Administraciones Zonales

#	SIGLAS	DENOMINACIÓN	TIPO	ACCIONES
11	SQ	SUR DE QUITO	REGION	Editar
1	CA	ADM. CALDERON	ZONAL	Editar
2	EA	ADM. ELOY ALFARO	ZONAL	Editar
3	EE	ADM. EUGENIO ESPEJO	ZONAL	Editar
4	LD	ADM. LA DELICIA	ZONAL	Editar
5	LC	ADM. LOS CHILLOS	ZONAL	Editar
6	MS	ADM. MANUELA SAENZ	ZONAL	Editar
10	NQ	NORTE DE QUITO	REGION	Editar
7	QU	ADM. QUITUMBE	ZONAL	Editar
8	TU	ADM. TUMBACO	ZONAL	Editar

Registrar administración zonal

Definición de la Administración Zonal

Tipo: No seleccionado ..

Denominación: []

Denominación: []

Guardar Cancelar

Editar datos de la administración zonal

Definición de la Administración Zonal

11

Tipo: Región

Denominación: SQ

Denominación: SUR DE QUITO

Guardar Cancelar

Cumple criterios de aceptación	Si
Estado	Aprobado
Observaciones	Se implementaron todas las tareas definidas para la historia de usuario.

Tabla 3.51 PLA-006 Gestionar administración zonal

Incremento:

Lista de sectores

#	SECTOR SIGLAS	DENOMINACIÓN	TIPO	REGIÓN SIGLAS	REGIÓN DENOMINACIÓN	ESTADO	ACCIONES
2122	EERMER15	CONTENERIZACION NOCTURNO EED1	geocerca	EE	ADM. EUGENIO ESPEJO	Activo	Editar Borrar
2008	EEBMER05	10 DE AGOSTO	geocerca	EE	ADM. EUGENIO ESPEJO	Activo	Editar Borrar
1056	BNRDA	10 DE AGOSTO	linea	NQ	NORTE DE QUITO	Activo	Editar Borrar
2113	EEBMER12	10 DE AGOSTO-EL INCA	geocerca	EE	ADM. EUGENIO ESPEJO	Activo	Editar Borrar
1235	HCOND01	10 De Agosto y Orellana	punto	DMQ1	DISTRITO METROPOLITANO	Activo	Editar Borrar
2460	CABMAL23	17 DE SEPTIEMBRE - LLANO CHICO	linea	CA	ADM. CALDERON	Activo	Editar Borrar
1294	HCOND36	24 de Mayo y Cuenca	punto	DMQ1	DISTRITO METROPOLITANO	Activo	Editar Borrar
1295	HCOND37	24 de Mayo y Libertad	punto	DMQ1	DISTRITO METROPOLITANO	Activo	Editar Borrar
1384	HOPN01	25 De Mayo Entre Culcocha y Lizardo Ruiz	punto	DMQ1	DISTRITO METROPOLITANO	Activo	Editar Borrar
1059	BNRNB	6 DE DICIEMBRE - ELOY ALFARO	linea	NQ	NORTE DE QUITO	Activo	Editar Borrar

Registrar sector

Definición del Sector

Adm. zonal: No seleccionado ...


Tipo sector: No seleccionado ...

Siglas:

Denominación:

Estado: No seleccionado ...

[Guardar](#) [Cancelar](#) [Validar Mapa](#)



Editar datos del sector

Definición del Sector

2122

Adm. zonal: EE -> ADM. EUGENIO ESPEJO

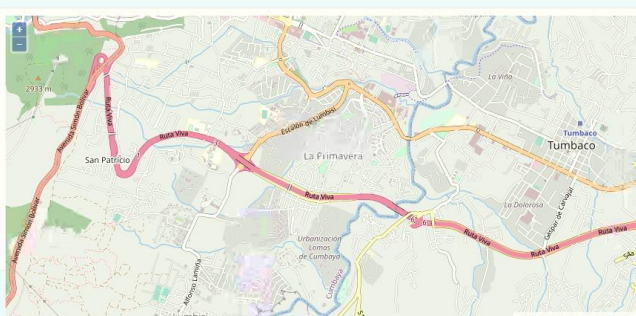
Tipo sector: G -> geocerca

Siglas: EERMER15

Denominación: CONTENERIZACION NOCTURNO EED1

Estado: Activo

[Guardar](#) [Cancelar](#) [Validar Mapa](#)



Cumple criterios de aceptación

Si

Estado

Aprobado

Observaciones	Se implementaron todas las tareas definidas para la historia de usuario.
----------------------	--

Tabla 3.52 PLA-007 Gestionar sectores

ID Historia de usuario: PLA-008 **Gestionar ruta ubicación**

Incremento:

Listado de Rutas

#	FECHA REGISTRO	SIGLAS DE LA RUTA	DENOMINACIÓN	CLASE	ESTADO	ACCIONES
2010	2020-06-17	PRUEBA I01		BARRIDO MANUAL	Activo	✎ Editar
2009	2020-06-17	PRUEBA XXXX		TRANSPORTE DE CONTENEDORES	Inactivo	✎ Editar
941	2017-04-01	EERMER15	CONTENERIZACION NOCTURNO EED1	sector	Inactivo	✎ Editar
90		QUNA2	GIRON SUR	sector	Inactivo	✎ Editar
547	2012-08-10	BNRDA	10 DE AGOSTO	Vía	Inactivo	✎ Editar
846	2016-08-29	EEBMER05	10 DE AGOSTO	sector	Inactivo	✎ Editar
932	2017-03-13	EEBMER12	10 DE AGOSTO-EL INCA	sector	Inactivo	✎ Editar
550	2012-08-10	BNRNB	6 DE DICIEMBRE - ELOY ALFARO	Vía	Inactivo	✎ Editar
234		CHRPVR20	6 DE DICIEMBRE - ONTANEDA	sector	Inactivo	✎ Editar
168		EENB20	ACACIAS	sector	Inactivo	✎ Editar


Registrar ruta ubicación

Definición de la Ruta

Servicio: No seleccionado ...
 Siglas:
 Descripción de la Ruta:
 Fecha Registro: 24/06/20
 Estado: No seleccionado ...
 Sectores disponibles: Secciones un sector ... D L M I J V S

Listado de Sectores

SECTOR	SIGLAS	SECTOR DENOMINACIÓN	FRECUENCIA	ACCIONES
No records found.				



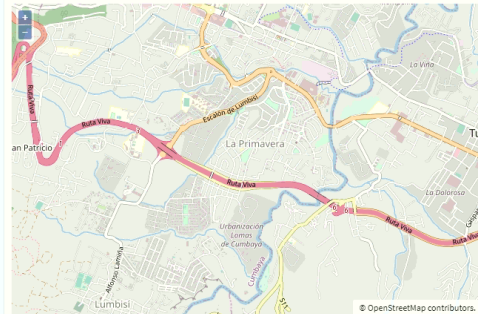
Editar datos de la ruta

Definición de la Ruta

2010
 Servicio BMA -> BARRIDO MANUAL
 Siglas PRUEBA I01
 Descripción de la Ruta:
 Fecha Registro: 17/06/20
 Estado: Activo
 Sectores disponibles: Secciones un sector ... D L M I J V S

Listado de Sectores

SECTOR	SIGLAS	SECTOR DENOMINACIÓN	FRECUENCIA	ACCIONES
BMEECEV1		BARRIDO MANUAL VISITA PAPANAL		✎ Editar ✕ Borrar
MINGAS		MINGAS DMQ		✎ Editar ✕ Borrar



Cumple criterios de aceptación	Si
Estado	Aprobado
Observaciones	Se implementaron todas las tareas definidas para la historia de usuario.

Tabla 3.53 PLA-008 Gestionar ruta ubicación

Resumen del sprint 3

Total de historias de usuario por implementar: 3

Total de historias de usuario implementadas y aprobadas: 3

Total de historias de usuario pendientes: 0

Sprint 4

En el sprint 4 se implementaron las siguientes historias de usuario:

ID Historia de usuario: PLA-009 Gestionar programación base

Incremento:

Listado Asignación de Rutas

Criterio de búsqueda: Ninguno Búsqueda:

código	SIGLAS	DESCRIPCIÓN	ADJONAL	SERVICIO	FRECUENCIA	HORARIO	ESTADO	RESPONSABLE	ACCIONES
000000001-PRG	P-EERPVR09	ATUCUCHO	EE	RECOLECCION A PIE DE VEREDA	L-V	Matutino	A	OCAÑA OÑATE	<input type="button" value="Modificar"/>
000000002-PRG	P-EEDA2	PRINCIPALES NORTE	EE	PUNTOS CRITICOS	L-M-J-V-S	Nocturno	A	CABRERA PINZA	<input type="button" value="Modificar"/>
000000003-PRG	P-EEDA3	EL BOSQUE					I	CABRERA PINZA	<input type="button" value="Modificar"/>
000000004-PRG	P-EERPVR29	PEDREGAL	EE	RECOLECCION A PIE DE VEREDA	L-V	Nocturno	A	OCAÑA OÑATE	<input type="button" value="Modificar"/>
000000005-PRG	P-EERPVR33	PRIMAVERA	EE	RECOLECCION A PIE DE VEREDA	L-V	Nocturno	A	OCAÑA OÑATE	<input type="button" value="Modificar"/>
000000006-PRG	P-EEDA6	SAN FERNANDO	EE	RECOLECCION A PIE DE VEREDA	L-V	Matutino	I	CABRERA PINZA	<input type="button" value="Modificar"/>
000000007-PRG	P-EEDA7	BAKER	EE	RECOLECCION A PIE DE VEREDA	D	Matutino 3	I	CABRERA PINZA	<input type="button" value="Modificar"/>
000000008-PRG	P-EEDA8	AJOYO NORTE	EE	RECOLECCION A PIE DE VEREDA	D	Nocturno	I	CABRERA PINZA	<input type="button" value="Modificar"/>
000000009-PRG	P-EERPVR31	SANTA ANITA	EE	RECOLECCION A PIE DE VEREDA	L-V	Nocturno	A	OCAÑA OÑATE	<input type="button" value="Modificar"/>
000000010-PRG	P-EERPVR07	MIRVALLE 1-2-3	EE	RECOLECCION A PIE DE VEREDA	L-V	Matutino	A	OCAÑA OÑATE	<input type="button" value="Modificar"/>

Registrar programación base de ruta

Formulario Asignación de Ruta

Código ruta: No seleccionado ... Clase: Programación Tipo: Descripción: Frecuencia: No seleccionado ... Horario: No seleccionado ... Servicio: No seleccionado ...
 Fecha Regla: Fecha Cierre: Descripción:

Maquinaria: No seleccionado ... + Agregar

Tabla de Maquinaria Registro

TIPO || DISCO || ACCIONES

No existen registros para el criterio de búsqueda

Empleado: No seleccionado ... + Agregar

Tabla de Empleados Registrados

CÓDIGO || APELLIDO || NOMBRE || CARGO || ACTIVO || ESTADO || RESP. || CO-RESP || ACCIONES

No existen registros para el criterio de búsqueda

Guardar ruta Cancelar

Editar datos de programación base

Formulario Asignación de Ruta

Código: 000000001-PRG Siglas: P-EERPVR09 Clase: Programación Tipo: sector Descripción: ATUCUCHO
 ruta: EERPVR09 Frecuencia: L-V Horario: D Servicio: RPV
 Fecha Regla: 2011-feb-01 Fecha Cierre: 2020-dic-31

Maquinaria: No seleccionado ... + Agregar

Tabla de Maquinaria Registro

TIPO || DISCO || ACCIONES

R-RECOLECTOR CARGA POSTERIOR 30-186

Empleado: No seleccionado ... + Agregar

Tabla de Empleados Registrados

CÓDIGO	APELLIDO	NOMBRE	CARGO	ACTIVO	ESTADO	RESP.	CO-RESP	SECTOR	ACCIONES PARA ASIGNAR SECTORES	ACCIONES
750	ACUÑA VILLAGOMEZ	DARIO JAVIER	CONDUCTOR	true	A	<input type="checkbox"/>	<input type="checkbox"/>		Cambiar X Quitar	X Eliminar
10378	JAMI QUILUMBA	EDISON JAVIER	OBRERO	true	A	<input type="checkbox"/>	<input type="checkbox"/>		Cambiar X Quitar	X Eliminar
12681	CANDO SANCHEZ	CRISTIAN DAVID	OBRERO	true	A	<input type="checkbox"/>	<input type="checkbox"/>		Cambiar X Quitar	X Eliminar
12763	RAMON ERAS	JORDY WILFRIDO	OBRERO	true	A	<input type="checkbox"/>	<input type="checkbox"/>		Cambiar X Quitar	X Eliminar
12296	MUZO SUQUILLO	ALEX DARIO	SUPERVISOR	true	A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Cambiar X Quitar	X Eliminar

Tabla de Micro rutas no asignadas

CÓDIGO MICRORUTA ATUCUCHO NOMBRE FRECUENCIAS

EERPVR09

Cumple criterios de aceptación	Si
Estado	Aprobado
Observaciones	Se implementaron todas las tareas definidas para la historia de usuario.

Tabla 3.54 PLA-009 Gestionar programación base de ruta

ID Historia de usuario: PLA-010

Gestionar cambio de supervisor de ruta

Incremento:

Reasignación de rutas : Lista de supervisores

CÓDIGO	APELLIDOS	NOMBRES	ESTADO	ACCIONES
28	ARCINIEGA PEREZ	ADOLFO WASHINGTON	Activo	Ver rutas
34	ARIAS PILAGUANO	LAURO MATEO	Activo	Ver rutas
1811	BALSECA RAMOS	DIEGO FERNANDO	Activo	Ver rutas
11229	BRAVO CHIMBOLEMA	MARCOS	Activo	Ver rutas
54	BUSTILLOS VILLACRES	MARIO NAPOLEON	Activo	Ver rutas
2392	CABRERA DAZA	TITO LIBIO	Activo	Ver rutas
10040	CARGUJA GUAÑUNA	ORLANDO JAVIER	Activo	Ver rutas
1701	CISNEROS BOLAÑOS	EDWIN WILFRIDO	Activo	Ver rutas
12059	COLLAGUAZO CARRERA	LUIS	Activo	Ver rutas
128	COLLAGUAZO SANGUÑA	CARLOS MARCELO	Activo	Ver rutas

Reasignación de rutas

Rutas pertenecientes a:

ARCINIEGA PEREZ ADOLFO WASHINGTON

RUTA	RUTASIGLA	ESTADO	DENOMINACIÓN	FRECUENCIA	JORNADA	SUPERVISOR
0000000436-PRG	P-BMQUDD02	I	BARRIDO MANUAL QUITUMBE SUR DIURNO DIARIO	lunes-martes-miércoles-jueves-viernes-sábado		28: ARCINIEGA PEREZ ADOLFO WASHINGTON
0000000439-PRG	P-BMQUDD04	I	BARRIDO MANUAL QUITUMBE ORIENTE DIURNO DIARIO	lunes-martes-miércoles-jueves-viernes-sábado		28: ARCINIEGA PEREZ ADOLFO WASHINGTON
0000000440-PRG	P-BMQUDD05	A	BARRIDO MANUAL QUITUMBE PARQUES DIURNO DIARIO	domingo		28: ARCINIEGA PEREZ ADOLFO WASHINGTON
0000000663-PRG	P-CQUINA1	I	QUITUMBE	lunes-miércoles-viernes		28: ARCINIEGA PEREZ ADOLFO WASHINGTON
0000001025-PRG	P-AEED01	I	CONTROL DE LLAVES DOMINGO	domingo	Matutino	28: ARCINIEGA PEREZ ADOLFO WASHINGTON
0000001179-PRG	P-LDRMER13	I	DOMINGO CONTENERIZACION LD	domingo	Matutino	28: ARCINIEGA PEREZ ADOLFO WASHINGTON
0000001186-PRG	P-LDBAAR05	I	DOMINGO BARRIDO MANUAL PARROQUIAS	domingo	Matutino	28: ARCINIEGA PEREZ ADOLFO WASHINGTON
0000001250-PRG	P-LDHIDR05	I	DOMINGO HIDROLAVADO LD	domingo	Matutino	28: ARCINIEGA PEREZ ADOLFO WASHINGTON
0000001251-PRG	P-LDRMER16	I	DOMINGO CONTENERIZACION LD	domingo	Matutino	28: ARCINIEGA PEREZ ADOLFO WASHINGTON
0000001310-PRG	P-QUIDA3	I	CIUDADELA IBARRA	lunes-martes-miércoles-jueves-viernes-sábado	Matutino	28: ARCINIEGA PEREZ ADOLFO WASHINGTON

Guardar Cancelar

Cumple criterios de aceptación	Si
Estado	Aprobado
Observaciones	Se implementaron todas las tareas definidas para la historia de usuario.

Tabla 3.55 PLA-010 Gestionar cambio de supervisor de ruta

Resumen del sprint 4

Total de historias de usuario por implementar: 2
 Total de historias de usuario implementadas y aprobadas: 2
 Total de historias de usuario pendientes: 0

Sprint 5

En el sprint 5 se implementó la siguiente historia de usuario:

ID Historia de usuario: PLA-011 **Reporte de programación diaria**

Incremento:	
<div style="border: 1px solid #ccc; padding: 10px;"> <p>Reporte Parte Diario</p> <p>Fecha de inicio: <input type="text" value="2020-ene-01"/> Supervisor: <input type="text" value="AGUIAR ANGULO ANGELICA ALEXANDRA"/> <input type="button" value="Buscar"/> <input type="button" value="pdf"/></p> <p>Pulse el botón buscar, para mostrar sus rutas</p> </div>	
Cumple criterios de aceptación	Si
Estado	Aprobado
Observaciones	Se implementaron todas las tareas definidas para la historia de usuario.

Tabla 3.56 PLA-011 Reporte de programación diaria

Resumen del sprint 5

Total de historias de usuario por implementar: 1

Total de historias de usuario implementadas y aprobadas: 1

Total de historias de usuario pendientes: 0

3.1.4. Implementación

a) Planificación de la implementación

- **Estándar de codificación**

El estándar de codificación que se utilizó para la implementación del Módulo de Logística del Sistema de Gestión de Operaciones fue lowerCamelCase.

- **Estándar de base de datos**

Respecto a la base de datos se utilizaron nombres y siglas para objetos de base de datos propios de la empresa.

b) Planificación de implementación de entregables

En este caso no se aplica la planificación de implementación de entregables porque todos los requerimientos se implementaron en los sprints correspondientes.

3.1.5. Cierre del proyecto

En el **ANEXO 3 Acta de Cierre del Proyecto** se describen los motivos de la culminación del proyecto, entregables aprobados y las firmas de aprobación de las partes interesadas.

3.2. Proceso de Aseguramiento de la calidad aplicado al Módulo de Logística

3.2.1. Evaluación de historias de usuario (Product Backlog)

En la Tabla 3.57 Evaluación del Product Backlog, se muestra la evaluación de cada una de las historias de usuario respecto al termino INVEST, además se muestra el número de criterios INVEST que cumple y el nivel de calidad acorde al número de criterios cumplidos.

Historia de usuario	Independiente	Negociable	Valorable	Estimable	Pequeña	Testeable	Nro criterios INVEST que cumple la HU	Nivel de calidad
Como analista de logística, quiero gestionar frecuencias, para asignar una frecuencia a la ruta que voy a programar	x	x	x	x	x	x	6	Alta
Como analista de logística, quiero gestionar horarios, para asignar un horario a la ruta que voy a programar	x	x	x	x	x	x	6	Alta
Como analista de logística, quiero gestionar tipos de servicios, para asignar un tipo de servicio a la ruta que voy a programar	x	x	x	x	x	x	6	Alta
Como analista de logística, quiero gestionar talento humano, para asignar empleados a la ruta que voy a programar	x	x	x	x		x	5	Alta
Como analista de logística, quiero gestionar maquinaria, para asignar la maquinaria adecuada acorde al servicio que voy a prestar en la ruta que voy a programar	x	x	x	x		x	5	Alta
Como analista de logística, quiero gestionar administraciones zonales, para asignar la administración zonal al sector que voy a crear.	x	x	x	x	x	x	6	Alta
Como analista de logística, quiero gestionar sectores, para asignar un sector a la ruta que voy a programar		x	x	x	x	x	5	Alta
Como analista de logística, quiero gestionar rutas, para posteriormente asignar los recursos necesarios para la ejecución de las mismas.		x	x	x		x	4	Media

Como analista de logística, quiero gestionar programación diaria, para asignar los recursos necesarios para la ejecución de las rutas.		x	x	x		x	4	Media
Como analista de logística, quiero gestionar cambio de supervisor de ruta, para reasignar una ruta programada a un supervisor distinto al asignado actualmente		x	x	x	x	x	5	Alta
Como analista de logística, quiero generar un reporte diario de todas las rutas programadas de un supervisor, para conocer indicadores de las rutas ejecutadas		x	x	x	x	x	5	Alta

Tabla 3.57 Evaluación del Product Backlog

3.2.2. Evaluación del Sprint Backlog

En la Tabla 3.58 Evaluación del Sprint Backlog se muestra la evaluación de cada una de las actividades asociadas a cada historia de usuario, las cuales se evaluarán respecto al termino SMART, además se muestra el número de criterios SMART que cada una de ellas cumple y el nivel de calidad asociado al número de criterios cumplidos.

ID historia de usuario	Actividades	Específico	Medible	Alcanzable	Relevante	A tiempo	Nro criterios SMART que cumple	Nivel de calidad
PLA-001	Registrar frecuencia de ruta	x	x	x	x	x	5	Alta
	Editar datos de frecuencia de ruta	x	x	x	x	x	5	Alta
	Consultar frecuencia de ruta	x	x	x	x	x	5	Alta
PLA-002	Registrar horario	x	x	x	x	x	5	Alta
	Editar datos de horario	x	x	x	x	x	5	Alta
	Consultar horario	x	x	x	x	x	5	Alta
PLA-003	Registrar tipo de servicio	x	x	x	x	x	5	Alta
	Editar datos de tipo de servicio	x	x	x	x	x	5	Alta
	Consultar tipo de servicio	x	x	x	x	x	5	Alta
PLA-004	Registrar cargo de empleado	x	x	x	x	x	5	Alta
	Editar cargo de empleado	x	x	x	x	x	5	Alta
	Consultar cargo de empleado	x	x	x	x	x	5	Alta
	Registrar empleado	x	x	x	x	x	5	Alta
	Editar datos de empleado	x	x	x	x	x	5	Alta
	Consultar empleado	x	x	x	x	x	5	Alta
PLA-005	Registrar maquinaria	x	x	x	x	x	5	Alta

	Editar datos de maquinaria	x	x	x	x	x	5	Alta
	Consultar maquinaria	x	x	x	x	x	5	Alta
PLA-006	Registrar administración zonal	x	x	x	x	x	5	Alta
	Editar datos de administración zonal	x	x	x	x	x	5	Alta
	Consultar administración zonal	x	x	x	x	x	5	Alta
PLA-007	Registrar sector	x	x	x	x	x	5	Alta
	Editar datos del sector	x	x	x	x	x	5	Alta
	Consultar sector	x	x	x	x	x	5	Alta
PLA-008	Registrar ruta ubicación	x	x	x	x	x	5	Alta
	Editar datos de ruta ubicación	x	x	x	x	x	5	Alta
	Consultar ruta ubicación	x	x	x	x	x	5	Alta
PLA-009	Registrar programación de ruta	x	x	x	x	x	5	Alta
	Editar datos de ruta programada	x	x	x	x	x	5	Alta
	Consultar ruta programada	x	x	x	x	x	5	Alta
PLA-010	Reasignar ruta a nuevo supervisor	x	x	x	x	x	5	Alta
	Consultar rutas asignadas a un supervisor	x	x	x	x	x	5	Alta
PLA-011	Generar reporte de programación diaria por supervisor	x	x	x	x	x	5	Alta

Tabla 3.58 Evaluación del Sprint Backlog

3.2.3. Evaluación de criterios de aceptación

En la Tabla 3.59 Evaluación de criterios de aceptación se muestra la evaluación a los criterios de aceptación de cada historia de usuario, los mismos que permiten identificar si una historia de usuario cumple con la funcionalidad descrita inicialmente. Estos criterios de aceptación se evaluarán en base a los parámetros: Atomicidad, No ambiguo, Verificable y Completo, además se muestra el número de parámetros con los que cumple el criterio de aceptación y su nivel de calidad.

ID	Historia de usuario	Criterios de aceptación	Atomicidad	No ambiguo	Verificable	Completo	Nro criterios cumple	Nivel de calidad
			x	x	x	x		
PLA-001	Como analista de logística, quiero gestionar frecuencias, para asignar una frecuencia a	Registrar una frecuencia con los siguientes datos: siglas, descripción, seleccionar uno o varios días (entre las opciones: lunes, martes, miércoles, jueves, viernes, sábado, domingo) y escoger un estado (entre las opciones: activo o inactivo).	x	x	x	x	4	Alta

	la ruta que voy a programar	Editar dato o datos de una frecuencia, entre los cuales se podrán editar: siglas, descripción, marcar o desmarcar uno o varios días(entre las opciones: lunes, martes, miércoles, jueves, viernes, sábado, domingo) y seleccionar el estado(entre las opciones: activo o inactivo)	x	x	x	x	4	Alta
		Listar las frecuencias registradas en una tabla con la opción de búsqueda en los campos: siglas y descripción.	x	x	x	x	4	Alta
		Consultar una frecuencia ingresando siglas o descripción en la barra de búsqueda.	x	x	x	x	4	Alta
PLA-002	Como analista de logística, quiero gestionar horarios, para asignar un horario a la ruta que voy a programar	Registrar un horario con los siguientes datos: siglas, descripción, hora inicial, minuto inicial, hora final, minuto final y estado (entre las opciones: activo o inactivo)	x	x	x	x	4	Alta
		Editar dato o datos de un horario, entre los cuales se pueden editar: siglas, descripción, hora inicial, minuto inicial, hora final, minuto final y estado (entre las opciones: activo o inactivo)	x	x	x	x	4	Alta
		Listar los horarios registrados en una tabla con la opción de búsqueda en los campos: siglas y descripción.	x	x	x	x	4	Alta
		Consultar un horario ingresando siglas o descripción en la barra de búsqueda	x	x	x	x	4	Alta
PLA-003	Como analista de logística, quiero gestionar tipos de servicios, para asignar un tipo de servicio a la ruta que voy a programar	Registrar un tipo de servicio con los siguientes datos: siglas, denominación, descripción y vehículo (si el servicio requiere vehículo marcar la casilla de verificación y si no dejarla desmarcada)	x	x	x	x	4	Alta
		Editar el dato o datos de un tipo de servicio, entre los cuales se pueden editar: siglas, denominación, descripción y vehículo (si el servicio requiere vehículo marcar la casilla de verificación y si no dejarla desmarcada)	x	x	x	x	4	Alta
		Listar los tipos de servicios en una tabla con la opción de búsqueda en los campos: siglas, denominación y descripción.	x	x	x	x	4	Alta

		Consultar un tipo de servicio ingresando: siglas, denominación o descripción en la barra de búsqueda	x	x	x	x	4	Alta
PLA-004	Como analista de logística, quiero gestionar talento humano, para asignar empleados a la ruta que voy a programar	Registrar cargo con los siguientes datos: siglas, denominación y descripción.	x	x	x	x	4	Alta
		Editar el dato o datos del cargo, entre los cuales se pueden editar: siglas, denominación o descripción.	x	x	x	x	4	Alta
		Listar los cargos en una tabla con la opción de búsqueda en los campos siglas, denominación y descripción	x	x	x	x	4	Alta
		Consultar el cargo ingresando: siglas, denominación o descripción en la barra de búsqueda.	x	x	x	x	4	Alta
		Registrar empleado con los siguientes datos: código, identificación, apellido, nombre, relación laboral, actividad, seleccionar el cargo (entre los cargos registrados anteriormente), seleccionar el rol(supervisor, administrador, operador de control, entre otros), seleccionar estado(entre las opciones: activo o inactivo)	x	x	x	x	4	Alta
		Editar dato o datos del empleado, entre los cuales se pueden editar: código, identificación, apellido, nombre, relación laboral, actividad, seleccionar cargo (entre los cargos registrados anteriormente), seleccionar rol(supervisor, administrador, operador), seleccionar estado(entre las opciones: activo o inactivo)	x	x	x	x	4	Alta
		Listar los empleados en una tabla con la opción de búsqueda en los campos: identificación, apellido, nombre, cargo y rol	x	x	x	x	4	Alta
		Consultar el empleado ingresando: identificación, apellido, nombre, cargo o rol en la barra de búsqueda.	x	x	x	x	4	Alta
PLA-005	Como analista de logística, quiero gestionar maquinaria, para asignar	Registrar tipo de maquinaria con los siguientes datos: siglas, denominación y descripción	x	x	x	x	4	Alta
		Editar dato o datos de tipo de maquinaria, entre los cuales se pueden editar: siglas, denominación o descripción.	x	x	x	x	4	Alta

	la maquinaria adecuada acorde al servicio que voy a prestar en la ruta que voy a programar	Listar los tipos de maquinaria en una tabla con la opción de búsqueda en los campos: siglas, denominación y descripción	x	x	x	x	4	Alta
		Consultar el tipo de maquinaria ingresando: siglas, denominación o descripción en la barra de búsqueda	x	x	x	x	4	Alta
		Registrar maquinaria con los siguientes datos: disco, placa, marca, clase, modelo y seleccionar tipo (entre los tipos de maquinaria registrados anteriormente).	x	x	x	x	4	Alta
		Editar dato o datos de la maquinaria, entre los cuales se podrá editar: disco, placa, marca, clase, modelo o seleccionar tipo.	x	x	x	x	4	Alta
		Listar la maquinaria en la tabla con opción de búsqueda en los campos: disco, placa, marca, clase, modelo y tipo (entre los tipos de maquinaria registrados anteriormente).	x	x	x	x	4	Alta
		Consultar la maquinaria ingresando: disco, placa, marca, clase, modelo o tipo en la barra de búsqueda.	x	x	x	x	4	Alta
		PLA-006	Como analista de logística, quiero gestionar administraciones zonales, para asignar la administración zonal al sector que voy a crear.	Registrar administración zonal con los siguientes datos: seleccionar tipo, denominación y descripción.	x	x	x	x
Editar dato o datos de la administración zonal, entre los cuales se podrá editar: selección de tipo, denominación o descripción.	x			x	x	x	4	Alta
Listar las administraciones zonales en una tabla con la opción de búsqueda en los campos: siglas, denominación y tipo.	x			x	x	x	4	Alta
Consultar la administración zonal ingresando: siglas, denominación o tipo en la barra de búsqueda.	x			x	x	x	4	Alta
PLA-007	Como analista de logística, quiero gestionar sectores, para asignar un sector a la ruta que voy a programar	Registrar sector con los siguientes datos: seleccionar administración zonal (entre las administraciones zonales registradas anteriormente), seleccionar tipo de sector, siglas, denominación, seleccionar estado(entre las opciones: activo o inactivo) y validar mapa.	x	x	x	x	4	Alta

		Editar dato o datos del sector, entre los cuales se podrá editar: selección de administración zonal(entre las administraciones zonales registradas anteriormente), selección del tipo de sector, siglas, denominación y selección de estado(entre las opciones: activo o inactivo) y validación de mapa	x	x	x	x	4	Alta
		Listar los sectores en una tabla con la opción de búsqueda en los campos: siglas, denominación, tipo, siglas de región, denominación de región y estado.	x	x	x	x	4	Alta
		Consultar el sector ingresando: siglas, denominación, tipo, siglas de región o denominación de región en la barra de búsqueda.	x	x	x	x	4	Alta
		Eliminar el sector con sus respectivos datos dando clic en el botón Borrar correspondiente al registro a borrar.	x	x	x	x	4	Alta
PLA-008	Como analista de logística, quiero gestionar rutas, para posteriormente asignar los recursos necesarios para la ejecución de las mismas.	Registrar la ruta ingresando los siguientes datos: seleccionar el tipo de servicio que voy a brindar (entre los tipos de servicio registrados anteriormente), ingresar las siglas y descripción de la ruta, seleccionar la fecha de registro, seleccionar el estado (entre las opciones: activo o inactivo) y agregar los sectores en donde se prestara el servicio seleccionado agregando la frecuencia de trabajo para cada uno de ellos.	x	x	x	x	4	Alta
		Editar dato o datos de la ruta, entre los cuales se podrá editar: selección del tipo de servicio (entre los tipos de servicio registrados anteriormente), siglas, descripción, fecha de registro, estado (entre las opciones: activo o inactivo), agregar o quitar sectores con sus respectivas frecuencias.	x	x	x	x	4	Alta
		Listar las rutas registradas en una tabla con la opción de búsqueda en los siguientes campos: fecha de registro, siglas, denominación, servicio y estado.	x	x	x	x	4	Alta

		Consultar la ruta ingresando: fecha de registro, siglas, denominación, servicio o estado en la barra de búsqueda.	x	x	x	x	4	Alta
PLA-009	Como analista de logística, quiero gestionar programación diaria, para asignar los recursos necesarios para la ejecución de las rutas.	Registrar la programación de ruta con los siguientes datos: seleccionar la ruta(entre las rutas registradas anteriormente), seleccionar frecuencia(entre las frecuencias registradas anteriormente), seleccionar horario(entre los horarios registrados anteriormente), seleccionar tipo de servicio(entre los tipos de servicios registrados anteriormente), seleccionar fecha de registro, seleccionar fecha de cierre, ingresar descripción, agregar la maquina adecuada para tipo de servicio a prestar(entre la maquinaria registrada anteriormente) y agregar el personal adecuado para tipo de servicio a prestar(entre el personal registrado anteriormente).	x	x	x	x	4	Alta
		Editar dato o datos de la ruta programada, entre los cuales se podrá editar: selección de ruta, selección de frecuencia, selección de horario, selección de tipo de servicio, selección de fecha de registro y fecha de cierre, descripción, agregar o quitar: maquinaria y personal.	x	x	x	x	4	Alta
		Listar las rutas programadas en una tabla con opción de búsqueda en los campos: código de ruta, siglas, estado, denominación, frecuencia y horario.	x	x	x	x	4	Alta
		Consultar la ruta programada ingresando: código, siglas, denominación, frecuencia o estado en la barra de búsqueda.	x	x	x	x	4	Alta
PLA-010	Como analista de logística, quiero gestionar cambio de supervisor de	Listar los supervisores en una tabla con los siguientes datos: código, apellido, nombre, estado y botón Ver rutas donde se podrá ver las rutas asignadas a dicho supervisor al dar clic sobre el mismo.	x	x	x	x	4	Alta

	ruta, para reasignar una ruta programada a un supervisor distinto al asignado actualmente	Reasignar una ruta a un supervisor diferente al asignado inicialmente, para lo cual, al abrir el cuadro de rutas asignadas a un supervisor dando clic en el botón Ver rutas , se podrá ver todas las rutas asignadas al mismo y la sobre la ruta que se desea reasignar se deberá seleccionar al supervisor a reasignar entre la lista de supervisores y luego dar clic en el botón Guardar .	x	x	x	x	4	Alta
PLA-011	Como analista de logística, quiero generar un reporte diario de todas las rutas programadas de un supervisor, para conocer indicadores de las rutas ejecutadas	Se deberá escoger la fecha desde la que se desea descargar el reporte, seleccionar el supervisor (entre los supervisores registrados anteriormente), dar clic en el botón Buscar y se desplegará la lista de rutas programadas, se podrá descargar el reporte en formato PDF y Hoja de cálculo Excel.	x	x	x	x	4	Alta

Tabla 3.59 Evaluación de criterios de aceptación

3.2.4. Evaluación del cumplimiento de metodología

En la Tabla 3.60 Evaluación del cumplimiento de la metodología se evaluó la aplicación de los criterios que definen la metodología de desarrollo de software en una escala de evaluación del 1 al 3, siendo: 1-> No se aplicó, 2-> Se aplico, pero no exitosamente, 3-> Se aplico exitosamente. A continuación, se muestra la puntuación obtenida por los evaluadores que en este caso fueron: miembro del equipo de desarrollo y Scrum Master, siendo los más indicados para realizar esta evaluación al estar durante todo el proceso de desarrollo del software.

CRITERIOS DE EVALUACION DE CUMPLIMIENTO DE METODOLOGIA	PUNTUACIÓN Miembro Scrum Team	PUNTUACIÓN Scrum Master
CRI-001 Definición de historias de usuario	3	3
CRI-002 Establecer prioridad a historias de usuario	3	3
CRI-003 Elaboración de diagramas de casos de uso	3	3
CRI-004 Cumplimiento de reunión de planificación	2	2
CRI-005 Definición de Sprint Backlog	2	2
CRI-006 Elaboración de diagramas de clases	3	3
CRI-007 Reuniones diarias	2	2

CRI-008 Entrega de incremento	3	3
CRI-009 Cumplimiento de reunión de revisión	3	2
Sumatoria total de la puntuación criterios	24	23
Promedio entre la sumatoria de la puntuación de los evaluadores	23.5	
Nivel de calidad obtenido acorde al promedio entre la sumatoria de los puntajes de los evaluadores	2.61 -> Alto	

Tabla 3.60 Evaluación del cumplimiento de la metodología

3.2.5. Evaluación del nivel de compromiso de las partes interesadas

En la Tabla 3.61 Evaluación del nivel de compromiso de las partes interesadas se evaluó el nivel de compromiso de todos los involucrados en el proyecto. A continuación, se muestran los puntajes obtenidos, los cuales servirán de retroalimentación dentro del equipo para mejorar las capacidades y ver las debilidades en las que se pueda trabajar en futuros proyectos o en los próximos módulos a desarrollar.

Criterios por evaluar	Product Owner	Scrum Master	Scrum Team	
			Miembro 1	Miembro 2
Nivel de compromiso en el desarrollo del proyecto (PO, SM y ST)	2	3	3	2
Nivel de conocimiento del negocio (SM y PO)	2	3	3	2
Nivel de cumplimiento de responsabilidades asignadas (PO, SM y ST)	2	3	3	3
Nivel de conocimiento técnico en las tecnologías a utilizar (Miembros del ST)	N/A	2	3	2
Capacidad de trabajar en equipo (PO, SM y ST)	3	3	3	3
Promedio del nivel de compromiso	2.25	2.8	3	2.4

Tabla 3.61 Evaluación del nivel de compromiso de las partes interesadas

3.2.6. Evaluación del cumplimiento de estándares

En la Tabla 3.62 Evaluación del cumplimiento de estándares se muestra la puntuación obtenida de la evaluación realizada por los miembros del Scrum Team sobre la aplicación de estándares. Se califico de con los términos: Cumple, No cumple o N/A.

Estándar a evaluar	Scrum Team		
	Scrum Master	Miembro 1	Miembro 2
lower camelcase (Codificación)	Cumple	Cumple	Cumple
Nombres propios para objetos definidos por la empresa (Base de datos)	Cumple	Cumple	Cumple

Tabla 3.62 Evaluación del cumplimiento de estándares

3.2.7. Plan de pruebas

El plan de pruebas de software del Módulo de Logística se encuentra en el **ANEXO 4 Plan de Pruebas**; en este enunciado se mostrarán los puntos más importantes del mismo para su respectivo análisis:

1. Alcance de las pruebas

a) Elementos por probar

En la primera etapa de desarrollo del Sistema de Gestión de Operaciones se ejecutarán las pruebas sobre el Módulo de Logística, que está compuesto por los siguientes submódulos:

- Frecuencia de ruta
- Horario de ruta
- Tipo de servicio
- Cargos de empleado-Talento humano
- Empleados-Talento humano
- Tipo de maquinaria-Maquinaria
- Maquinaria
- Administración zonal
- Sectores
- Ruta ubicación
- Programación base
- Cambio de supervisor de ruta
- Reporte de programación diaria

b) Enfoque de pruebas

Los tipos de pruebas a realizarse sobre el Módulo de Logística son los siguientes:

- **Pruebas de humo:** se probará que las funcionalidades más significativas del sistema de software funcionen correctamente; las funcionalidades principales son:
 - ✓ Registro de datos
 - ✓ Edición de datos
 - ✓ Consulta de datos

✓ Listar datos

Se deberá comprobar que cada una de ellas funcionen correctamente para todos los submódulos del Módulo de Logística.

- **Pruebas de carga:** se probará el comportamiento de cada uno de los submódulos de Logística bajo cierta cantidad de peticiones (1, 10, 20 y 50); esta carga de peticiones simulará desde el número mínimo de usuarios esperado, hasta el número máximo esperado de peticiones.

Esta prueba mostrará los tiempos de respuesta ante el número de peticiones aplicada a los submódulos y por ende el rendimiento de cada submódulo ante dicha carga.

- **Pruebas de usabilidad:** se probarán los submódulos del Módulo de Logística por un usuario que conoce el sistema a perfección, pero que no fue parte del desarrollo del mismo y usuarios de experiencia media, previa capacitación de los mismos sobre el Módulo de Logística; además se harán encuestas acerca de la experiencia que tuvieron para considerar posibles mejoras.
- **Pruebas de aceptación de usuario:** una vez que se hayan implementado las pruebas requeridas para el sistema de software, se ejecutarán las pruebas de aceptación por parte de los usuarios finales que manejarán el Módulo de Logística y, se considerarán las observaciones que tuvieron respecto al mismo, para discutir con el Scrum Team, Scrum Master y Product Owner si es factible realizar mejoras.

2. Herramientas de pruebas requeridas

JMeter

Es un software de código abierto diseñado en Java para verificar el comportamiento funcional y rendimiento de una página web o sistema de software alojado localmente. Originalmente se diseñó para probar aplicaciones web, pero desde entonces sus funcionalidades se han extendido.

Apache JMeter se puede utilizar para probar el rendimiento tanto en recursos estáticos como dinámicos y aplicaciones web dinámicas.

Se puede simular cargas pesadas de usuarios en un servidor, grupo de servidores, red para probar su fuerza o analizar el rendimiento general bajo diferentes tipos de carga.

3. Matriz de responsabilidades

A continuación, se muestra la matriz de responsabilidades RACI con la descripción de todas las actividades que se llevarán a cabo para poder ejecutar el plan de pruebas y los roles que asumirán los participantes en el desarrollo del Módulo de Logística.

Tareas	Personas				
	QA Tester	Product Owner	Scrum Master	Scrum team member 1	Scrum team member 2
Escribir el plan de pruebas	R	A	A	C	I
Escribir los casos de prueba (descripción de casos de uso)	R	A	A	C	I
Escribir los criterios de aceptación	R	A	A	C	I

Ejecutar pruebas de carga	R	A	A	C	I
Ejecutar pruebas de aceptación	I	A	R	R	I
Ejecutar pruebas de humo	R	A	A	R	R
Ejecutar pruebas de usabilidad	R	A	A	R	R

Tabla 3.63 Matriz RACI



Pruebas de rendimiento

A continuación, se muestran los resultados obtenidos en las pruebas de rendimiento que se ejecutaron para todos los submódulos del Módulo de Logística, se evaluó el rendimiento del sistema de software ante cargas de 1, 10, 20 y 50 usuarios.

Los resultados obtenidos respecto a tiempo se muestran en milisegundos a excepción del rendimiento.

Número de usuarios: 1

Submódulo	usuarios	Tiempo min	Tiempo máx	Media	%Error	Rendimiento
Login	1	5	186	46	0 %	19,4/sec
Frecuencia	1	7	148	24	0 %	33,2/sec
Horario	1	48	480	138	0 %	7,2/sec
Tipo de servicio	1	362	3059	864	0 %	1,2/sec
Cargo	1	711	1962	1140	0 %	52,6/min
Empleados	1	618	3410	1598	0 %	37,5/min
Tipo de maquinaria	1	7	57	24	0 %	23,0/sec
Maquinaria	1	7	30	13	0 %	43,7/sec
Administración zonal	1	825	17872	4893	0 %	12,3/min
Sector	1	46	11920	1412	0 %	42,4/min
Ruta ubicación	1	9	79	24	0 %	13,8/sec
Programación diaria	1	675	8200	2043	0 %	29,3/min
Cambio supervisor ruta	1	64	1790	930	0 %	1,1/sec
Reporte programación	1	391	42677	5611	0 %	10,7/min

Tabla 3.64 Resultados pruebas rendimiento 1 usuario

En la Tabla 3.64 se puede observar que la media más alta pertenece al submódulo de **Reporte programación**, el cual tardaría un aproximado de 5 s en responder ante la petición de un usuario, esto puede variar dependiendo de la cantidad de información que

un usuario solicita visualizar en el reporte a generar. También se puede observar que la media de los submódulos de **Cargo, Empleados, Sector, Administración zonal y Programación diaria** está entre 1 y 5 s, un tiempo considerable para la gran cantidad de información que se manejan en dichos submódulos, considerando que en cada uno de ellos a excepción de Login, Reporte programación y Cambio supervisor de ruta, se puso a prueba el rendimiento al hacer operaciones de registro, listado, consulta y edición de datos.

Número de usuarios: 10

Submódulo	usuarios	Tiempo min	Tiempo máx	Media	% Error	Rendimiento
Login	10	4	539	38	0 %	57,0/sec
Frecuencia	10	6	150	22	0 %	113,0/sec
Horario	10	73	916	212	0 %	40,8/sec
Tipo de servicio	10	659	6507	1111	0 %	7,1/sec
Cargo	10	54	43093	2787	0 %	1,8/sec
Empleados	10	75	43961	3016	0,91 %	2,3/sec
Tipo de maquinaria	10	8	200	27	0 %	63,9/sec
Maquinaria	10	9	202	30	0 %	76,1/sec
Administración zonal	10	50	37190	3922	0 %	1,6/sec
Sector	10	48	59239	1523	0,53 %	6,1/sec
Ruta ubicación	10	6	417	70	0 %	94,0/sec
Programación diaria	10	59	44438	2301	0 %	3,7/sec
Cambio supervisor ruta	10	36	641	186	0 %	26,0/sec
Reporte programación	10	551	43225	3656	0 %	1,6/sec

Tabla 3.65 Resultados pruebas rendimiento 10 usuarios

En la Tabla 3.65 se puede observar que los submódulos de **Empleados, Administración zonal y Reporte programación** tienen las medias más altas que van entre 3 y 4 s, lo cual muestra un buen rendimiento para un número de 10 usuarios ya que el tiempo de respuesta es muy pequeño.

Los submódulos **de Empleados y Sector** muestran un porcentaje de error muy bajo 0,091 y 0,053 respectivamente, lo cual representa una probabilidad muy pequeña de fallo menor a 1 petición.

Número de usuarios: 20

Submódulo	usuarios	Tiempo min	Tiempo máx	Media	%Error	Rendimiento
Login	20	4	775	67	0 %	79,7/sec
Frecuencia	20	5	171	22	0 %	217,2/sec
Horario	20	105	1690	320	0 %	54,7/ sec
Tipo de servicio	20	610	7463	1254	0 %	12 /sec
Cargo	20	594	35619	2547	0 %	3,4 sec
Empleados	20	63	52090	3938	0 %	3,1 sec
Tipo de maquinaria	20	7	159	28	0 %	110,9/ sec
Maquinaria	20	9	215	34	0 %	147,6 /sec

Administración zonal	20	58	58204	4188	0 %	2,4 sec
Ruta ubicación	20	6	532	91	0 %	155,8 /sec
Programación diaria	20	250	47615	2383	0,20 %	7,5 sec
Cambio supervisor ruta	20	31	830	257	0 %	34,6 sec
Reporte programación	20	50	110059	5329	0 %	1,5 sec

Tabla 3.66 Resultados pruebas rendimiento 20 usuarios

En la Tabla 3.66 se puede observar que los submódulos de **Empleados**, **Administración zonal** y **Reporte programación** tienen las medias más altas que van entre los 3 y 5 segundos, se puede observar que en los anteriores resultados para 10 usuarios los mismos submódulos mostraban los tiempos medios más altos.

El submódulo de **Programación diaria** muestra un porcentaje de error muy bajo de 0,04 lo cual demuestra una posibilidad de fallo muy pequeña menor a 1 petición.

Número de usuarios: 50

Submódulo	usuarios	Tiempo min	Tiempo máx	Media	%Error	Rendimiento
Login	50	4	678	88	0 %	172,4 /sec
Frecuencia	50	5	207	23	0 %	516,7/ sec
Horario	50	79	6558	991	0 %	45,3 /sec
Tipo de servicio	50	49	72251	2221	0 %	7,3 sec
Cargo	50	50	61923	2971	0,17 %	7,5 sec
Tipo de maquinaria	50	7	261	45	0 %	255,3 /sec
Maquinaria	50	6	454	55	0 %	273,2 /sec
Ruta ubicación	50	6	2877	273	0 %	153,6/ sec
Programación diaria	50	1	69689	2982	0,24 %	13,1 sec
Cambio supervisor ruta	50	33	823	257	0 %	73,8 /sec
Reporte programación	50	0	68385	5366	1,50 %	5,3 sec

Tabla 3.67 Resultados pruebas rendimiento 50 usuarios

En la Tabla 3.67 se puede observar que los submódulos **Reporte programación**, **Programación diaria**, **Cargo** y **Tipo de servicio** tienen las medias más altas que están entre 2 y 5 segundos, los cuales son tiempos bastante aceptables para una carga de 50 usuarios conectados simultáneamente. También se puede ver que los submódulos **Cargo**, **Programación diaria** y **Reporte programación** tienen porcentajes de error muy insignificantes de 0.085, 0.12 y 0.75 respectivamente, menores a 1 lo cual se traduce en una tasa de error menor a 1 petición.

Pruebas de rendimiento con tiempo aleatorio agregado entre peticiones de 5 s

Se practicaron pruebas de rendimiento para todos los submódulos del Módulo de Logística, con número de peticiones que van desde 1 a 50. Además, se les agregó un tiempo aleatorio de 5 segundos entre peticiones, el cual puede ser como máximo 5 s y como mínimo 0, es decir, no agregarse nada entre peticiones.

Número de usuarios: 1

Submódulo	usuarios	Tiempo min	Tiempo máx	Media	%Error	Rendimiento
Frecuencia	1	8	137	23	0 %	28,5 / min
Horario	1	72	60116	4590	0 %	8,3 min
Tipo de servicio	1	805	9245	2396	0 %	14,2 min
Cargo	1	592	7209	2025	0 %	13,4 min
Tipo de maquinaria	1	8	56	17	0 %	24,4/min
Maquinaria	1	8	22	11	0 %	22,1/min
Ruta ubicación	1	7	60	13	0 %	20,9/min
Cambio supervisor ruta	1	8	20	12	0 %	28,4 / min

Tabla 3.68 Resultados pruebas rendimiento 1 usuario

En la Tabla 3.68 se puede ver que los submódulos **Horario**, **Tipo de servicio** y **Cargo** tienen las medias de tiempo más altas que van desde 2 a 4 segundos, sin embargo, el porcentaje de error es 0 % lo cual significa que todas las peticiones se ejecutaron correctamente sin errores.

Número de usuarios: 10

Submódulo	usuarios	Tiempo min	Tiempo máx	Media	%Error	Rendimiento
Login	10	5	566	53	0 %	3,3 / s
Frecuencia	10	6	135	18	0 %	2,7/s
Horario	10	69	79482	4515	0 %	1,1 s
Tipo de servicio	10	96	47626	3736	0 %	1,3 s
Cargo	10	334	25371	3115	0 %	1,6 s
Tipo de maquinaria	10	7	68	16	0 %	3,2/ s
Maquinaria	10	6	253	14	0 %	3,2/s
Ruta ubicación	10	6	158	14	0 %	3,6/s
Cambio supervisor ruta	10	7	406	37	0 %	2,8/s

Tabla 3.69 Resultados pruebas rendimiento 10 usuarios

En la Tabla 3.69 se puede ver que los submódulos de **Horario**, **Tipo de servicio** y **Cargo** tienen las medias de tiempo más altas que van desde 3 a 4 segundos, lo cual sigue siendo un tiempo de respuesta considerablemente bajo para el número de peticiones actual, el porcentaje de error se mantiene en 0 % lo cual asegura el correcto funcionamiento de estos submódulos ante la carga de 10 peticiones.

Número de usuarios: 20

Submódulo	usuarios	Tiempo min	Tiempo máx	Media	%Error	Rendimiento
Login	20	5	648	64	0 %	5,6 / s
Frecuencia	20	6	210	22	0 %	6,1/s
Horario	20	73	77578	3083	0,21 %	2 s
Tipo de maquinaria	20	6	151	19	0 %	7 / s
Maquinaria	20	6	195	13	0 %	6,2/s
Ruta ubicación	20	6	746	32	0 %	6,8/s
Cambio supervisor ruta	20	6	606	47	0 %	5/s

Tabla 3.70 Resultados pruebas rendimiento 20 usuarios

En la Tabla 3.70 se puede observar que el submódulo con mayor media es el de **Horario** con tiempo de respuesta de 3 segundos, además muestra un porcentaje de error de 0,21 % lo cual se traduce en menos de una petición, indicando una probabilidad de error en peticiones de 0,042 lo cual representa menos de una petición.

Número de usuarios: 50

Submódulo	usuarios	Tiempo min	Tiempo máx	Media	%Error	Rendimiento
Login	50	3	1159	72	0 %	11,7/ s
Frecuencia	50	5	393	22	0 %	15,4/ s
Horario	50	1	60312	1984	0,17 %	7,4 s
Tipo de maquinaria	50	6	299	20	0 %	14,9/s
Maquinaria	50	6	221	13	0 %	14,4/s
Ruta ubicación	50	6	1302	42	0 %	15,9/s
Cambio supervisor ruta	50	6	771	75	0 %	16,2/s

Tabla 3.71 Resultados pruebas rendimiento 50 usuarios

En la Tabla 3.71 se puede observar que el submódulo con media más alta es el de **Horario** con tiempo de respuesta de 2 segundos aproximadamente ante un número de peticiones de 50, lo cual está bien ante tal cantidad de peticiones ya que es un tiempo de respuesta muy pequeño. Además, se muestra un porcentaje de error de 0,17 % lo cual se traduce en 0,085 indicando una probabilidad de error en peticiones menor a 1 usuario.

Pruebas de humo

Para evaluar las principales funcionalidades que debía cumplir cada submódulo del Módulo de Logística (registro, edición, lectura y consulta de datos) se utilizó como guía los casos de uso escritos anteriormente, con ellos se pudo tener una guía para realizar este tipo de prueba y se llegó a los siguientes resultados:

Submódulo	Actividad	Estado
Frecuencia	Registrar frecuencia	Aprobado
	Editar frecuencia	Aprobado
	Consultar frecuencia	Aprobado
Horario	Registrar horario	Aprobado
	Editar horario	Aprobado
	Consultar horario	Aprobado
Tipo de servicio	Registrar tipo de servicio	Aprobado
	Editar tipo de servicio	Aprobado
	Consultar tipo de servicio	Aprobado
Cargo-Talento humano	Registrar cargo	Aprobado
	Editar cargo	Aprobado
	Consultar cargo	Aprobado
Empleados-Talento humano	Registrar empleado	Aprobado
	Editar empleado	Aprobado
	Consultar empleado	Aprobado

Tipo de maquinaria	Registrar tipo de maquinaria	Aprobado
	Editar tipo de maquinaria	Aprobado
	Consultar tipo de maquinaria	Aprobado
Maquinaria	Registrar maquinaria	Aprobado
	Editar maquinaria	Aprobado
	Consultar maquinaria	Aprobado
Administración zonal	Registrar administración zonal	Aprobado
	Editar administración zonal	Aprobado
	Consultar administración zonal	Aprobado
Sectores	Registrar sector	Aprobado
	Editar sector	Aprobado
	Consultar sector	Aprobado
	Eliminar sector	Aprobado
Ruta ubicación	Registrar ruta ubicación	Aprobado
	Editar ruta ubicación	Aprobado
	Consultar ruta ubicación	Aprobado
Programación diaria de rutas	Registrar programación de ruta	Aprobado
	Editar programación de ruta	Aprobado
	Consultar programación de ruta	Aprobado
Cambiar supervisor de ruta	Cambiar ruta de supervisor	Aprobado
Reporte de programación diaria	Reporte de programación	Aprobado

Tabla 3.72 Resultados pruebas de humo

3.2.8. Evaluación del producto del sprint

A continuación, se muestran las fichas por cada sprint con los ID's de los requerimientos implementados y los atributos de calidad con los que estos cumplen.

Sprint 1

NRO Sprint		1
ID's HISTORIAS DE USUARIO IMPLEMENTADAS		PLA-001 PLA-002 PLA-003
Atributos de calidad	Seleccione con una x los atributos que cumple el producto	Observaciones
Funcionalidad	x	
Usabilidad	x	
Fiabilidad	x	
Rendimiento	x	
Mantenibilidad	x	
Total atributos calidad cumplidos		5

Tabla 3.73 Evaluación incremento sprint 1

Sprint 2

NRO Sprint		2
ID's HISTORIAS DE USUARIO IMPLEMENTADAS		PLA-004 PLA-005
Atributos de calidad	Seleccione con una x los atributos que cumple el producto	Observaciones
Funcionalidad	x	
Usabilidad	x	
Fiabilidad	x	
Rendimiento	x	
Mantenibilidad	x	
Total atributos calidad cumplidos		5

Tabla 3.74 Evaluación incremento sprint 2

Sprint 3

NRO Sprint		3
ID's HISTORIAS DE USUARIO IMPLEMENTADAS		PLA-006 PLA-007 PLA-008
Atributos de calidad	Seleccione con una x los atributos que cumple el producto	Observaciones
Funcionalidad	x	
Usabilidad	x	
Fiabilidad	x	
Rendimiento	x	
Mantenibilidad	x	
Total atributos calidad cumplidos		5

Tabla 3.75 Evaluación incremento sprint 3

Sprint 4

NRO Sprint		4
ID's HISTORIAS DE USUARIO IMPLEMENTADAS		PLA-009 PLA-010
Atributos de calidad	Seleccione con una x los atributos que cumple el producto	Observaciones
Funcionalidad	x	
Usabilidad	x	
Fiabilidad	x	
Rendimiento	x	
Mantenibilidad	x	
Total atributos calidad cumplidos		5

Tabla 3.76 Evaluación incremento sprint 4

Sprint 5

NRO Sprint		5
ID's HISTORIAS DE USUARIO IMPLEMENTADAS	PLA-011	
Atributos de calidad	Seleccione con una x los atributos que cumple el producto	Observaciones
Funcionalidad	x	
Usabilidad	x	
Fiabilidad	x	
Rendimiento	x	
Mantenibilidad	x	
Total atributos calidad cumplidos	5	

Tabla 3.77 Evaluación incremento sprint 5

4. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Se pudo combinar efectivamente el proceso ágil Scrum con el lenguaje de modelamiento UML sin perder la esencia del principio ágil, para ello se utilizaron únicamente los diagramas de casos de uso y clases que permitieron comprender de mejor manera la funcionalidad y estructura del sistema de software.
- La aplicación del proceso Scrum permitió al Scrum Team y Scrum Master reforzar las habilidades de trabajo en equipo y autoorganización. Aunque Scrum resalta la importancia de las reuniones diarias, esto no se pudo cumplir al pie de la letra, pero se procuró como mínimo mantener una reunión semanal, lo cual permitió a los miembros del Scrum Team interactuar más entre sí y como producto de ello reforzar sus habilidades técnicas sobre las tecnologías usadas en el desarrollo del sistema de software
- La aplicación del proceso de aseguramiento de la calidad desde etapas iniciales del desarrollo del sistema de software permitió mitigar el riesgo en etapas avanzadas del desarrollo del sistema, ya que se evaluó desde la correcta definición de requerimientos y criterios de aceptación hasta la ejecución de pruebas esenciales sobre el sistema de software.
- Los resultados de las pruebas de rendimiento mostraron un tiempo de respuesta no mayor a 6 segundos, incluso en escenarios de carga de 50 usuarios simultáneamente, lo cual es un tiempo pequeño y aceptable, puesto que también se incluye el acceso a la base de datos para recuperar los datos para mostrarlos y guardarlos en el caso de registro de datos.
- A pesar del acceso limitado al Módulo de Logística se hicieron las pruebas que se consideraron más importantes con herramientas gratuitas y que no requerían de acceso al código fuente por asuntos de acceso restringido a dicha información; los resultados de las pruebas ejecutadas sirvieron de retroalimentación al Scrum Team para realizar algunas mejoras tanto en desempeño como en usabilidad.
- Para el desarrollo del sistema se utilizaron herramientas seleccionadas por el Scrum Team y Scrum Master en base a la experiencia del Scrum Team en las mismas, usando licencias otorgadas por la empresa; en cuanto a la ejecución de pruebas, se utilizó la herramienta JMeter la cual es open source, que demostró ser muy completa para ejecutar pruebas de rendimiento y carga.
- Se capacitó al Product Owner, Scrum Team y Scrum Master en cuanto la metodología definida y al proceso de aseguramiento de la calidad, como producto de ello se desarrolló el Módulo de Logística del SGO, en donde se aplicaron los procesos definidos y se adaptó el personal involucrado de manera efectiva al nuevo ambiente de trabajo del Área de Desarrollo de EMASEO EP.
- La evaluación del nivel de compromiso de las partes interesadas dio como resultado general un promedio de **Excelente** lo cual muestra que el interés y colaboración en la aplicación de la metodología de desarrollo fue satisfactorio, también el nivel de cumplimiento de la metodología dio un promedio general **Alto**, lo cual demuestra que se tomaron en cuenta las sugerencias dadas para una buena adaptación de la metodología y proceso de aseguramiento de la calidad.

4.2. Recomendaciones

- Antes de sugerir un proceso o metodología de desarrollo de software, se recomienda analizar a profundidad el ambiente de trabajo y valores de la empresa en donde se desea aplicarlos, puesto que se requiere de un gran compromiso y cambio de mentalidad el poder adaptar el estilo de trabajo de las personas a una nueva propuesta de desarrollo de software.
- Se recomienda capacitar a profundidad al Scrum Team sobre herramientas para el desarrollo de pruebas, de preferencia que sean open source para que se pueda tener acceso total a todas las funcionalidades que estas ofrecen.
- Se recomienda aplicar la metodología definida poco a poco en los proyectos asignados al área, empezando por los de menor complejidad para ir aprendiendo más sobre los principios ágiles y adaptarlos al ambiente de trabajo de manera efectiva.
- Se recomienda incluir en el equipo de trabajo del Área de Desarrollo una persona especializada en el proceso de aseguramiento de la calidad, aunque lo más recomendable sería tener un departamento enfocado al aseguramiento de la calidad. Como esto representa un gasto de recursos muy grande, se recomienda empezar con un especialista que guíe en el proceso de aprendizaje y aplicación de buenas prácticas al desarrollo de nuevos sistemas de software en la empresa.

5. Bibliografía

- [1] F. Naranjo and Subdirección de TIC's, "Documento metodología de desarrollo de software EMASEO EP," Quito, Ecuador, 2018.
- [2] R. S. Pressman and D. Ph, *Ingeniería del software*, Séptima Ed. Distrito Federal, México: University of Connecticut.
- [3] I. Sommerville, *Ingeniería de software Novena Edición*, Novena Edi. 2011.
- [4] D. Gallardo, "Presentación Manifiesto ágil," pp. 1–13, 2001.
- [5] G. Booch, *Análisis de diseño orientado a objetos con aplicaciones*, Segunda Ed., vol. 34, no. 9. Santa Clara, California, 1992.
- [6] I. Jacobson, G. Booch, and J. Rumbaugh, *El Proceso Unificado de Desarrollo de Software*. España, 2000.
- [7] I. J. Grady Booch, James Rumbaugh, "The Unified Modeling Language User Guide," *Language*. 2002.
- [8] A. N. Cadavid, "Revisión de metodologías ágiles para el desarrollo de software," *Prospectiva*, vol. 11, no. 2, p. 30, 2013, doi: 10.15665/rp.v11i2.36.
- [9] L. Gimson, G. D. G. Co-director, and G. Rossi, "Metodologías ágiles y desarrollo basado en el conocimiento," UNIVERSIDAD NACIONAL DE LA PLATA, 2012.
- [10] K. Schwaber and J. Sutherland, "Guía de Scrum," p. 21, 2013.
- [11] I. C. S. Society, *Swebok v3.0*. 2014.
- [12] D. E. L. Software and D. E. L. A. Calidad, "Aseguramiento de la calidad del software," no. capítulo 23, pp. 368–382, 2000.
- [13] "Documento Vision." .
- [14] M. G. P., "Gestion De Cierre Del Proyecto." [Online]. Available: https://www.academia.edu/7798180/GESTION_DE_CIERRE_DEL_PROYECTO.
- [15] "Las Mejores Historias de Usuario INVEST & SMART ★ Be Agile My Friend." <https://www.beagilemyfriend.com/historias-de-usuario-invest-smart/> (accessed Apr. 11, 2020).
- [16] Apiumhub, "CÓMO ESCRIBIR BUENAS HISTORIAS DE USUARIO." <https://apiumhub.com/es/tech-blog-barcelona/como-escribir-buenas-historias-de-usuario/> (accessed Apr. 24, 2020).
- [17] Cillero Manuel, "Especificación del Plan de Pruebas - manuel.cillero.es." <https://manuel.cillero.es/doc/metrica-3/procesos-principales/asi/actividad-10/> (accessed May 27, 2020).
- [18] J. Cervantes, "Pruebas Ágiles o Agile testing. Equipo feliz, clientes contentos |

- Pensemos web,” *AGILE Y SCRUM*, 2019. <https://www.pensemosweb.com/pruebas-agiles-agile-testing-equipo-feliz-clientes-contentos/> (accessed May 27, 2020).
- [19] PMO- Oficina de Proyectos de Informatica, “Pruebas de software Agile: Planificar con los 4 cuadrantes del Agile Testing (1era parte) - La Oficina de Proyectos de Informática.” <http://www.pmoinformatica.com/2015/04/pruebas-software-agile-cuadrantes-1.html> (accessed May 27, 2020).
- [20] M. Cillero, “Pruebas Unitarias - manuel.cillero.es.” <https://manuel.cillero.es/doc/metrica-3/tecnicas/pruebas/unitarias> (accessed May 31, 2020).
- [21] Anna, “Test de Regresión en un escenario Ágil,” Jan. 20, 2018. <http://www.hablamosdecalidad.com/test-de-regresion-en-un-escenario-agil/> (accessed Jun. 11, 2020).
- [22] “060- Testing exploratorio. – Una QA En apuros.” <https://unaqaenapuros.wordpress.com/2019/11/06/060-testing-exploratorio/> (accessed May 31, 2020).
- [23] “Plantilla del plan de pruebas de software - La Oficina de Proyectos de Informática.” <http://www.pmoinformatica.com/2014/05/plan-de-pruebas-de-software.html> (accessed Jun. 04, 2020).
- [24] PMO Informatica, “Plantilla del plan de pruebas de software ,” May 19, 2014. <http://www.pmoinformatica.com/2014/05/plan-de-pruebas-de-software.html> (accessed Jun. 09, 2020).
- [25] “Mantenibilidad.” <https://iso25000.com/index.php/normas-iso-25000/iso-25010/26-mantenibilidad> (accessed Jun. 16, 2020).
- [26] “Maintainability Testing .” https://www.tutorialspoint.com/software_testing_dictionary/maintainability.htm (accessed Jun. 16, 2020).
- [27] S. Villarreal and E. Loachamin, “Arquitectura Sistema de Gestion de Operaciones.” Quito, Ecuador, 2020.

6. ANEXOS

En el CD se encuentran adjuntos los anexos detallados a continuación:

ANEXO 1: Documento de Visión del proyecto en formato .docx.

ANEXO 2: Criterios de aceptación en formato .xlsx.

ANEXO 3: Acta de cierre del proyecto en formato PDF.

ANEXO 4: Plan de pruebas en formato .docx.