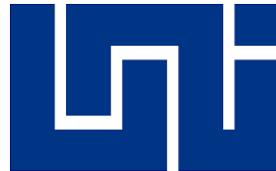


UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN
INGENIERÍA EN COMPUTACIÓN



**Sistema web de asignación y monitoreo de roles de aseo para la residencia estudiantil de la
UNI (SISREUNI).**

Elaborado por:

Br. Jonathan Alejandro Acuña Barrios.

Carnet: 2016-0091U

Br. Joaquín Gamaliel Ordoñez Díaz.

Carnet: 2016-0019U

Tutor:

MSc. Luis Eduardo Chávez Mairena.

Managua, septiembre 2023

Dedicatoria

Dedicamos este trabajo monográfico primeramente a Dios, que es el que nos ha guiado por el buen camino, nos ha llenado de fuerzas para seguir adelante y terminar esta monografía y no darnos por vencidos ante los inconvenientes que se presentaron, enseñándonos a enfrentarlos sin perder nunca nuestros valores éticos y morales. A nuestras familias que son el pilar en nuestras vidas y que, con su amor, comprensión y su ayuda en todo momento nos encaminaron a efectuar este gran logro.

Agradecimientos

Agradecemos en primer lugar, a Dios por habernos guiado en todo momento hasta cumplir este gran logro.

A nuestros padres por esforzarse para que saliéramos adelante y apoyarnos incondicionalmente en todo momento.

A nuestro estimado tutor MSc. Luis Eduardo Chávez Mairena por habernos guiado y compartido sus conocimientos con paciencia y generosidad para el logro de este gran reto.

A la Universidad Nacional de Ingeniería por habernos brindado los conocimientos que adquirimos en el transcurso de la carrera

A todos nuestros maestros por compartir con nosotros sus conocimientos.

A la encargada de la Residencia Estudiantil de la Universidad Nacional de Ingeniería por brindarnos toda la información y apoyo necesario en el proceso de recolección de información.

Resumen

El presente trabajo monográfico titulado “Sistema web de asignación y monitoreo de roles de aseo para la residencia estudiantil de la UNI (SISREUNI)” tiene como finalidad facilitar la labor de los encargados de la residencia estudiantil de la UNI.

El desarrollo de este sistema se llevó a cabo utilizando la metodología UWE UML lo que permitió avanzar de manera ágil, con la recopilación de información brindada por parte de los involucrados, obteniendo información de procesos, modelar mediante los diagramas más esenciales de UML el sistema, también codificar e implementar hasta validar el cumplimiento de los objetivos propuestos.

Al culminar este proyecto monográfico se logró que la Residencia Estudiantil tenga sus procesos más agiles y precisos, dejando a un lado manejar la información de diferentes fuentes de datos y procesos de manera manual y monótonos.

El sistema web está enfocado en ser una herramienta capaz de ayudar a gestionar diferentes procesos habituales como asignación de cuartos, roles de aseo, reporte de entrada/salida de los estudiantes con beca de residencia y control de solicitudes de comidas al comedor de la residencia estudiantil.

Tabla de contenido

1.	Introducción	1
2.	Antecedentes	2
3.	Justificación	3
4.	Objetivos	5
4.1.	Objetivo General	5
4.2.	Objetivos Específicos	5
5.	Marco Teórico	6
5.1.	Metodología UWE	6
5.1.1.	Fases de UWE	6
5.2.	Tecnologías y herramientas	9
5.2.1.	Patrón de diseño M.V.C.....	9
5.2.2.	Lenguaje de programación PHP.....	10
5.2.3.	Gestor de base de datos MySQLi.....	10
5.2.4.	Lenguaje HTML5 y CSS3.....	10
5.2.5.	Lenguaje de programación JavaScript	11
5.2.6.	Biblioteca Bootstrap.....	11
5.2.7.	Biblioteca jQuery	11
5.2.8.	AJAX.....	12
5.2.9.	SDK.....	12
5.2.10.	Tareas CRON.....	12
5.3.	Seguridad informática	13
5.3.1.	Protocolo de seguridad HTTPS	13
5.3.2.	Protección de cadena de conexión.....	13
6.	Estudio de factibilidad	14
6.1.	Factibilidad Técnica	14
6.2.	Factibilidad Operativa	16
6.3.	Factibilidad Económica	16
7.	Diseño metodológico.....	18
7.1.	Análisis del sistema.....	18
7.1.1.	Levantamiento de requerimientos	19
7.1.2.	Casos de Uso	25

7.1.3.	Detalles Casos de Uso Modelados.....	32
7.1.4.	Diagrama de clase.....	50
7.1.5.	Diagrama de actividades	51
7.1.6.	Diagramas de secuencias	67
7.2.	Diseño del sistema.....	80
7.2.1.	Modelo de navegación.....	80
7.2.2.	Modelo de presentación	81
7.2.3.	Modelo arquitectónico.....	85
7.2.4.	Diagrama de componentes.....	85
7.3.	Codificación.....	86
7.3.1.	Base de datos.....	86
7.3.2.	Tecnologías de desarrollo del código	86
7.3.3.	Aspectos de seguridad	86
7.4.	Pruebas.....	87
7.4.1.	Detalle de pruebas	88
7.5.	Implementación.....	91
8.	Conclusiones.....	92
9.	Recomendaciones	93
10.	Anexos	94
10.1.	Carta de solicitud de autorización para realizar sistema.....	94
10.2.	Carta de aceptación del sistema.....	95
11.	Bibliografía	96

Índice de Ilustraciones

ILUSTRACIÓN 1 FASES METODOLOGÍA UWE	18
ILUSTRACIÓN 2 CASO DE USO INICIO DE SESIÓN	25
ILUSTRACIÓN 3 CASO DE USO SISTEMA GENERAL	26
ILUSTRACIÓN 4 CASO DE USO GESTIÓN ESTUDIANTE	27
ILUSTRACIÓN 5 CASO DE USO GESTIÓN GRUPOS TELEGRAM	27
ILUSTRACIÓN 6 CASO DE USO GESTIÓN ASIGNACIONES DE ASEO	28
ILUSTRACIÓN 7 CASO DE USO PANEL COMEDOR	28
ILUSTRACIÓN 8 CASO DE USO GESTIÓN USUARIOS Y ROLES	29
ILUSTRACIÓN 9 CASO DE USO GESTIÓN TIPO DE ASEO	29
ILUSTRACIÓN 10 CASO DE USO GESTIÓN MOTIVO DE SALIDA	30
ILUSTRACIÓN 11 CASO DE USO GESTIÓN CUARTOS	30
ILUSTRACIÓN 12 CASO DE USO GESTIÓN CARRERAS	31
ILUSTRACIÓN 13 CASO DE USO GESTIÓN RECINTOS	31
ILUSTRACIÓN 14 DIAGRAMA DE CLASES	50
ILUSTRACIÓN 15 DIAGRAMA DE ACTIVIDAD INICIO DE SESIÓN	51
ILUSTRACIÓN 16 DIAGRAMA DE ACTIVIDAD GESTIÓN ESTUDIANTE	52
ILUSTRACIÓN 17 DIAGRAMA DE ACTIVIDAD CAMBIO DE CUARTO ESTUDIANTE	53
ILUSTRACIÓN 18 DIAGRAMA DE ACTIVIDAD REPORTAR SALIDA DE ESTUDIANTE	54
ILUSTRACIÓN 19 DIAGRAMA DE ACTIVIDAD REPORTAR REGRESO DE ESTUDIANTE	55
ILUSTRACIÓN 20 DIAGRAMA DE ACTIVIDAD HISTORIAL CAMBIO DE CUARTO Y DE SALIDAS	56
ILUSTRACIÓN 21 DIAGRAMA DE ACTIVIDAD VISUALIZAR ESTUDIANTE EN SISTEMA	57
ILUSTRACIÓN 22 DIAGRAMA DE ACTIVIDAD ASIGNAR ACTIVIDAD DE ASEO	58
ILUSTRACIÓN 23 DIAGRAMA DE ACTIVIDAD EDITAR ASIGNACIÓN DE ASEO	59
ILUSTRACIÓN 24 DIAGRAMA DE ACTIVIDAD ELIMINAR ASIGNACIÓN DE ASEO	60
ILUSTRACIÓN 25 DIAGRAMA DE ACTIVIDAD CALENDARIO DE ASIGNACIONES	61
ILUSTRACIÓN 26 DIAGRAMA DE ACTIVIDAD GESTIÓN TIPO DE ASEO	62
ILUSTRACIÓN 27 DIAGRAMA DE ACTIVIDAD GESTIÓN CUARTO	63
ILUSTRACIÓN 28 DIAGRAMA DE ACTIVIDAD GESTIÓN MOTIVO DE SALIDA	64
ILUSTRACIÓN 29 DIAGRAMA DE ACTIVIDAD GESTIÓN CARRERAS	65
ILUSTRACIÓN 30 DIAGRAMA DE ACTIVIDAD GESTIÓN RECINTOS	66
ILUSTRACIÓN 31 DIAGRAMA DE SECUENCIA INICIO DE SESIÓN	67
ILUSTRACIÓN 32 DIAGRAMA DE SECUENCIA INGRESAR NUEVO ESTUDIANTE	68
ILUSTRACIÓN 33 DIAGRAMA DE SECUENCIA EDITAR ESTUDIANTE	69
ILUSTRACIÓN 34 DIAGRAMA DE SECUENCIA ELIMINAR ESTUDIANTE	70
ILUSTRACIÓN 35 DIAGRAMA DE SECUENCIA VISUALIZAR ESTUDIANTE	71
ILUSTRACIÓN 36 DIAGRAMA DE SECUENCIA CAMBIO DE CUARTO A ESTUDIANTE	72
ILUSTRACIÓN 37 DIAGRAMA DE SECUENCIA REPORTAR ENTRADA/SALIDA ESTUDIANTE	73
ILUSTRACIÓN 38 DIAGRAMA DE SECUENCIA NUEVA ASIGNACIÓN DE ASEO	74
ILUSTRACIÓN 39 DIAGRAMA DE SECUENCIA EDITAR ASIGNACIÓN DE ASEO	75
ILUSTRACIÓN 40 DIAGRAMA DE SECUENCIA CALENDARIO DE ASIGNACIONES	76
ILUSTRACIÓN 41 DIAGRAMA DE SECUENCIA ENVÍO DE ENCUESTAS	77
ILUSTRACIÓN 42 DIAGRAMA DE SECUENCIA PANEL COMEDOR	78
ILUSTRACIÓN 43 DIAGRAMA DE SECUENCIA NUEVO USUARIO	79
ILUSTRACIÓN 44 MODELO DE NAVEGACIÓN	80
ILUSTRACIÓN 45 MODELO DE PRESENTACIÓN INICIO DE SESIÓN	81
ILUSTRACIÓN 46 MODELO DE PRESENTACIÓN GESTIÓN ESTUDIANTE	82
ILUSTRACIÓN 47 MODELO DE PRESENTACIÓN GESTIÓN ASIGNACIÓN DE ASEO	82

ILUSTRACIÓN 48 MODELO DE PRESENTACIÓN CALENDARIO DE ASIGNACIONES	83
ILUSTRACIÓN 49 MODELO DE PRESENTACIÓN PANEL COMEDOR	83
ILUSTRACIÓN 50 MODELO DE PRESENTACIÓN GESTIÓN USUARIOS Y ROLES	84
ILUSTRACIÓN 51 MODELO ARQUITECTÓNICO SISREUNI	85
ILUSTRACIÓN 52 DIAGRAMA DE COMPONENTES SISREUNI	85
ILUSTRACIÓN 53 SEGURIDAD CONSULTAS SQL	86
ILUSTRACIÓN 54 PRUEBA INTERFAZ NUEVO ESTUDIANTE	89
ILUSTRACIÓN 55 PRUEBA INTERFAZ LISTADO NUEVO REGISTRO	89
ILUSTRACIÓN 56 PRUEBA INTERFAZ NUEVA ASIGNACIÓN	90
ILUSTRACIÓN 57 PRUEBA NOTIFICACIÓN NUEVA ASIGNACIÓN	90
ILUSTRACIÓN 58 CARPETA CONTENEDORA DE CÓDIGO FUENTE DE SISREUNI	91
ILUSTRACIÓN 59 ESTRUCTURA DEL SISTEMA DESPLEGADA EN EL HOSTING	91
ILUSTRACIÓN 60 CARTA DE SOLICITUD SISTEMA	94
ILUSTRACIÓN 61 CARTA DE ACEPTACIÓN DIRECCIÓN DE BIENESTAR ESTUDIANTIL	95

Índice de Tablas

TABLA 1 DETALLE FACTIBILIDAD TÉCNICA	15
TABLA 2 DETALLE FACTIBILIDAD ECONÓMICA	17
TABLA 3 REQUERIMIENTOS FUNCIONALES GESTIÓN ESTUDIANTE	19
TABLA 4 REQUERIMIENTOS FUNCIONALES GESTIÓN ASIGNACIÓN DE ASEO	20
TABLA 5 REQUERIMIENTOS FUNCIONALES PANEL COMEDOR	21
TABLA 6 REQUERIMIENTOS FUNCIONALES GESTIÓN CATÁLOGOS TIPO ASEO	21
TABLA 7 REQUERIMIENTOS FUNCIONALES GESTIÓN CATÁLOGOS CUARTOS	22
TABLA 8 REQUERIMIENTOS FUNCIONALES GESTIÓN MOTIVO DE SALIDA	22
TABLA 9 REQUERIMIENTOS FUNCIONALES GESTIÓN CATÁLOGOS CARRERAS	22
TABLA 10 REQUERIMIENTOS FUNCIONALES GESTIÓN CATÁLOGOS RECINTO	23
TABLA 11 REQUERIMIENTOS FUNCIONALES GESTIÓN USUARIOS Y ROLES	23
TABLA 12 REQUERIMIENTOS FUNCIONALES ASPECTOS DE SEGURIDAD	24
TABLA 13 DETALLES REQUERIMIENTOS NO FUNCIONALES	24
TABLA 14 DETALLE CASO DE USO INICIO DE SESIÓN	32
TABLA 15 DETALLE CASO DE USO NUEVO ESTUDIANTE	33
TABLA 16 DETALLE CASO DE USO EDITAR ESTUDIANTE	33
TABLA 17 DETALLE CASO DE USO ELIMINAR ESTUDIANTE	34
TABLA 18 DETALLE CASO DE USO VISUALIZAR DETALLE DE ESTUDIANTE	34
TABLA 19 DETALLE CASO DE USO CAMBIO A ESTUDIANTE DE CUARTO	35
TABLA 20 DETALLE CASO DE USO REPORTAR SALIDA DE ESTUDIANTE	36
TABLA 21 DETALLE CASO DE USO REPORTAR REGRESO DE ESTUDIANTE	37
TABLA 22 DETALLE CASO DE USO VISUALIZAR HISTORIAL DE CAMBIO DE CUARTO DE ESTUDIANTE	37
TABLA 23 DETALLE CASO DE USO VISUALIZAR HISTORIAL DE SALIDAS DE ESTUDIANTE	38
TABLA 24 DETALLE CASO DE USO CALENDARIO DE ASIGNACIONES	39
TABLA 25 DETALLE CASO DE USO NUEVA ASIGNACIÓN DE ASEO	40
TABLA 26 DETALLE CASO DE USO EDITAR ASIGNACIÓN DE ASEO	41
TABLA 27 DETALLE CASO DE USO VISUALIZAR DETALLE DE ASIGNACIÓN	41
TABLA 28 DETALLE CASO DE USO ELIMINAR ASIGNACIÓN DE ASEO	42
TABLA 29 DETALLE CASO DE USO PANEL COMEDOR	43
TABLA 30 DETALLE CASO DE USO ENVÍO DE ENCUESTAS	43
TABLA 31 DETALLE CASO DE USO AGREGAR NUEVO CUARTO	44

TABLA 32 DETALLE CASO DE USO EDITAR CUARTO	44
TABLA 33 DETALLE CASO DE USO AGREGAR NUEVO TIPO DE ASEO	45
TABLA 34 DETALLE CASO DE USO EDITAR TIPO DE ASEO	46
TABLA 35 DETALLE CASO DE USO AGREGAR NUEVO USUARIO	46
TABLA 36 DETALLE CASO DE USO EDITAR USUARIO	47
TABLA 37 DETALLE CASO DE USO ELIMINAR USUARIO	48
TABLA 38 DETALLE CASO DE USO NUEVO GRUPO DE TELEGRAM	48
TABLA 39 DETALLE CASO DE USO EDITAR GRUPO DE TELEGRAM	49
TABLA 40 DETALLE CASO DE USO CAMBIAR ESTADO GRUPO DE TELEGRAM	49
TABLA 41 PRUEBAS DE VERIFICACIÓN	87
TABLA 42 CASO DE PRUEBA N°.1 REGISTRAR ESTUDIANTE	88
TABLA 43 DETALLE CASO DE PRUEBA N° 1 NUEVA ASIGNACIÓN DE ASEO	89

1. Introducción

En el presente documento se hace referencia al análisis y desarrollo de un sistema web de asignación y monitoreo de roles de aseo para la residencia estudiantil de la UNI.

Hoy en día los sistemas de información basados en web se han convertido en herramientas esenciales para la toma de decisiones y la trasferencia de información desde cualquier lugar, debido a su alta disponibilidad, ofreciendo gran dinamismo.

La Residencia estudiantil de la Universidad Nacional de Ingeniería UNI, es una extensión de la dirección de bienestar estudiantil, se encarga de brindar un hogar temporal a los estudiantes que provienen de diferentes regiones del país, que buscan una preparación profesional.

En la actualidad la residencia estudiantil UNI, lleva los procesos administrativos como asignar cuartos, roles de aseo, registro de entrada y salida de los estudiantes con beca de residencia y control de solicitudes de comidas al comedor estudiantil, de forma manual, y dicha información se almacena en hojas de Excel no especializada para este fin.

En el presente trabajo tomando en cuenta las ventajas de la programación web se desarrolló un sistema web que se ajusta a la necesidad de agilizar, automatizar y hacer más eficientes los procesos antes descritos.

A continuación, se describe la manera de cómo está estructurado el presente documento para una correcta comprensión del mismo, empezando por los antecedentes el cual refiere la revisión de trabajos previos.

Posteriormente se brinda la justificación en donde se describe del porque concretar la propuesta planteada, luego se plantean los objetivos de manera clara, directa, precisa y factibles, con el fin de tener una visión clara del desarrollo del sistema web.

Luego se procede a brindar la fundamentación teórica de la solución (marco teórico), y concretar con la metodología UWE (diseño metodológico), mostrando las fases de desarrollo de software, y culminar con la conclusión y finalmente los anexos que soportan este trabajo.

2. Antecedentes

A nivel nacional se desconoce la existencia de algún sistema especializado o similar al que se plantea para el área de residencia estudiantil en universidades públicas y privadas del país, por lo que el desarrollo del sistema web existiría únicamente en la residencia estudiantil de la Universidad Nacional de Ingeniería (UNI), ya que según el encargado de la residencia no tienen ningún sistema parecido, por lo que sus procesos se realizan algunos de forma manual y otros mediante hojas de cálculo de Excel.

Se desconoce de un software sea Desktop, web o móvil para residencias estudiantiles en el país, pero internacionalmente se pueden encontrar software como [Student Accommodation Management Software](#) que entre sus características esta ubicar a los estudiantes en dormitorios según las preferencias de estos (kinetic software, 2022).

Roompact que según su información puede manejar la información de asignación de residentes de campus universitarios (RoomPact, 2022) , a nivel empresarial encontramos Resource Guru este es un software de paga que entre sus características se encuentra la programación y gestión de tareas que muestra las actividades en un calendario de actividades, practico al momento de asignar y monitorear las actividades (Guru, 2019). Sin embargo, ninguno de estos está enfocado en los procesos que realmente se pretenden abarcar en esta propuesta de desarrollo, ni tampoco están diseñados de acuerdo a los objetivos que se plantean en el presente documento, ni al modelo de negocio de la residencia estudiantil de la UNI.

3. Justificación

Hoy en día las empresas e instituciones están en búsqueda de nuevos métodos que agilicen sus procesos y aporten nuevas estrategias de negocio debido a los progresos que han venido obteniendo las denominadas tecnologías de la información (Aguilera Trujillo, Miranda Alvarado, & Varela Mercado, 2016).

Actualmente hay necesidad de un software que agilice y automatice los procesos en la residencia estudiantil UNI, ya que algunos de estos como asignación de habitaciones, aseo, monitoreo de cumplimiento, registro de entrada y salida de los estudiantes con beca de residencia, y control de solicitudes de comidas al comedor estudiantil, se realizan de forma manual y toda esta información se maneja en Microsoft Excel y Word al no contar con un sistema enfocado para estas tareas.

En la residencia estudiantil UNI, la persona encargada tiene la tarea de asignar a cada becado de residencia una habitación teniendo en cuenta que hay determinada cantidad de cuartos para varones y cuartos para mujeres tomando en cuenta el año de ingreso a la universidad del estudiante, posteriormente realizar la compleja labor de asignar a cada estudiante distintos roles de aseo, para posteriormente monitorear el cumplimiento de los mismos y llevar registro en hojas de Excel. Estos procesos antes mencionados se llevan a cabo por semestres.

En el caso particular de la asignación de roles de aseo es un proceso variante ya que pueden haber reasignaciones de aseo en el transcurso de las semanas, todo en dependencia de la cantidad de estudiantes que se encuentren en la residencia estudiantil, ya que estos pueden reportar su salida, por lo que el encargado debe estar en constante monitoreo de la cantidad de estudiantes, para que no queden aseos sin realizarse, esta logística termina siendo tediosa y ocupan muchas horas de planificación según el encargado.

Por ende, se desarrolló un sistema web capaz de llevar a cabo las tareas antes mencionadas de manera más eficiente, ágil y automatizados los procesos de asignaciones de habitaciones y aseos, llevar registros de cumplimientos de aseo, entrada y salida de los estudiantes y control de solicitudes de comida con beca interna, asimismo

la generación de reportes que ayudaran en la toma de decisiones y otros beneficios que se describen a continuación:

- ❖ Se agiliza y automatiza los procesos de asignación de cuartos, roles de aseo, también control de solicitudes de comida al comedor estudiantil y se llevará mejor control.
- ❖ Se mejora el registro de información de comidas solicitadas al comedor estudiantil y la posibilidad de generar reportes por fechas.
- ❖ Mejor control de las entradas y salidas de los estudiantes.
- ❖ Se automatiza el envío de encuestas a estudiantes para saber si comerán los distintos tiempos de comida.
- ❖ La aplicación tendrá beneficios económicos a la Dirección de Bienestar Estudiantil, ya que no incurrirá en gastos de papelería debido a que no habrá necesidad de imprimir la información de asignaciones para los estudiantes.
- ❖ Los estudiantes podrán recibir mensajes indicándoles asignaciones de aseos nuevas o modificadas, asimismo poder solicitar la generación de un reporte al momento de salir de la residencia estudiantil de la UNI.
- ❖ Se mejora considerablemente los tiempos de búsqueda de información relacionada con la residencia estudiantil de la UNI, ya que actualmente se maneja en diferentes documentos de Word y Excel.

4. Objetivos

4.1. Objetivo General

- ❖ Desarrollar un sistema web que permita gestionar los procesos de asignación de habitaciones y roles de aseo para la Universidad Nacional de Ingeniería tomando como punto de implementación la residencia estudiantil.

4.2. Objetivos Específicos

- ❖ Analizar los procesos que se llevan a cabo en la residencia estudiantil UNI, para identificar los requerimientos funcionales asimismo como no funcionales y alcance del sistema a desarrollar.
- ❖ Diseñar el sistema a través del uso de las características de la metodología UWE y la herramienta case del lenguaje unificado de modelado UML con Enterprise Architect.
- ❖ Codificar los módulos seleccionados utilizando PHP como lenguaje de programación, MySQLi como gestor de base de datos, Bootstrap para la creación de las interfaces y siguiendo el patrón MVC.
- ❖ Implementar el sistema web en servidor para su uso.
- ❖ Realizar pruebas de validación y verificación del sistema.

5. Marco Teórico

En este apéndice se abordarán las definiciones que resultan de peculiar interés para conocer la base teórica del presente documento, como primer punto se abordará la metodología que se utilizó para el desarrollo del sistema web, como segundo punto las tecnologías y herramientas web.

5.1. Metodología UWE

UWE es una metodología de Ingeniería de software para el desarrollo de aplicaciones web basado en UML. Como indica (Alor-Hernández, 2014) “UWE es un enfoque metodológico para el desarrollo de aplicaciones web basadas en el proceso unificado. Se basa principalmente en los conceptos más relevantes proporcionados por otros métodos, pero se define una notación UML, que define un proceso de diseño sistemático y semiautomático”.

UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML (Solano, Miranda, & Dávila, 2020).

5.1.1. Fases de UWE

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrando además su atención en aplicaciones personalizadas o adaptativas. Las fases o etapas a utilizar son:

5.1.1.1. Análisis del sistema

En simple palabras, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web (Cantarero, Guido, & Rivera, 2016). Esta fase abarca las abstracciones primarias (clases y objetos) y mecanismos que están presentes en el dominio del problema. Las clases que se modelan son identificadas, con sus relaciones y descritas en un diagrama de clases. Las colaboraciones entre las clases para ejecutar los casos de uso también se consideran en esta fase a través de los modelos dinámicos en UML. (Zamitziz, 2020)

Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipo de la interfaz de usuario. Diseño del sistema

Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web (Cantarero, Guido, & Rivera, 2016).

5.1.1.2. Diseño del sistema

Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web, la particularidad de utilizar UWE es que los diagramas que por lo general son parte del diseño se llevan a cabo en la fase de análisis, dejando para la etapa de diseño los siguientes diagramas:

5.1.1.2.1. Modelo de navegación

El modelo de navegación define el flujo de navegación de la aplicación, así como el acceso a la navegación.

Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web. (Juárez Balmaceda & Contrera Mendoza, 2018).

5.1.1.2.2. Modelo de presentación

El modelo de presentación de UWE dibuja bloques lógicos de construcción de la interfaz de usuario de una aplicación web.

Describe dónde y cómo los objetos de navegación y accesos primitivos son presentados al usuario, es decir, una representación esquemática de los objetos visibles al usuario (Juárez Balmaceda & Contrera Mendoza, 2018).

5.1.1.2.3. Modelo Arquitectónico

El diseño arquitectónico se centra básicamente en la representación de la estructura de componentes del software, propiedades e interacciones existentes dentro del sistema. Además, permite ver claramente decisiones tempranas relacionadas con el diseño y que tendrán un gran impacto en el sistema (Juárez Balmaceda & Contrera Mendoza, 2018).

5.1.1.3. Codificación

Durante esta etapa se realizarán las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior (Cantarero, Guido, & Rivera, 2016).

5.1.1.4. Pruebas

Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de Código. (Cantarero, Guido, & Rivera, 2016).

Es una prueba básica para revisar si el sitio web desarrollado incluye todos los contenidos que se han especificado en los términos de referencia o los que se hayan definido en el marco del plan de desarrollo. Se puede hacer en forma manual o automática (Gobierno Digital, 2020).

5.1.1.5. Instalación o fase de implementación

Proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final. (Juárez Balmaceda & Contrera Mendoza, 2018).

5.2. Tecnologías y herramientas

Como parte de las herramientas que se utilizaron para el desarrollo del sistema web, tenemos las siguientes:

5.2.1. Patrón de diseño M.V.C

En líneas generales, MVC es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos (Alvarez, desarrolloweb, 2014)

A continuación, se definen los conceptos de modelo, vista y controlador según lo referido (Alvarez, desarrolloweb, 2014):

5.2.1.1. Modelos

Es la capa donde se trabaja con los datos, por tanto, contendrá mecanismos para acceder a la información y también para actualizar su estado. correspondientes selects, updates, inserts, delete, etc. (Alvarez, desarrolloweb, 2014).

5.2.1.2. Vistas

Las vistas, como su nombre hacen entender, contienen el código de la aplicación que va a producir la visualización de las interfaces de usuario. En las vistas nada más tenemos los códigos HTML y PHP que nos permite mostrar la salida. (Arias Cortez, Lopez Almendarez, & Navarrete Sequeria, 2018).

5.2.1.3. Controladores

Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, una búsqueda de información, etc. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino que es una capa que sirve de enlace entre las vistas y los modelos. (Alvarez, Desarrollo Web, 2020).

5.2.2. Lenguaje de programación PHP

PHP es un lenguaje de programación del lado del servidor que es capaz de ejecutarse en casi todo tipo de dispositivo electrónico moderno. (Angel, 2013) “Es un acrónimo recursivo para "PHP: Hypertext Preprocessor", originalmente Personal Home Page, es un lenguaje interpretado libre, usado originalmente solo para el desarrollo de aplicaciones presentes y que actuará en el lado del servidor, capaces de generar contenido dinámico en la World Wide Web”.

5.2.3. Gestor de base de datos MySQLi

MySQLi es una extensión mejorada (la i final es de improved) de PHP para acceder a bases de datos MySQL, que utiliza el lenguaje SQL (Lenguaje de Consulta Estructurada, del inglés Structured Query Language) como interfaz. Actualmente es una de las bases de datos más populares en versiones más actuales de PHP (Angel, 2013)

Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos.

5.2.4. Lenguaje HTML5 y CSS3

HTML usa un lenguaje de etiquetas para construir páginas web. HTML5 provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado

oficialmente, pero, incluso cuando algunas APIs (Interfaz de Programación de Aplicaciones) y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y Javascript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y Javascript hace el resto (Gauchat, 2012).

5.2.5. Lenguaje de programación JavaScript

En el Desarrollo se trabajó con JavaScript para dar mayor dinamismo a las páginas web como efectos, alertas animadas y utilizar otro framework. Según (Maza, 2012) “JavaScript fue diseñado para ser un lenguaje de elaboración de script que pudieran incrustarse en archivos HTML. No es compilado, sino que, en vez de ello, es interpretado por el navegador y este fue creado para darle dinamismo a las páginas web”. En otras palabras, los programas escritos con JavaScript se pueden probar en cualquier navegador.

5.2.6. Biblioteca Bootstrap

Para la elaboración del diseño visual se utilizó la biblioteca Bootstrap, según (Otto, 2017) Bootstrap “es un conjunto de herramientas de código abierto para desarrollar con HTML, CSS y JS”. Este permite rápidamente construir prototipos o construir una aplicación completa con variables, sistema de rejilla de respuesta, amplios componentes preconstruidos y poderosos complementos construidos en jQuery.

5.2.7. Biblioteca jQuery

jQuery es una biblioteca JavaScript libre y Open Source, del lado cliente, que se centra en la interacción entre el DOM, JavaScript, AJAX y HTML. Cuyo objetivo de esta librería es simplificar los comandos comunes de JavaScript. De hecho, el lema de jQuery es << Escribir menos para hacer más >>. (Lancker, 2014).

5.2.8. AJAX

Se utilizó esta tecnología debido a que hace que las aplicaciones web sean desarrolladas con más funcionalidad y más user-friendly, Según (Arias, 2015) “Ajax (Asynchronous JavaScript and XML) usa la transferencia de datos asíncrona (peticiones del HTTP) entre el navegador y el servidor web, permitiendo que las páginas webs envíen pedazos de pequeñas informaciones del usuario en vez de enviar las páginas enteras”.

5.2.9. SDK

Un kit de desarrollo de software (SDK) es un conjunto de herramientas de creación específicas de plataformas para desarrolladores. Se necesitan componentes como depuradores, compiladores y bibliotecas para crear código que se ejecute en una plataforma, sistema operativo o lenguaje de programación específicos. Los SDK ponen todo lo que necesita para desarrollar y ejecutar software en un solo lugar. Además, contienen recursos como documentación, tutoriales y guías, así como API y marcos para un desarrollo de aplicaciones más rápido. (AMAZON, 2023)

5.2.10. Tareas CRON

Las tareas cron son trabajos detrás de escena que se ejecutan automáticamente y en una planificación fija.

Cron es un demonio (que es como se conoce a un proceso en segundo plano) que se ejecuta desde el mismo instante en el que arranca el sistema operativo. Cron se encargará de comprobar si existe alguna tarea (job) para ser ejecutada, de acuerdo a la hora configurada en el propio sistema operativo. Es muy importante que la hora esté bien configurada, y también la zona horaria, de lo contrario, las ejecuciones que realice Cron no se corresponderán con nuestras configuraciones. (Redeszone, 2023).

5.3. Seguridad informática

La seguridad informática es una disciplina que se encarga de proteger la integridad y la privacidad de la información almacenada en un sistema informático (S.A, 2019).

Un sistema seguro debe ser íntegro (con información modificable sólo por las personas autorizadas), confidencial (los datos tienen que ser legibles únicamente para los usuarios autorizados), irrefutable (el usuario no debe poder negar las acciones que realizó) y tener buena disponibilidad (debe ser estable) (Google, 2016).

5.3.1. Protocolo de seguridad HTTPS

HTTPS (HyperText Transfer Protocol Secure, protocolo seguro de transferencia de hipertexto) es un protocolo de comunicación de Internet que protege la integridad y la confidencialidad de los datos de los usuarios entre sus ordenadores y el sitio web (developers google, 2021).

5.3.2. Protección de cadena de conexión

La protección del acceso al origen de datos es uno de los objetivos más importantes a la hora de proteger una aplicación. Una cadena de conexión presenta una vulnerabilidad potencial si no se protege o si se construye incorrectamente (microsoft, 2021).

6. Estudio de factibilidad

Después de definir la problemática presente y establecer las causas que ameritan el servicio web es pertinente un estudio de factibilidad para determinar la infraestructura tecnológica y la capacidad técnica que implica la implantación del sistema en cuestión, así como los costos y beneficios y el grado de aceptación. La factibilidad se apoya en 3 aspectos básicos, y estos son, Operativo, Técnico y Económico.

6.1. Factibilidad Técnica

La factibilidad técnica tiene por objeto proveer información para cuantificar el monto de las inversiones y de los costos de operación pertinentes a esta área. Podrá obtenerse la información de las necesidades de capital, mano de obra y recursos materiales, tanto para la puesta en marcha como para la posterior operación del proyecto". En este caso toda aquella información técnica necesaria para el sistema. (Olivas, 2020).

Para el proyecto se requieren los recursos que a continuación se detallaran:

Tabla 1 Detalle factibilidad técnica

Equipo o recurso	Cantidad	Especificación
Hardware		
Computador	Mínima 2	Procesador: Intel(R) Core (TM) i5-3470 CPU @3.20 GHz RAM:4 GB Disco duro:500 GB (MINIMO)
Impresora	Mínima 1	----
Software		
Enterprise Architect	1 licencia	Enterprise Architect para modelado de sistema
SO: Windows 10pro	2 licencia	Licencia para el uso de Windows.
Instalar php para ejecutar MySQL	1 instalador	instalador para el uso de dichas herramientas
Visual Studio Code	1 instalador	Instalador para codificación
Navegador Chrome o Firefox	1 instalador	Instalador Última versión
Microsoft Word	1 licencia	Licencia para documentación del desarrollo del sistema.
Comunicación		
Servidor	1	Servidor en la nube para alojar el sistema web. •Procesador a 1,4 GHz o superior • 8 GB RAM o Superior

Fuente: elaboración propia

6.2. Factibilidad Operativa

La factibilidad operativa determina si el proyecto a implementarse funcionara y será utilizado una vez que sea implementado, esto depende de los recursos humanos disponibles en la organización y de brindar la capacitación adecuada para el uso óptimo del mismo.

El proyecto desarrollado es viable operativamente por los beneficios que se obtendrán lo cual garantizará su permanencia en el tiempo. Pero podría generarse inconformidad entre los usuarios por ser una herramienta nueva, por lo que se hace énfasis a la importancia de la capacitación de los usuarios y capturar todas las sugerencias que estos puedan hacer con referente al mismo.

6.3. Factibilidad Económica

La factibilidad Económica es “ordenamiento y sistematización de la información de carácter monetario y elaboración de los cuadros analíticos que sirven de base para la evaluación económica. Comienza con la determinación de los costos totales y de la inversión inicial a partir de los estudios de ingeniería, ya que estos costos dependen de la tecnología seleccionada”. (Olivas, 2020)

Tabla 2 Detalle factibilidad económica

Equipo o recurso	Costo (Expresado en dólares)	Cantidad	Total (Expresado en dólares)
Hardware			
Computador (Ya Poseen)	\$500	2	0
Impresora	\$300	1	300
Software			
Windows 10 pro	\$200(anual)	2	400
Instalador de PHP	Gratis	2	0
Navegador Chrome	Gratis	2	0
Visual Studio Code	Gratis	2	0
Microsoft Word	100 (anual)	2	200
Enterprise Architect	299 (anual)	2	598
Comunicación			
Servidor GoDaddy Deluxe	167.88 (anual)	1	167.88
Recursos Humanos			
Analista de Sistemas	500/Mensual	1 (2 Meses)	1000
Programador	600/Mensual	2 (3 Meses)	1800
Administrador de bases de datos	500/Mensual	1 (1 Mes)	500
TOTAL			4,965.88

Fuente: elaboración propia

7. Diseño metodológico

En el diseño metodológico se detallan las fases de la metodología UWE UML a utilizar y además del cumplimiento de los objetivos a continuación se presentan de manera gráfica las fases y detalle de cada etapa de la metodología.

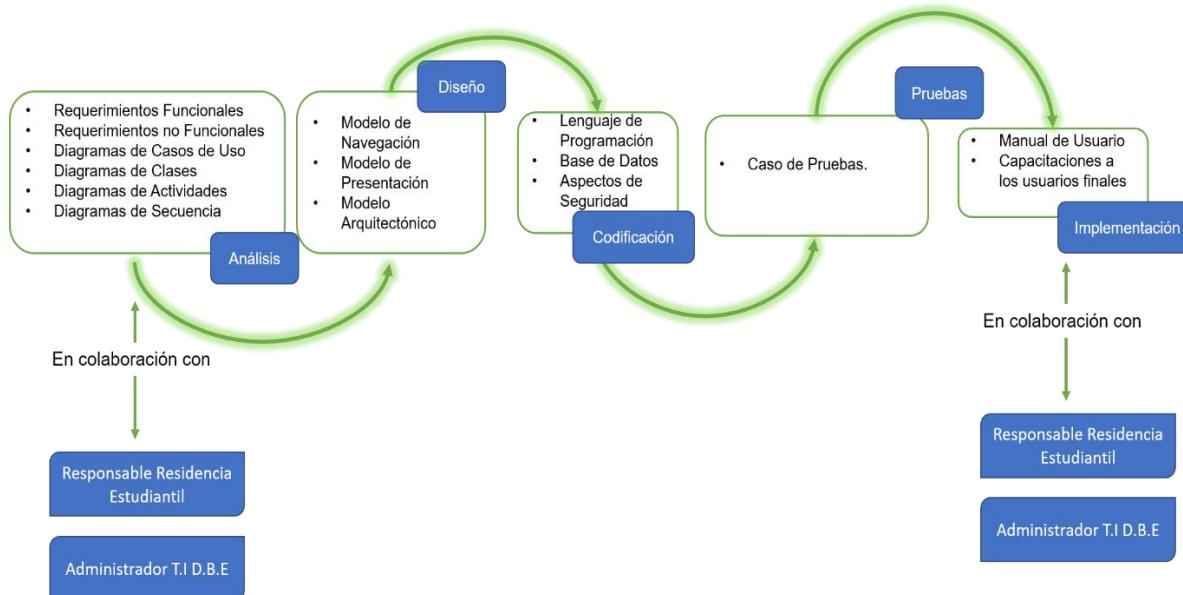


Ilustración 1 Fases metodología UWE

Fuente: elaboración propia

7.1. Análisis del sistema

Dentro de la metodología UWE que se explicó en el marco teórico, el primer paso es la recopilación de información necesaria para determinar las características funcionales y no funcionales con las se regirá el sistema para atender las necesidades de la administración, así como la elaboración de los diagramas UML y obtener una vista del sistema a modelar.

7.1.1. Levantamiento de requerimientos

Esta etapa se basa en las entrevistas verbales realizadas al personal que estará haciendo uso del sistema de la Residencia estudiantil UNI recinto Simón Bolívar, así como el análisis de los documentos que manejan como Excel, Word, con el fin de conocer más a detalle el flujo de procesos, todo ello con el objetivo de tener un mejor conocimiento del modelo del negocio y de la misma manera conocer los requerimientos funcionales y no funcionales del sistema.

7.1.1.1. Requerimientos Funcionales del sistema

Tabla 3 Requerimientos funcionales Gestión estudiante

Gestión Estudiantes	
RF 01	El sistema permitirá a usuarios autorizados el ingresar un nuevo estudiante en el sistema.
RF 02	El sistema permitirá a usuarios autorizados el editar información un estudiante en el sistema.
RF 03	El sistema permitirá a usuarios autorizados el eliminar un registro estudiante en el sistema.
RF 04	El sistema debe permitir visualizar listado de estudiantes registrados en sistema y que no estén eliminados.
RF 05	El sistema permitirá filtrar la información en base un dato en específico.
RF 06	El sistema debe mostrar cuando un estudiante este fuera este ausente de la residencia.
RF 07	El sistema debe validar que el número de carnet por ingresar sea único y no exista en sistema.
RF 08	El sistema debe validar que se muestren los cuartos asociados al género del estudiante.
RF 09	El sistema debe validar que no se combinen en un cuarto estudiantes con diferentes años de ingreso mediante el carné.
RF 10	El sistema debe validar los tipos de datos en los campos.
RF 11	El sistema permitirá cambiar estudiantes de cuarto aplicando los mismos criterios de verificación de asignación de cuarto.
RF 12	El sistema debe permitir visualizar el historial de cambio de cuarto de un estudiante en específico.
RF 13	El sistema debe permitir visualizar información detallada de un estudiante en específico.
RF 14	El sistema permitirá ingresar la salida del estudiante de la residencia según motivo de vacaciones, enfermedad, entre otros.

RF 15	El sistema debe generar un archivo PDF con detalle de la salida del estudiante para su posterior impresión.
RF 16	El sistema permitirá visualizar el historial de salidas del un estudiante en específico.
RF 17	El sistema debe mostrar mensajes amigables al momento de guardar un registro de manera exitosa, o al ocurrir un error.
RF 18	El sistema debe validar que los campos obligatorios no vayan vacíos.
RF 19	Se debe poder filtrar la información al mostrar el historial de cambios de cuarto
RF 20	Se debe poder filtrar la información al mostrar el historial de Entradas y salidas.
RF 21	El sistema debe permitir cancelar el proceso de guardar, editar y eliminar.

Fuente: elaboración propia

Tabla 4 Requerimientos funcionales Gestión Asignación de aseo

Gestión Asignaciones de Aseo	
RF 22	El sistema permitirá a usuarios autorizados el asignar actividades de aseo a estudiantes en el sistema.
RF 23	El sistema permitirá a usuarios autorizados el editar asignaciones de aseo de los estudiantes en el sistema.
RF 24	El sistema permitirá a usuarios autorizados el eliminar un registro asignación de aseo en el sistema.
RF 25	El sistema debe permitir visualizar listado de asignaciones de aseo registradas en sistema y que no estén eliminados.
RF 26	El sistema permitirá filtrar la información en base un dato en específico al presentar listado de asignaciones.
RF 27	El sistema debe presentar las asignaciones de aseo en un calendario en donde se agrupen dichas asignaciones por día.
RF 28	El sistema debe validar los tipos de datos en los campos.
RF 29	El sistema debe enviar un SMS al estudiante al asignársele una actividad de aseo.
RF 30	El sistema debe permitir visualizar información detallada de una asignación.
RF 31	El sistema debe enviar un SMS cuando se modifique su asignación de aseo.
RF 32	El sistema debe permitir actualizar el estado de una asignación a completada o incompleta.
RF 33	El sistema debe mostrar mensajes amigables al momento de guardar un registro de manera exitosa, o al ocurrir un error.
RF 34	El sistema debe validar que los campos obligatorios no vayan vacíos.

Fuente: elaboración propia

Tabla 5 Requerimientos funcionales Panel comedor

Panel Comedor	
RF 35	El sistema permitirá a usuarios autorizados visualizar los datos de solicitudes de Almuerzos, cenas y desayunos de forma amigable y comprensible mediante gráficos.
RF 36	El sistema debe permitir visualizar los datos de comidas solicitadas agrupadas por carreras y tiempo de comida.
RF 37	El sistema permitirá a usuarios autorizados generar un reporte en formato Excel con la información de las solicitudes de comida al comedor.
RF 38	El sistema debe permitir seleccionar un rango de fechas para generar el reporte.
RF 39	El sistema permitirá filtrar la información en base un dato en específico al presentar listado de asignaciones.
RF 40	El sistema debe automatizar el envío de encuestas de solicitud de comida a los grupos de la residencia estudiantil por carrera mediante una aplicación de mensajería instantánea.
RF 41	El sistema debe enviar la encuesta de solicitud de almuerzo a la 10 AM y dejarla activa por 35 minutos, luego de este tiempo cerrar la encuesta.
RF 42	El sistema debe enviar la encuesta de solicitud de cena y desayuno a la 3 PM y dejarla activa por 35 minutos, luego de este tiempo cerrar la encuesta
RF 43	Los resultados de las encuestas de comida deben visualizarse en sistema.

Fuente: elaboración propia

Tabla 6 Requerimientos funcionales Gestión catálogos Tipo aseo

Gestión Catálogos Tipo Aseos	
RF 44	El sistema permitirá a usuarios autorizados el ingresar un tipo de aseo en el sistema.
RF 45	El sistema permitirá a usuarios autorizados el editar información un tipo de aseo en el sistema.
RF 46	El sistema permitirá a usuarios autorizados el eliminar un registro de tipo de aseo en el sistema.
RF 47	El sistema debe permitir visualizar listado de tipo de aseo registrados en sistema y que no estén eliminados.
RF 48	El sistema permitirá filtrar la información en base un dato en específico.
RF 49	El sistema debe validar que los campos requeridos no estén vacíos.
RF 50	El sistema debe mostrar alertas intuitivas al guarda correctamente o al ocurrir un error.
RF 51	El sistema debe permitir cancelar la acción ya sea al guardar, editar o eliminar.

Fuente: elaboración propia

Tabla 7 Requerimientos funcionales Gestión catálogos cuartos

Gestión Catálogos Cuartos	
RF 52	El sistema permitirá a usuarios autorizados el ingresar información de un nuevo cuarto en el sistema.
RF 53	El sistema permitirá a usuarios autorizados editar información un cuarto en el sistema.
RF 54	El sistema permitirá a usuarios autorizados el eliminar un registro de un cuarto en el sistema.
RF 55	El sistema debe permitir visualizar listado de los cuartos registrados en sistema y que no estén eliminados.
RF 56	El sistema permitirá filtrar la información en base un dato en específico.
RF 57	El sistema debe validar que los campos requeridos no estén vacíos.
RF 58	El sistema debe mostrar alertas intuitivas al guarda correctamente o al ocurrir un error.
RF 59	El sistema debe permitir cancelar la acción ya sea al guardar, editar o eliminar.
RF 60	El sistema debe permitir asignar un género al cuarto para distinguir cuartos de mujeres y hombres.

Fuente: elaboración propia

Tabla 8 Requerimientos funcionales Gestión Motivo de salida

Gestión Catálogos Motivos de Salida	
RF 61	El sistema permitirá a usuarios autorizados el ingresar nuevo motivo de salida en el sistema.
RF 62	El sistema permitirá a usuarios autorizados editar información un motivo de salida en el sistema.
RF 63	El sistema permitirá a usuarios autorizados el eliminar un registro de un motivo de salida en el sistema.
RF 64	El sistema debe permitir visualizar listado de los motivos de salidas registrados en sistema y que no estén eliminados.
RF 65	El sistema permitirá filtrar la información en base un dato en específico.
RF 66	El sistema debe validar que los campos requeridos no estén vacíos.
RF 67	El sistema debe mostrar alertas intuitivas al guarda correctamente o al ocurrir un error.
RF 68	El sistema debe permitir cancelar la acción ya sea al guardar, editar o eliminar.

Fuente: elaboración propia

Tabla 9 Requerimientos funcionales Gestión catálogos carreras

Gestión Catálogos Carreras	
RF 69	El sistema permitirá a usuarios autorizados el ingresar una nueva carrera en el sistema.

RF 70	El sistema permitirá a usuarios autorizados editar información una carrera en el sistema.
RF 71	El sistema permitirá a usuarios autorizados el eliminar un registro de carrera en el sistema.
RF 72	El sistema debe permitir visualizar listado de las carreras registradas en sistema y que no estén eliminados.
RF 73	El sistema permitirá filtrar la información en base un dato en específico.
RF 74	El sistema debe validar que los campos requeridos no estén vacíos.
RF 75	El sistema debe mostrar alertas intuitivas al guarda correctamente o al ocurrir un error.
RF 76	El sistema debe permitir cancelar la acción ya sea al guardar, editar o eliminar.

Fuente: elaboración propia

Tabla 10 Requerimientos funcionales Gestión catálogos Recinto

Gestión Catálogos Recinto	
RF 77	El sistema permitirá a usuarios autorizados el ingresar un nuevo Recinto en el sistema.
RF 78	El sistema permitirá a usuarios autorizados editar información un Recinto en el sistema.
RF 79	El sistema permitirá a usuarios autorizados el eliminar un registro de un Recinto en el sistema.
RF 80	El sistema debe permitir visualizar listado de los Recintos registrados en sistema y que no estén eliminados.
RF 81	El sistema permitirá filtrar la información en base un dato en específico.
RF 82	El sistema debe validar que los campos requeridos no estén vacíos.
RF 83	El sistema debe mostrar alertas intuitivas al guarda correctamente o al ocurrir un error.
RF 84	El sistema debe permitir cancelar la acción ya sea al guardar, editar o eliminar.

Fuente: elaboración propia

Tabla 11 Requerimientos funcionales Gestión Usuarios y Roles

Gestión Usuarios y Roles	
RF 85	El sistema permitirá a únicamente al administrador ingresar un nuevo usuario en el sistema.
RF 86	El sistema permitirá al administrador editar información un usuario en el sistema.
RF 87	El sistema permitirá al administrador eliminar un registro de un usuario en el sistema.
RF 88	El sistema debe permitir visualizar listado de los usuarios registrados en sistema y que no estén eliminados.

RF 89	El sistema permitirá filtrar la información en base un dato en específico.
RF 90	El sistema debe validar que los campos requeridos no estén vacíos.
RF 91	El sistema debe exigir la asignación de un rol al momento de registrar un usuario.
RF 92	El sistema debe mostrar alertas intuitivas al guarda correctamente o al ocurrir un error.
RF 93	El sistema debe permitir cancelar la acción ya sea al guardar, editar o eliminar.

Fuente: elaboración propia

Tabla 12 Requerimientos funcionales Aspectos de seguridad

Aspectos de Seguridad	
RF 94	El sistema se inicia con un usuario administrador registrado de manera manual en la base de datos
RF 95	Todo usuario registrado tendrá credenciales de usuario y contraseña para iniciar sesión en el sistema.
RF 96	Solo podrán ingresar al sistema los usuarios que estén registrados.
RF 97	El usuario administrador podrá crear otros usuarios y sus respectivas credenciales de inicio de sesión.
RF 98	El usuario administrador podrá editar a un usuario registrado y sus credenciales de inicio de sesión.
RF 99	Los usuarios creados que no sean del tipo de rol administrador no podrán editar, crear o eliminar usuarios.

Fuente: elaboración propia

7.1.1.2. Requerimientos no funcionales

Tabla 13 Detalles requerimientos no funcionales

Requerimientos no funcionales	
RNF 01	Al ser el sistema una plataforma web, la conexión es muy importante
RNF 02	Se tendrá acceso al sistema con un computador o un dispositivo móvil.
RNF 03	Los usuarios del sistema deben tener a mano un manual de usuario del sistema.
RNF 04	Interfaz intuitiva y fácil de usar.
RNF 05	Diseño con MVC para agregar nuevas funcionalidades en el tiempo.
RNF 06	El sistema debe mostrarse correctamente en dispositivos con resolución menor a una PC de escritorio

Fuente: elaboración propia

7.1.2. Casos de Uso

Los casos de uso ayudan a determinar la funcionalidad y características del software desde la perspectiva del usuario.

- **Inicio de sesión.**

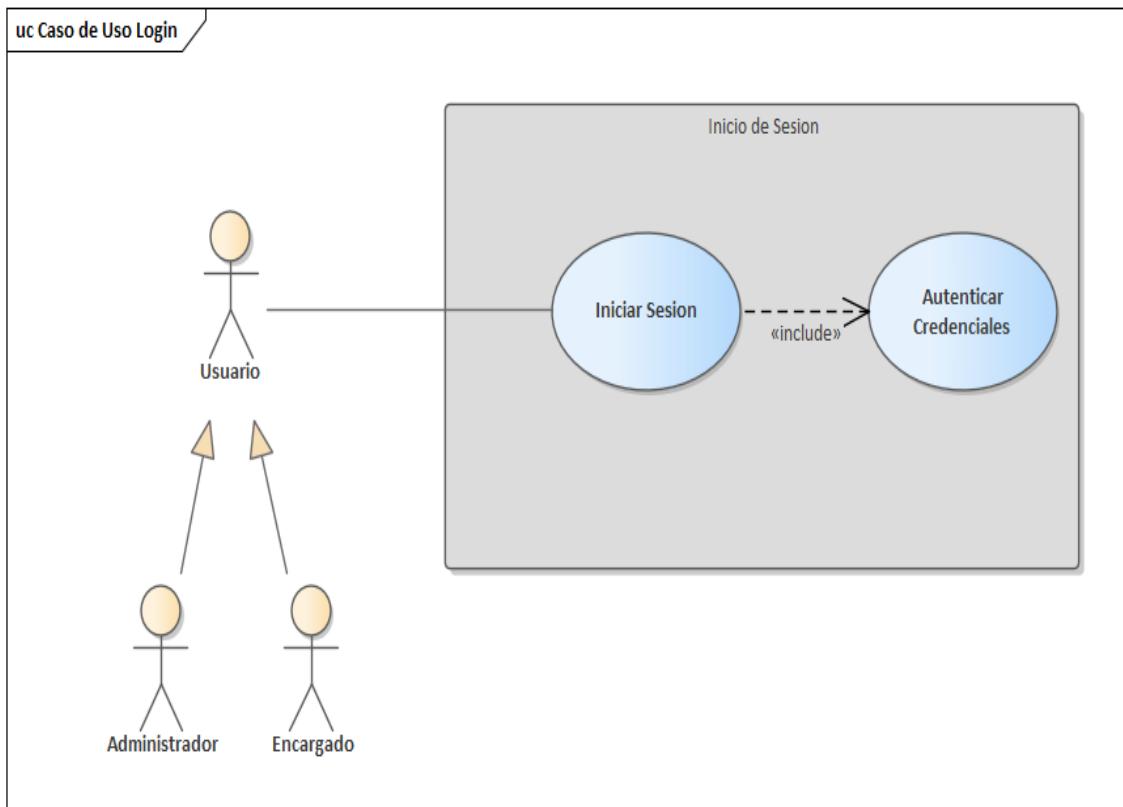


Ilustración 2 Caso de uso Inicio de sesión

Fuente: elaboración propia

• Sistema General

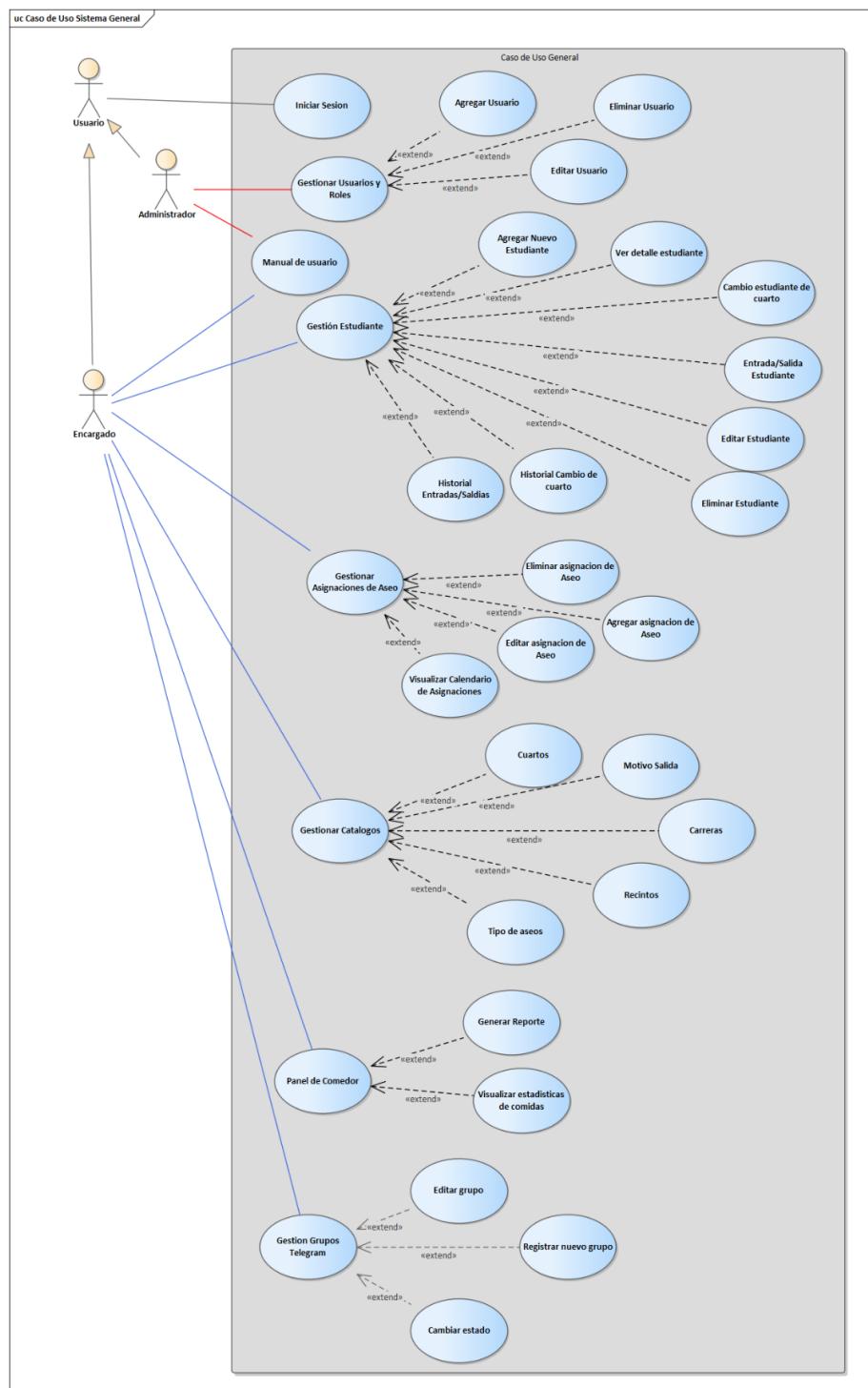


Ilustración 3 Caso de uso sistema general

Fuente: elaboracion propia

- **Gestión Estudiante**

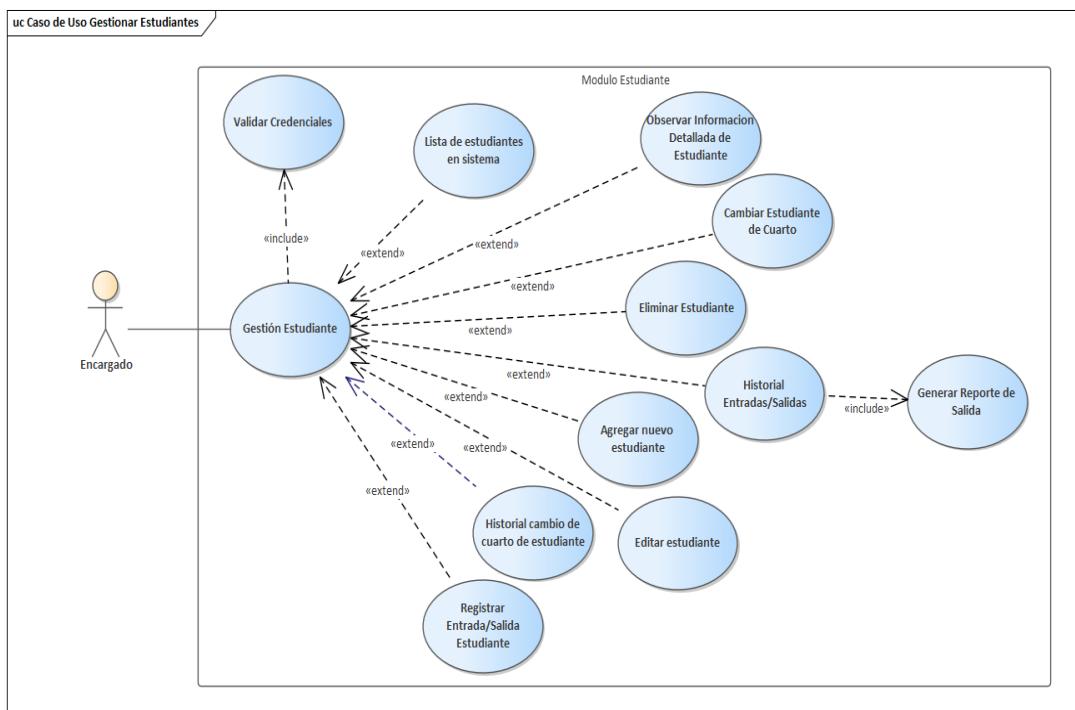


Ilustración 4 Caso de uso Gestión estudiante

Fuente: elaboración propia.

- **Gestionar Grupos Telegram**

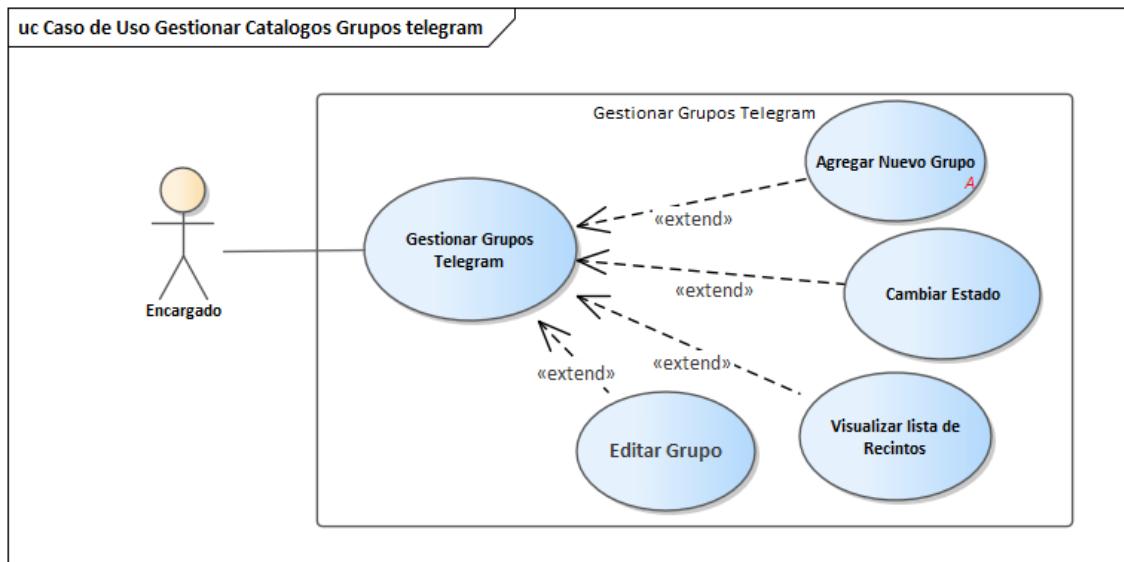


Ilustración 5 Caso de uso Gestión grupos Telegram

Fuente: elaboración propia

- **Gestion Asignaciones de aseo**

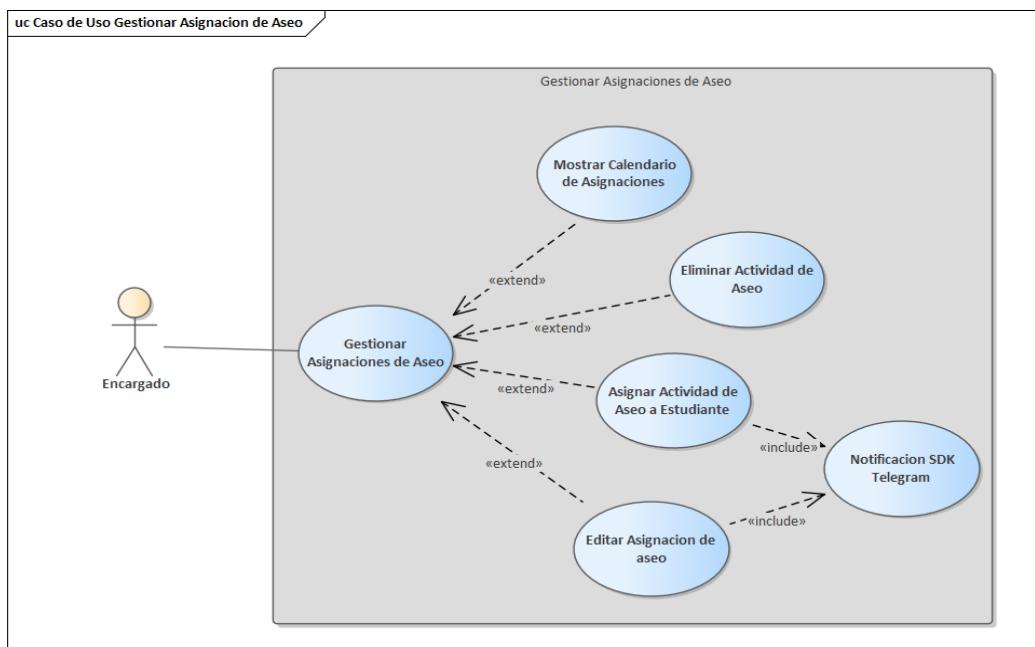


Ilustración 6 Caso de uso Gestión asignaciones de aseo

Fuente: elaboración propia.

- **Panel comedor**

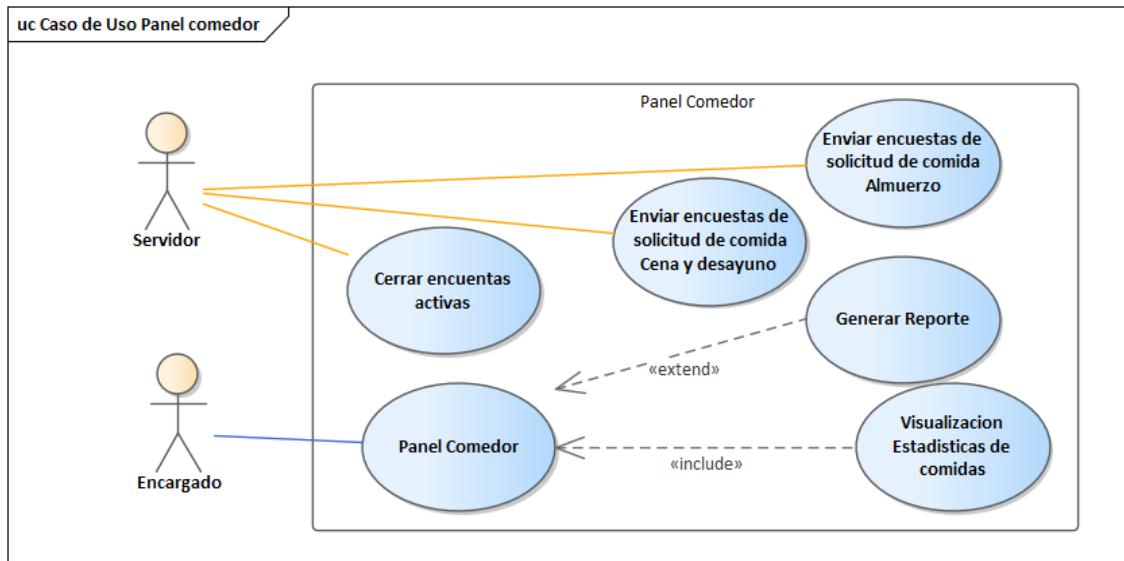


Ilustración 7 Caso de uso Panel comededor

Fuente: elaboración propia.

- **Gestión Usuarios y Roles**

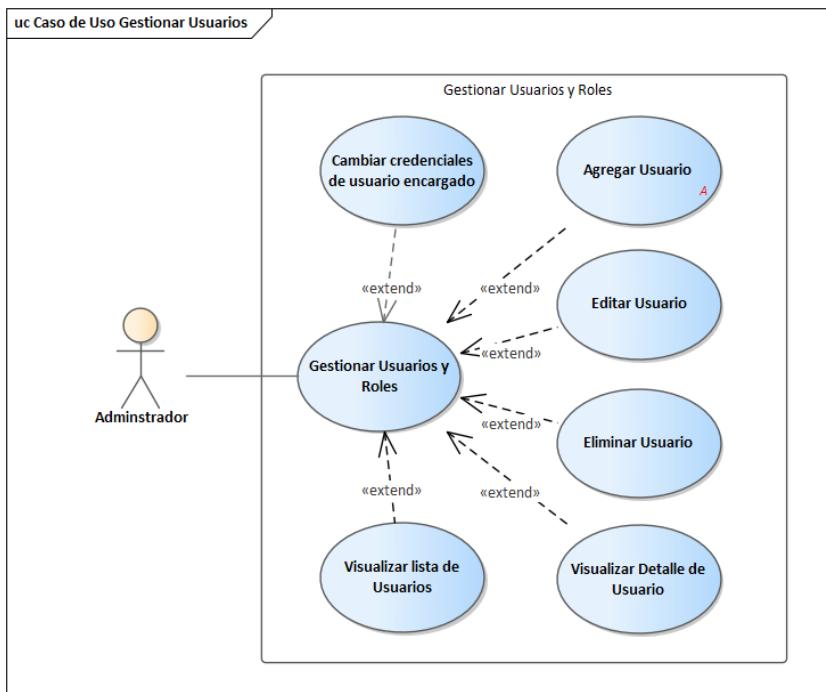


Ilustración 8 Caso de uso Gestión Usuarios y Roles

Fuente: elaboración propia.

- **Gestionar Tipos de Aseo**

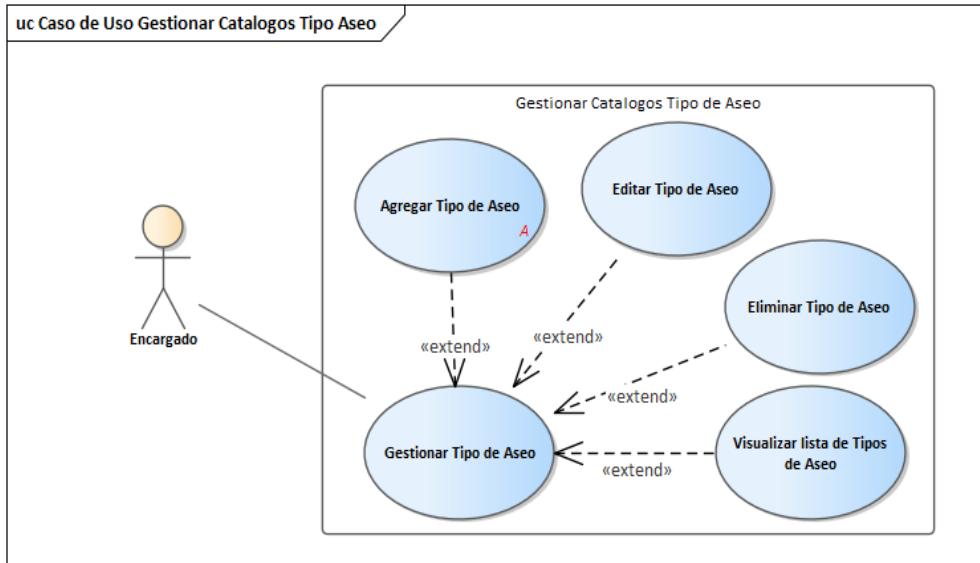


Ilustración 9 Caso de uso Gestión tipo de aseo

Fuente: elaboración propia.

- **Gestionar Motivos de Salida**

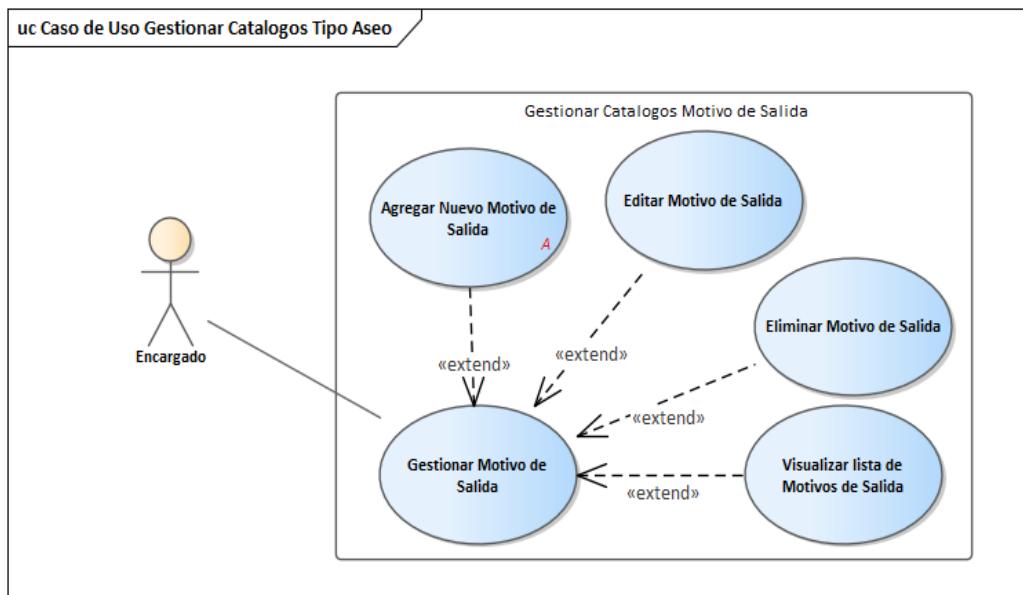


Ilustración 10 Caso de uso gestión Motivo de salida

Fuente: elaboración propia.

- **Gestionar Cuartos**

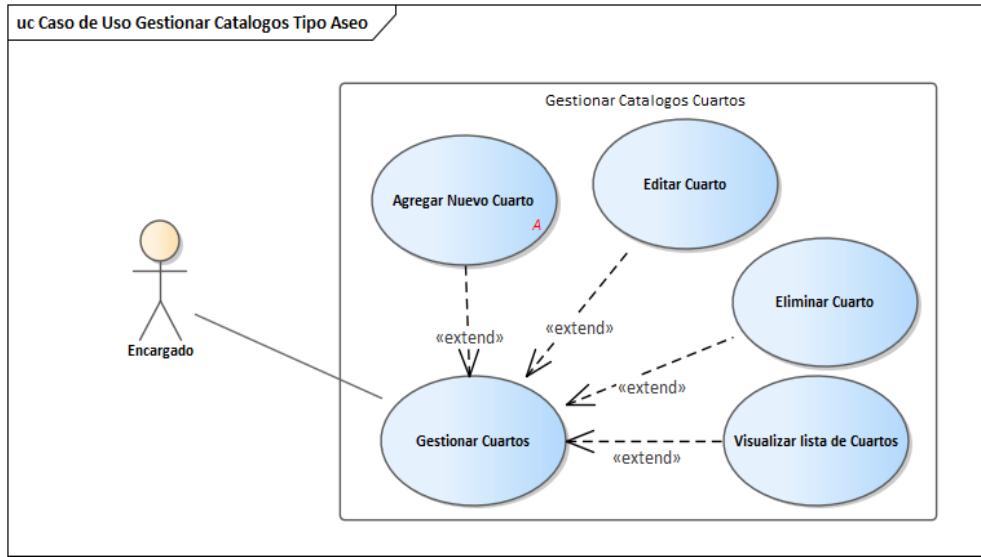


Ilustración 11 Caso de uso gestión cuartos

Fuente: elaboración propia.

- **Gestionar Carreras**

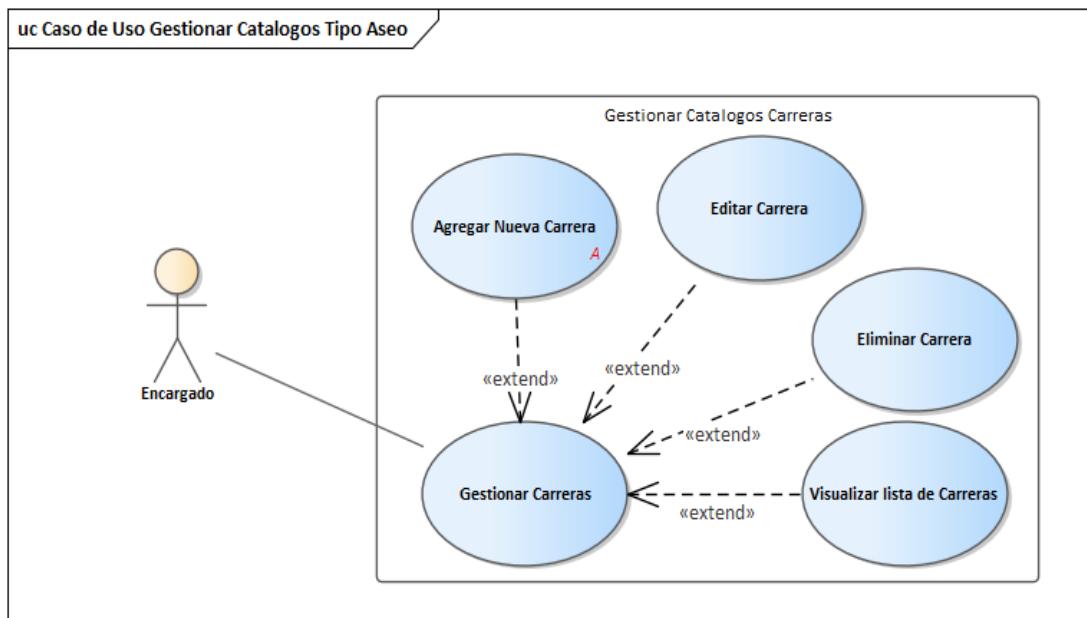


Ilustración 12 Caso de uso Gestión carreras

Fuente: elaboración propia.

- **Gestionar Recintos**

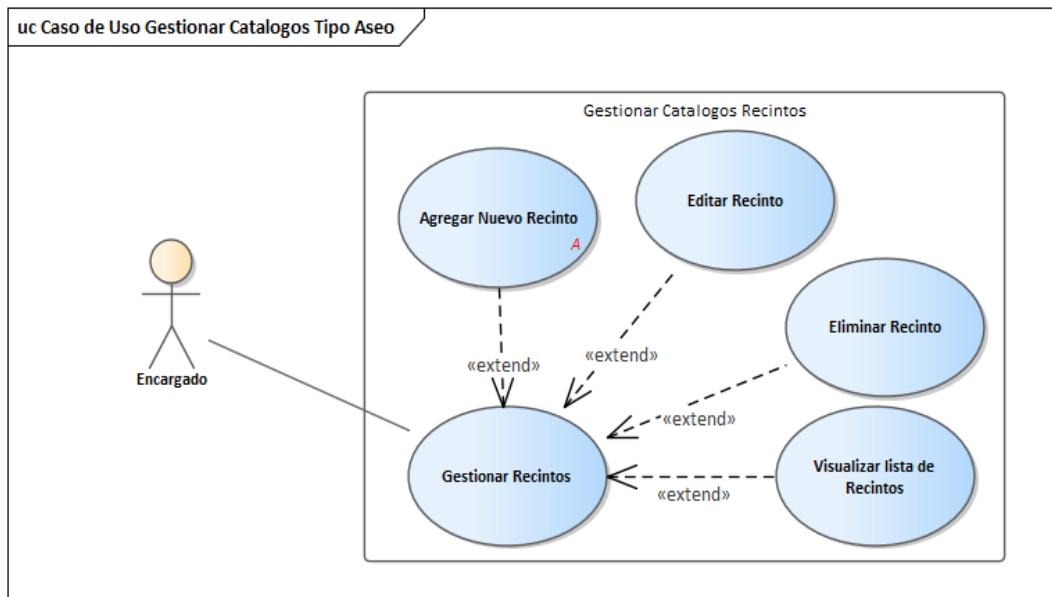


Ilustración 13 Caso de uso Gestión recintos

Fuente: elaboración propia.

7.1.3. Detalles Casos de Uso Modelados

En este acápite se aborda la documentación de los casos de usos que se especificaron con detalles textuales y flujos de eventos para el sistema web de asignación y monitoreo de roles de aseo para la residencia estudiantil de la UNI.

- **Gestionar acceso al sistema (Inicio de sesión)**

Tabla 14 Detalle Caso de uso Inicio de sesión

Inicio de Sesión		
Actores	Encargado, Administrador	
ID	UC-001	
Descripción	El sistema valida si un usuario puede o no entrar a las funcionalidades de acuerdo a las credenciales ingresadas.	
Flujo	Actor	Sistema
	1. Inicia cuando el actor ingresa a la página web	2. Muestra formulario de inicio de sesión
	3. Ingresa Usuario y contraseña	4. Verifica usuario y contraseña ingresados.
		5. Muestra las funciones de acuerdo al usuario verificado.
Flujo Alternativo	5.a. Mensaje de datos inválidos y se reinicia paso 3	

Fuente: elaboración propia

- **Nuevo estudiante**

Tabla 15 Detalle Caso de uso Nuevo estudiante

Ingresar Nuevo Estudiante											
Actores	Encargado										
ID	UC-002										
Descripción	Ingresar un nuevo estudiante al sistema.										
Flujo	<table border="1"> <thead> <tr> <th>Actor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1. Inicia cuando selecciona la opción de nuevo estudiante</td> <td>2. Muestra formulario con campos a llenar.</td> </tr> <tr> <td>3. Ingresa información en el formulario cargado por sistema</td> <td>4. Presenta cuartos disponibles por año de carnet ingresado.</td> </tr> <tr> <td>5. Presiona botón guardar.</td> <td>6. Valida información ingresada y campos requeridos.</td> </tr> <tr> <td></td> <td>7. Muestra mensaje de acción exitosa y redirige a lista de estudiantes en sistema.</td> </tr> </tbody> </table>	Actor	Sistema	1. Inicia cuando selecciona la opción de nuevo estudiante	2. Muestra formulario con campos a llenar.	3. Ingresa información en el formulario cargado por sistema	4. Presenta cuartos disponibles por año de carnet ingresado.	5. Presiona botón guardar.	6. Valida información ingresada y campos requeridos.		7. Muestra mensaje de acción exitosa y redirige a lista de estudiantes en sistema.
Actor	Sistema										
1. Inicia cuando selecciona la opción de nuevo estudiante	2. Muestra formulario con campos a llenar.										
3. Ingresa información en el formulario cargado por sistema	4. Presenta cuartos disponibles por año de carnet ingresado.										
5. Presiona botón guardar.	6. Valida información ingresada y campos requeridos.										
	7. Muestra mensaje de acción exitosa y redirige a lista de estudiantes en sistema.										
Flujo Alternativo	4.a. El sistema presenta error al no tener cuartos para el año del carnet. 7.a. El sistema muestra campos con datos erróneos y repite el paso 3.										
Pre condición	UC-001										

Fuente: elaboración propia

- **Editar estudiante**

Tabla 16 Detalle Caso de uso Editar estudiante

Editar Estudiantes									
Actores	Encargado								
ID	UC-003								
Descripción	Editar datos de estudiante en sistema.								
Flujo	<table border="1"> <thead> <tr> <th>Actor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1. Inicia cuando selecciona la opción de gestión estudiante</td> <td>2. Carga lista de estudiantes en sistema.</td> </tr> <tr> <td>3. Selecciona opciones y editar.</td> <td>4. Carga formulario con datos del estudiante</td> </tr> <tr> <td>5. Modifica datos.</td> <td></td> </tr> </tbody> </table>	Actor	Sistema	1. Inicia cuando selecciona la opción de gestión estudiante	2. Carga lista de estudiantes en sistema.	3. Selecciona opciones y editar.	4. Carga formulario con datos del estudiante	5. Modifica datos.	
Actor	Sistema								
1. Inicia cuando selecciona la opción de gestión estudiante	2. Carga lista de estudiantes en sistema.								
3. Selecciona opciones y editar.	4. Carga formulario con datos del estudiante								
5. Modifica datos.									

	6. Presiona botón guardar.	7. Valida información ingresada y campos requeridos
		8. Muestra mensaje de acción exitosa y redirige a lista de estudiantes en sistema.
Flujo Alternativo	2.a. Muestra mensaje no hay estudiantes ingresados en el sistema. 8.a. El sistema muestra campos con datos erróneos para repetir el paso 5.	
Pre condición	UC-001	

Fuente: elaboración propia

- **Eliminar estudiante**

Tabla 17 Detalle Caso de uso Eliminar estudiante

Eliminar Estudiante		
Actores	Encargado	
ID	UC-004	
Descripción	Ingresar un nuevo estudiante al sistema.	
Flujo	Actor	Sistema
	1. Inicia cuando selecciona la opción de gestión estudiante	2. Carga lista de estudiantes en sistema.
	3. Selecciona opciones y eliminar.	4. Solicita confirmación de la acción eliminar
	5. Confirma acción.	6. Coloca el registro en estado eliminado.
Flujo Alternativo	2.a. Muestra mensaje no hay estudiantes ingresados en el sistema. 5.a. El actor cancela la acción.	
Pre condición	UC-001	

Fuente: elaboración propia

- **Visualizar información a detalle de estudiante**

Tabla 18 Detalle Caso de uso Visualizar detalle de estudiante

Visualizar Detalle de Estudiante		
Actores	Encargado	
ID	UC-005	
Descripción	El actor podrá visualizar información a detalle de un estudiante.	

Flujo	Actor	Sistema
	1. Inicia cuando selecciona la opción de Gestión estudiante	2. Muestra los estudiantes en sistema con su información relevante en una tabla.
	3. Hace clic en la caja de texto buscar	
	4. Ingresa carnet del estudiante para filtrar información	5. Muestra únicamente el registro del estudiante asociado al número de carnet.
	6. Hace clic en botón opciones y ver datos.	7. Muestra un modal con información más relevante del estudiante.
	8. Presiona botón cerrar para visualizar información de todos los estudiantes	
Flujo Alternativo	3.a. El usuario no filtra la información. 5.a. El sistema retorna mensaje que no se encontró registro asociado a la búsqueda y se reinicia el paso 2. 8.a. El usuario no presiona cerrar el modal.	
Pre condición	UC-001	

Fuente: elaboración propia

- **Cambio a estudiante de cuarto**

Tabla 19 Detalle Caso de uso Cambio a estudiante de cuarto

Cambio de cuarto a estudiante		
ID	UC-006	
Descripción	El actor podrá asignar una nueva habitación a un estudiante en específico.	
Flujo	Actor	Sistema
	1. Inicia cuando selecciona Gestión Estudiantes, opción cambio de cuarto.	2. Muestra los campos requeridos.
	3. Selecciona el número de cuarto a asignar.	4. Verifica la disponibilidad de cuarto.
	5. Ingresa Comentario acerca de la razón del cambio y clic guardar cambio	6. Verifica que la descripción y habitación nueva no esté vacía
		7. Actualiza información

Flujo Alternativo	4.a. Muestra mensaje no hay cuartos disponibles 5.a. el usuario puede cancelar operación 6.a. Mensaje de campos vacíos y reinicia el paso 2. 7.a. El sistema muestra mensaje el estudiante no ha sido cambiado de cuarto y sigue con el flujo.
Pre condición	UC-001

Fuente: elaboración propia

- Reportar salida de estudiante**

Tabla 20 Detalle Caso de uso Reportar salida de estudiante

Reportar Salida Estudiante		
Actores	Encargado	
ID	UC-007	
Descripción	El actor podrá generar un reporte de salida de un estudiante determinado, cuando este lo solicite.	
Flujo	Actor	Sistema
	1. Inicia cuando el actor selecciona la opción Gestión Estudiante.	2. Muestra los estudiantes en sistema con su información relevante en una tabla.
	3. Hace clic en la caja de texto buscar	
	4. Ingresa carnet del estudiante para filtrar información	5. Muestra únicamente el registro del estudiante asociado al número de carnet.
	6. Selecciona opciones y Entradas/Salidas.	7. Genera Modal con datos requeridos
	8. Selecciona un motivo de salida y una descripción y guarda.	9. Verifica que los campos han sido llenados con la información necesaria.
		10. Genera reporte de salida con los datos ingresados previamente
	11. Imprime reporte	
	8.a. La operación puede ser cancelada 5.a. El sistema mostrara un aviso de que no hay estudiante asociado al número de carnet y repite paso 2. 9.a. El sistema muestra un aviso que hay campos vacíos y vuelve al paso 8	
Pre condición	UC-001	

Fuente: elaboración propia

- **Reportar regreso de estudiante**

Tabla 21 Detalle Caso de uso Reportar regreso de estudiante

Reportar retorno de Estudiante		
Actores	Encargado	
ID	UC-008	
Descripción	El actor podrá registrar retorno del estudiante a la Residencia Estudiantil	
Flujo	Actor	Sistema
	1. Inicia cuando el actor selecciona apartado de estudiantes ausentes 3. Busca estudiante a reportar regreso 4. Presiona opciones y Entrada/Salida. 6. Confirma acción.	2. Muestra lista de estudiantes ausentes en residencia estudiantil 5. Solicita confirmación para reportar regreso. 7. Guarda fecha de regreso en sistema 8. Cambia estado del estudiante como activo en residencia.
Flujo Alternativo	2.a. Muestra mensaje no hay estudiantes asociados al carnet ingresado 6.a. El actor puede cancelar la operación.	
Pre condición	UC-001, UC-007	

Fuente: elaboración propia

- **Visualizar historial de cambios de cuarto de estudiante**

Tabla 22 Detalle Caso de uso Visualizar historial de cambio de cuarto de estudiante

Visualizar historial de cambio de cuarto de Estudiante		
Actores	Encargado	
ID	UC-009	
Descripción	El actor podrá visualizar historial de los cambios de cuarto del estudiante	
Flujo	Actor	Sistema
	1. Inicia cuando selecciona la opción de Gestión estudiante 3. Hace clic en la caja de texto buscar	2. Muestra los estudiantes en sistema con su información relevante en una tabla.

	4. Ingresa carnet del estudiante para filtrar información 6. Hace clic en botón opciones e historial de cambios de cuarto. 8. Presiona botón cerrar para visualizar información de todos los estudiantes	5. Muestra únicamente el registro del estudiante asociado al número de carnet. 7. Muestra un modal con información de los cambios de cuartos asociados al estudiante.
Flujo Alternativo	3.a. El usuario no filtra la información 5.a. El sistema retorna mensaje que no se encontró registro asociado al carnet y se reinicia el paso 2 8.a. El usuario no presiona cerrar el modal.	
Pre condición	UC-001	

Fuente: elaboración propia

- **Visualizar historial de salidas de estudiante**

Tabla 23 Detalle Caso de uso Visualizar historial de salidas de estudiante

Visualizar historial de salida de Estudiante		
Actores	Encargado	
ID	UC-010	
Descripción	El actor podrá visualizar historial de reportes de salidas del estudiante.	
Flujo	Actor	Sistema
	1. Inicia cuando selecciona la opción de Gestión estudiante	2. Muestra los estudiantes en sistema con su información relevante en una tabla.
	3. Hace clic en la caja de texto buscar	
	4. Ingresa carnet del estudiante para filtrar información	5. Muestra únicamente el registro del estudiante asociado al número de carnet.
	6. Hace clic en botón opciones e historial de salidas.	7. Muestra un modal con información las salidas reportadas asociadas al estudiante.
	8. Presiona botón cerrar para visualizar información de todos los estudiantes	

Flujo Alternativo	3.a. El usuario no filtra la información 5.a. El sistema retorna mensaje que no se encontró registro asociado al carnet y se reinicia el paso 2 8.a. El usuario no presiona cerrar el modal.
Pre condición	UC-001

Fuente: elaboración propia

- **Visualizar calendario de asignaciones de aseo**

Tabla 24 Detalle Caso de uso Calendario de asignaciones

Calendario de asignaciones de aseo		
Actores	Encargado	
ID	UC-011	
Descripción	El actor podrá visualizar las asignaciones de aseos en un calendario de actividades.	
Flujo	Actor	Sistema
	1. Inicia cuando selecciona el Módulo Asignaciones de aseo y opción calendario	2. Muestra asignaciones agrupadas en los días del calendario.
	3. Selecciona tag determinado en el calendario	4. Muestra nombre del estudiante y el rol de aseo que debe cumplir.
	5. El actor da clic en cualquier día del calendario para generar una nueva asignación de aseo.	6. Genera formulario para el ingreso de los datos.
	7. Llena formulario y da clic en guardar.	8. Genera nueva asignación de aseo y notifica vía telegram al estudiante.
Flujo Alternativo	2.a. muestra mensaje no hay asignaciones. 3.a. El actor no selecciona ningún tag. 5.a. El actor no genera una nueva asignación. 7.a. El actor puede cancelar la operación.	
Pre condición	UC-001	

Fuente: elaboración propia

- **Nueva asignación de aseo**

Tabla 25 Detalle Caso de uso Nueva asignación de aseo

Asignar actividad de aseo		
Actores	Encargado	
ID	UC-012	
Descripción	El actor podrá asignar un rol de aseo a un estudiante específico	
Flujo	Actor	Sistema
	1. Inicia cuando el actor selecciona Nueva asignación.	2. Genera formulario necesario para la asignación
		3. Muestra las habitaciones con estudiantes
	4. Selecciona un N.º de cuarto	5. Muestra los estudiantes disponibles
	6. Selecciona un estudiante	
	7. Selecciona rol de aseo a asignar	
	8. Selecciona una fecha de finalización para el aseo asignado	
	9. Clic en botón Guardar	10. Verifica que no haya campos vacíos.
		11. Guarda Asignaciones en sistema
		12. Notifica al estudiante vía mensaje en telegram con detalle de la asignación.
Flujo Alternativo	12.a. Muestra mensaje de error al notificar. 10.a. El sistema mostrara un aviso que hay campos vacíos. 9.a. El actor puede cancelar la operación, se reinicia el caso de uso.	
Pre condición	UC-001	

Fuente: elaboración propia

- **Editar asignación de aseo**

Tabla 26 Detalle Caso de uso editar asignación de aseo

Editar Asignación de Aseo		
Actores	Encargado	
ID	UC-013	
Descripción	El actor podrá modificar asignación de aseo a un estudiante específico	
Flujo	Actor	Sistema
	1. Inicia cuando el actor selecciona gestión asignaciones.	2. Muestra lista de asignaciones.
	3. Selecciona un registro, presiona opciones y editar	4. Muestra formulario con detalle de la asignación.
	5. Realiza actualización de la asignación y realiza clic en guardar.	6. Verifica estudiantes con menos de 6 asignaciones.
		7. Verifica que no haya campos vacíos.
		8. Actualiza Asignación en sistema.
		9. Notifica al estudiante vía mensaje en telegram con detalle de la actualización de asignación.
Flujo Alternativo	2.a. Muestra mensaje no hay asignaciones. 8.a. El sistema mostrara un aviso que hay campos vacíos y regresa al paso 5 9.a. Muestra mensaje de error al notificar.	
Pre condición	UC-001, UC-012	

Fuente: elaboración propia

- **Visualizar detalle de asignación**

Tabla 27 Detalle Caso de uso visualizar detalle de asignación

Visualizar Detalle de asignación		
Actores	Encargado	
ID	UC-014	
Descripción	El actor podrá visualizar detalle de asignación.	
Flujo	Actor	Sistema
	1. Inicia cuando selecciona la opción Gestión asignaciones	2. Muestra asignaciones de aseo en sistema con su

		información relevante en una tabla.
	3. Hace clic en la caja de texto buscar	
	4. Ingresa dato a filtrar.	5. Muestra únicamente el registro del estudiante asociado a la búsqueda.
	6. Hace clic en botón opciones y ver datos.	7. Muestra un modal con información de la asignación de aseo.
	8. Presiona botón cerrar para visualizar información de todas las asignaciones.	
Flujo Alternativo	2.a. El sistema indica mensaje que no hay registros de asignaciones de aseo. 3.a. El usuario no filtra la información. 5.a. El sistema retorna mensaje que no se encontró registro asociado a la búsqueda reinicia el paso 2. 8.a. El usuario no presiona cerrar el modal.	
Pre condición	UC-001, UC-012	

Fuente: elaboración propia

- **Eliminar asignación de aseo**

Tabla 28 Detalle Caso de uso Eliminar asignación de aseo

Eliminar asignación de aseo		
Actores	Encargado	
ID	UC-015	
Descripción	Elimina registro de asignación de aseo.	
Flujo	Actor	Sistema
	1. Inicia cuando selecciona la Gestión de asignaciones	2. Muestra asignaciones de aseo en sistema con su información relevante en una tabla.
	3. Selecciona opciones y eliminar.	4. Solicita confirmación de la acción eliminar
	5. Confirma acción.	6. Coloca el registro en estado eliminado.
Flujo Alternativo	2.a. Muestra mensaje no hay asignaciones ingresadas en el sistema. 5.a. El actor cancela la acción.	
Pre condición	UC-001, UC-012	

Fuente: elaboración propia

- **Panel Comedor**

Tabla 29 Detalle Caso de uso Panel comedor

Visualizar Estadísticas Comedor		
Actores	Encargado	
ID	UC-016	
Descripción	El actor podrá visualizar estadísticas de desayunos, almuerzos y cenas de residentes por carrera	
Flujo	Actor	Sistema
	1. Inicia cuando selecciona la opción panel comedor	2. Muestra estadísticas de tiempos de comida a preparar por carrera en diferentes graficas.
	3. Hace clic en generar reporte y selecciona un rango de fechas.	
	4. Confirma generación de reporte.	5. Genera reporte en Excel.
Flujo Alternativo	3.a. El actor no genera reporte. 4.a. El actor no confirmar la acción.	
Pre condición	UC-001	

Fuente: elaboración propia

- **Envío de encuestas de comidas a preparar**

Tabla 30 Detalle Caso de uso Envío de encuestas

Envío de encuestas de solicitud de comidas		
Actores	Servidor	
ID	UC-017	
Descripción	Tarea programada para el envío de encuesta vía Telegram a residentes	
Flujo	Actor	Sistema
	1. Inicia cuando el servidor detecta que la hora es igual a la hora de ejecución.	2. Obtiene lista de grupos de telegram.
		3. Envía encuesta de solicitud de comida a los grupos de estudiantes configurados.
	4. Ejecuta tarea para cerrar encuesta después de 30 minutos.	5. Cierra encuesta para calcular total de Almuerzos, cenas y desayuno a solicitar.
Flujo Alternativo	3.a. No envía encuesta por error de conexión.	
Pre condición	Hora igual a la de ejecución configurada.	

Fuente: elaboración propia

- Agregar nuevo cuarto

Tabla 31 Detalle Caso de uso Agregar nuevo cuarto

Agregar nuevo cuarto		
Actores	Encargado	
ID	UC-018	
Descripción	El usuario podrá guardar un nuevo cuarto para el catálogo.	
Flujo	Actor	Sistema
	1. Inicia cuando selecciona la opción de catálogos, cuartos y nuevo cuarto	2. Muestra formulario con campos a llenar.
	3. Ingresa información en el formulario cargado por sistema	
	4. Presiona botón guardar.	5. Valida información ingresada y campos requeridos.
		6. Muestra mensaje de acción exitosa y redirige a lista de cuartos
Flujo Alternativo	4.a. El actor puede cancelar la acción. 6.a. El sistema muestra campos con datos erróneos para repetir el paso 3.	
Pre condición	UC-001	

Fuente: elaboración propia

- Editar cuarto

Tabla 32 Detalle Caso de uso Editar cuarto

Editar cuarto		
Actores	Encargado	
ID	UC-019	
Descripción	Editar datos de un cuarto registrado en sistema.	
Flujo	Actor	Sistema
	1. Inicia cuando selecciona modulo catálogos y cuartos.	2. Carga lista de cuartos en sistema.
	3. Selecciona opciones y editar.	4. genera formulario con datos del cuarto
	5. Modifica datos y selecciona genero para el cuarto.	

	6. Presiona botón guardar.	7. Valida información ingresada y campos requeridos
		8. Muestra mensaje de acción exitosa y redirige a lista de cuartos en sistema.
Flujo Alternativo	2.a. Muestra mensaje no hay cuartos ingresados en el sistema. 8.a. El sistema muestra campos con datos erróneos para repetir el paso 5.	
Pre condición	UC-001	

Fuente: elaboración propia

- **Agregar nuevo tipo de aseo**

Tabla 33 Detalle Caso de uso Agregar nuevo tipo de aseo

Nuevo tipo de aseo		
Actores	Encargado	
ID	UC-021	
Descripción	El usuario podrá ingresar un nuevo tipo de aseo al sistema	
Flujo	Actor	Sistema
	1. Inicia cuando selecciona la opción de catálogos, tipo aseos y nuevo tipo de aseo.	2. Muestra formulario con campos a llenar.
	3. Ingresa información en el formulario cargado por sistema.	
	4. Presiona botón guardar.	5. Valida información ingresada y campos requeridos.
Flujo Alternativo		6. Muestra mensaje de acción exitosa y redirige a lista de tipos de aseo
	4.a. El actor puede cancelar la acción. 6.a. El sistema muestra campos con datos erróneos para repetir el paso 3	
Pre condición	UC-001	

Fuente: elaboración propia

- **Editar tipo de aseo**

Tabla 34 Detalle Caso de uso Editar tipo de aseo

Editar datos de tipo de aseo		
Actores	Encargado	
ID	UC-022	
Descripción	El usuario podrá editar los datos de un tipo de aseo en sistema.	
Flujo	Actor	Sistema
	1. Inicia cuando selecciona la opción catálogos y tipo aseos	2. Carga lista de tipos de aseo en sistema.
	3. Selecciona opciones y editar.	4. genera formulario con datos del tipo de aseo
	5. Modifica datos.	
	6. Presiona botón guardar.	7. Valida información ingresada y campos requeridos
		8. Muestra mensaje de acción exitosa y redirige a lista de tipos de aseo en sistema.
Flujo Alternativo	2.a. Muestra mensaje no hay tipos de aseo ingresados en el sistema. 8.a. El sistema muestra campos con datos erróneos para repetir el paso 5.	
Pre condición	UC-001	

Fuente: elaboración propia

- **Agregar nuevo usuario al sistema**

Tabla 35 Detalle Caso de uso Agregar nuevo usuario

Nuevo Usuario		
Actores	Administrador	
ID	UC-024	
Descripción	El administrador del sistema podrá seleccionar un nuevo usuario, así mismo como uno o más roles.	
Flujo	Actor	Sistema
	1. Empieza cuando el actor selecciona el módulo Usuario y opción nuevo usuario.	2. Genera formulario con campos necesarios.
		3. Verifica los roles disponibles
	4. Ingresa información en formulario y selecciona un rol.	

	5. Clic botón guardar	6. Verifica que no haya datos vacíos	
		7. Guarda Registros	
Flujo Alternativo	5.a. El actor puede cancelar la operación 3.a. El sistema muestra mensaje de que no hay roles disponibles. 6.a El sistema muestra aviso que hay campos vacíos y repite paso 4. 1a . El usuario puede no contar con los permisos necesarios.		
Pre condición	UC-001		

Fuente: elaboración propia

• **Editar usuario**

Tabla 36 Detalle Caso de uso Editar usuario

Editar datos de usuario		
Actores	Administrador	
ID	UC-025	
Descripción	El usuario podrá editar la información, así como credenciales de un usuario en sistema.	
Flujo	Actor	Sistema
	1. Inicia cuando selecciona módulo de usuarios y roles.	2. Carga lista de usuarios en sistema.
	3. Selecciona opciones y editar.	4. genera formulario con datos del usuario
	5. Modifica datos.	
	6. Presiona botón guardar.	7. Valida información ingresada y campos requeridos
		8. Muestra mensaje de acción exitosa y redirige a lista de usuarios en sistema.
Flujo Alternativo	2.a. Muestra mensaje no hay usuarios ingresados en el sistema. 8.a. El sistema muestra campos con datos erróneos para repetir el paso 5.	
Pre condición	UC-001, UC-024	

Fuente: elaboración propia

- **Eliminar usuario**

Tabla 37 Detalle Caso de uso Eliminar usuario

Eliminar usuario		
Actores	Administrador	
ID	UC-026	
Descripción	El usuario podrá eliminar registros de usuarios en sistema.	
Flujo	Actor	Sistema
	1. Inicia cuando selecciona modulo usuarios y roles	2. Carga lista usuarios en sistema.
	3. Selecciona opciones y eliminar.	4. Solicita confirmación de la acción eliminar
	5. Confirma acción.	6. Coloca el registro en estado eliminado.
Flujo Alternativo	2.a. Muestra mensaje no hay usuarios ingresados en el sistema. 5.a. El actor cancela la acción.	
Pre condición	UC-001	

Fuente: elaboración propia

- **Añadir nuevo grupo Telegram**

Tabla 38 Detalle Caso de uso Nuevo grupo de Telegram

Nuevo grupo de Telegram		
Actores	Encargado	
ID	UC-027	
Descripción	El usuario podrá ingresar un nuevo grupo de Telegram al sistema	
Flujo	Actor	Sistema
	1. Inicia cuando selecciona la opción de Grupos Telegram y nuevo grupo Telegram.	2. Muestra formulario con campos a llenar.
	3. Ingresa información en el formulario cargado por sistema y Presiona botón guardar.	4. Valida información ingresada y campos requeridos.
		5. Muestra mensaje de acción exitosa y redirige a lista de grupos de Telegram
Flujo Alternativo	4.a. El actor puede cancelar la acción. 6.a. El sistema muestra campos con datos erróneos para repetir el paso 3.	
Pre condición	UC-001	

Fuente: elaboración propia

- **Editar grupo Telegram**

Tabla 39 Detalle Caso de uso Editar grupo de Telegram

Editar grupo de Telegram									
Actores	Encargado								
ID	UC-028								
Descripción	El usuario podrá editar la información, así como credenciales de un grupo de Telegram en sistema.								
Flujo	<table border="1"> <thead> <tr> <th>Actor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1. Inicia cuando selecciona módulo de grupos Telegram.</td> <td>2. Carga lista de grupos Telegram en sistema.</td> </tr> <tr> <td>3. Selecciona opciones y editar.</td> <td>4. genera formulario con datos del grupo</td> </tr> <tr> <td>5. Modifica datos y Presiona guardar.</td> <td>6. Valida información ingresada y campos requeridos y Muestra mensaje de acción exitosa y redirige a lista de grupos telegram en sistema.</td> </tr> </tbody> </table>	Actor	Sistema	1. Inicia cuando selecciona módulo de grupos Telegram.	2. Carga lista de grupos Telegram en sistema.	3. Selecciona opciones y editar.	4. genera formulario con datos del grupo	5. Modifica datos y Presiona guardar.	6. Valida información ingresada y campos requeridos y Muestra mensaje de acción exitosa y redirige a lista de grupos telegram en sistema.
Actor	Sistema								
1. Inicia cuando selecciona módulo de grupos Telegram.	2. Carga lista de grupos Telegram en sistema.								
3. Selecciona opciones y editar.	4. genera formulario con datos del grupo								
5. Modifica datos y Presiona guardar.	6. Valida información ingresada y campos requeridos y Muestra mensaje de acción exitosa y redirige a lista de grupos telegram en sistema.								
Flujo Alternativo	2.a. Muestra mensaje no hay grupos ingresados en el sistema. 8.a. El sistema muestra campos con datos erróneos para repetir el paso 5.								
Pre condición	UC-001, UC-027								

Fuente: elaboración propia

- **Cambiar estado grupo Telegram**

Tabla 40 Detalle Caso de uso Cambiar estado grupo de Telegram

Cambiar estado grupo telegram							
Actores	Encargado						
ID	UC-028						
Descripción	El usuario podrá cambiar el estado de los grupos en sistema.						
Flujo	<table border="1"> <thead> <tr> <th>Actor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1. Inicia cuando selecciona módulo de grupos Telegram.</td> <td>2. Carga lista de grupos telegram en sistema.</td> </tr> <tr> <td>3. Selecciona opciones y cambiar estado.</td> <td>4. Cambiar estado del registro.</td> </tr> </tbody> </table>	Actor	Sistema	1. Inicia cuando selecciona módulo de grupos Telegram.	2. Carga lista de grupos telegram en sistema.	3. Selecciona opciones y cambiar estado.	4. Cambiar estado del registro.
Actor	Sistema						
1. Inicia cuando selecciona módulo de grupos Telegram.	2. Carga lista de grupos telegram en sistema.						
3. Selecciona opciones y cambiar estado.	4. Cambiar estado del registro.						
Flujo Alternativo	2.a. Muestra mensaje no hay grupos ingresados en el sistema.						
Pre condición	UC-001, UC-027						

Fuente: elaboración propia

7.1.4. Diagrama de clase

El objetivo principal de este modelo es la representación de los aspectos estáticos del sistema recoge las clases de objetos y sus asociaciones. En este diagrama se representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos.

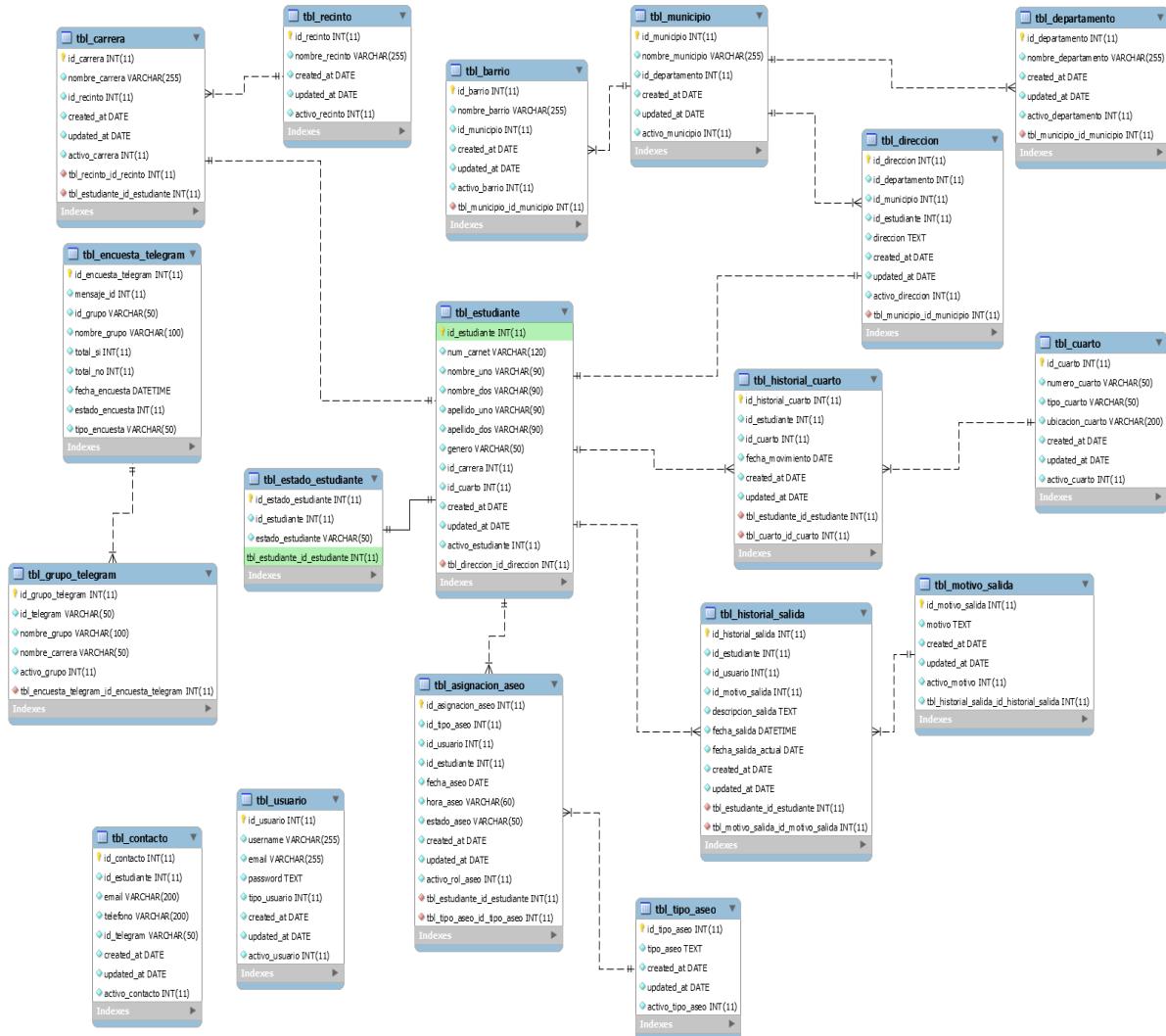


Ilustración 14 Diagrama de clases

Fuente: elaboración propia.

7.1.5. Diagrama de actividades

Los diagramas de actividades se usan para mostrar la secuencia de actividades, el flujo de trabajo desde el punto de inicio hasta el punto final detallando muchas de las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad. Los Diagramas de Actividades son útiles para el Modelado de Negocios donde se usan para detallar el proceso involucrado en las actividades de negocio. (sparxsystems, 2023).

- **Inicio de sesión**

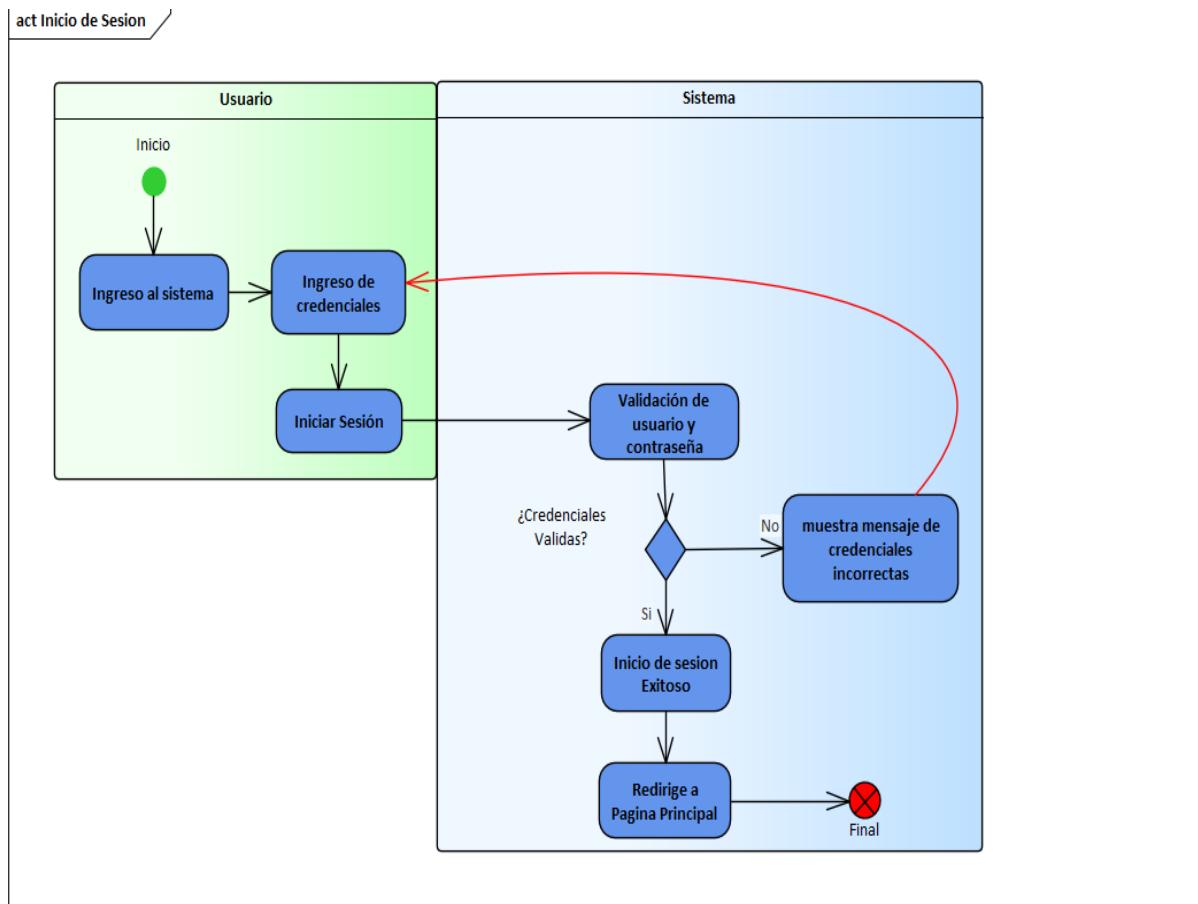


Ilustración 15 Diagrama de Actividad Inicio de sesión

Fuente: elaboración propia.

- Gestión estudiante

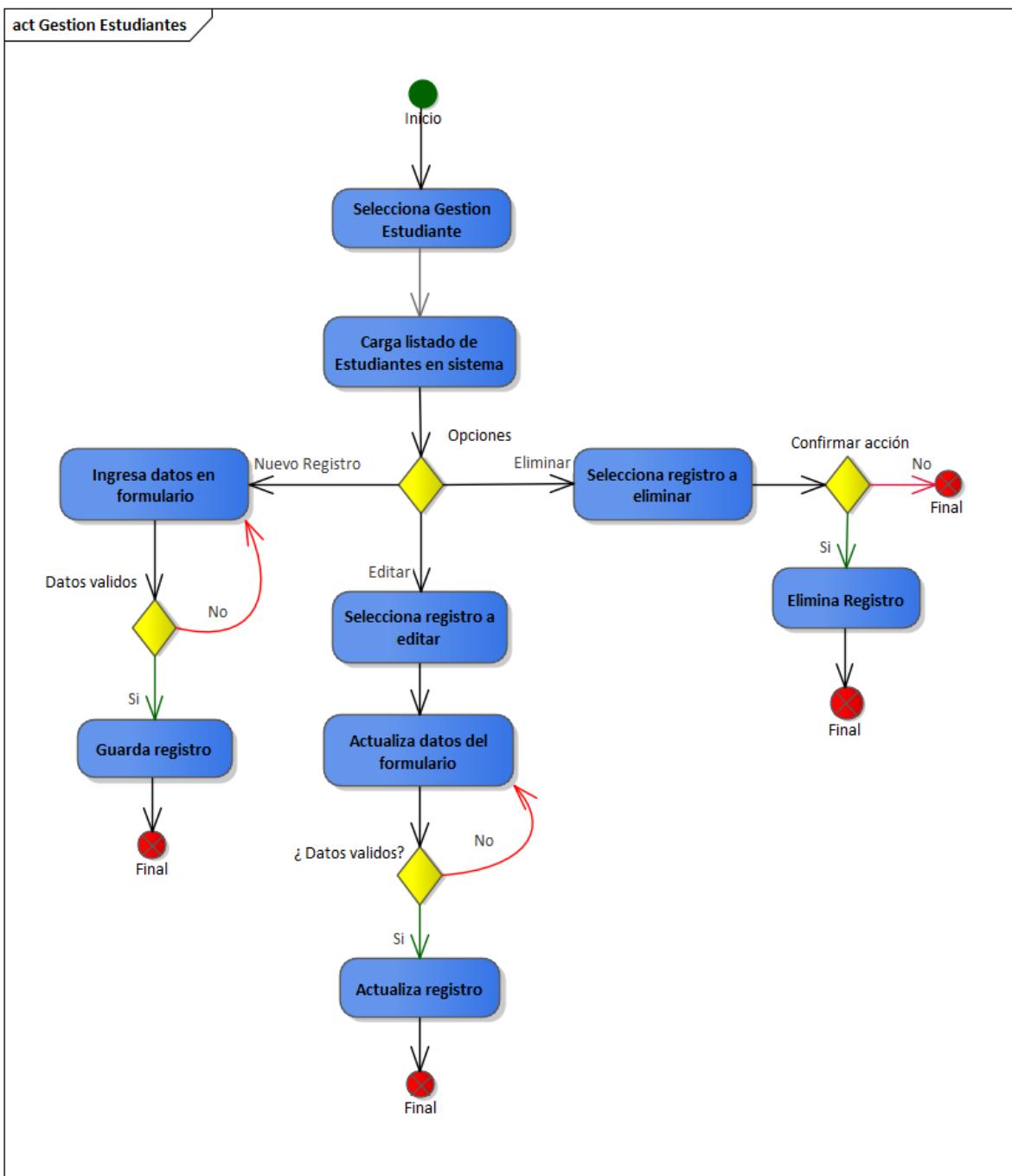


Ilustración 16 Diagrama de Actividad Gestión estudiante

Fuente: elaboración propia.

- Cambio de cuarto estudiante

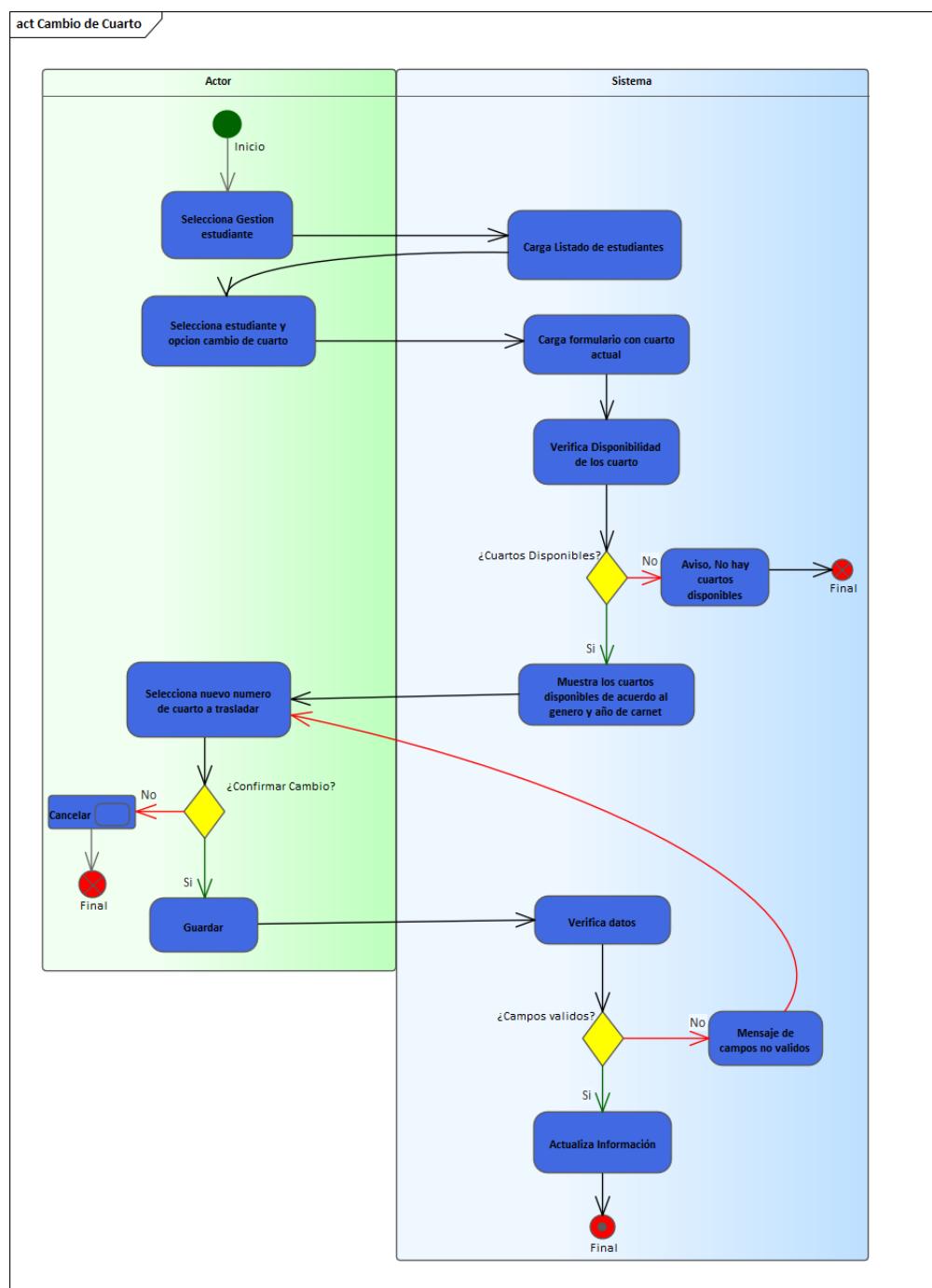


Ilustración 17 Diagrama de Actividad Cambio de cuarto estudiante

Fuente: elaboración propia.

- **Reportar Salida de estudiante**

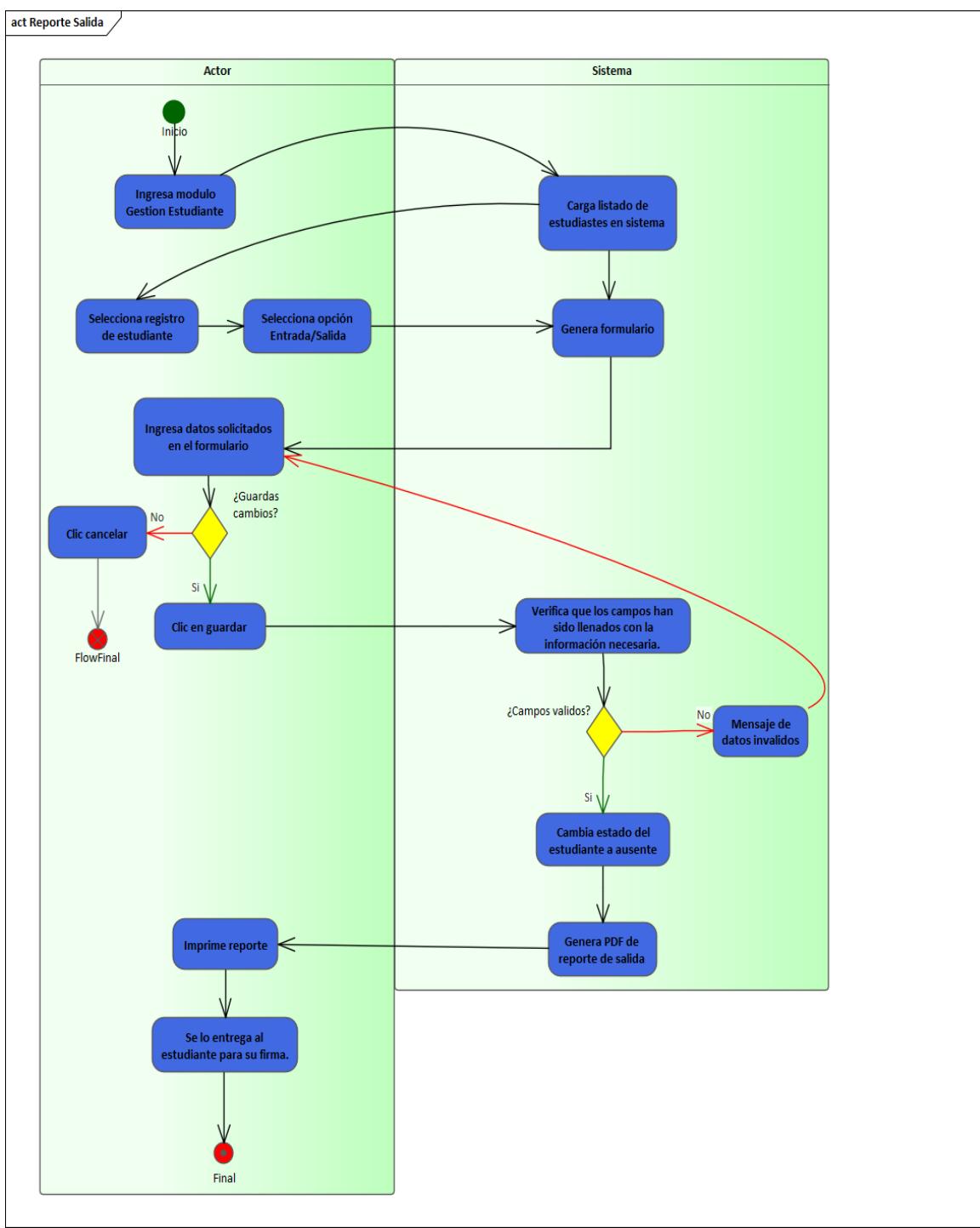


Ilustración 18 Diagrama de Actividad Reportar salida de estudiante

Fuente: elaboración propia.

- **Reportar regreso de estudiante**

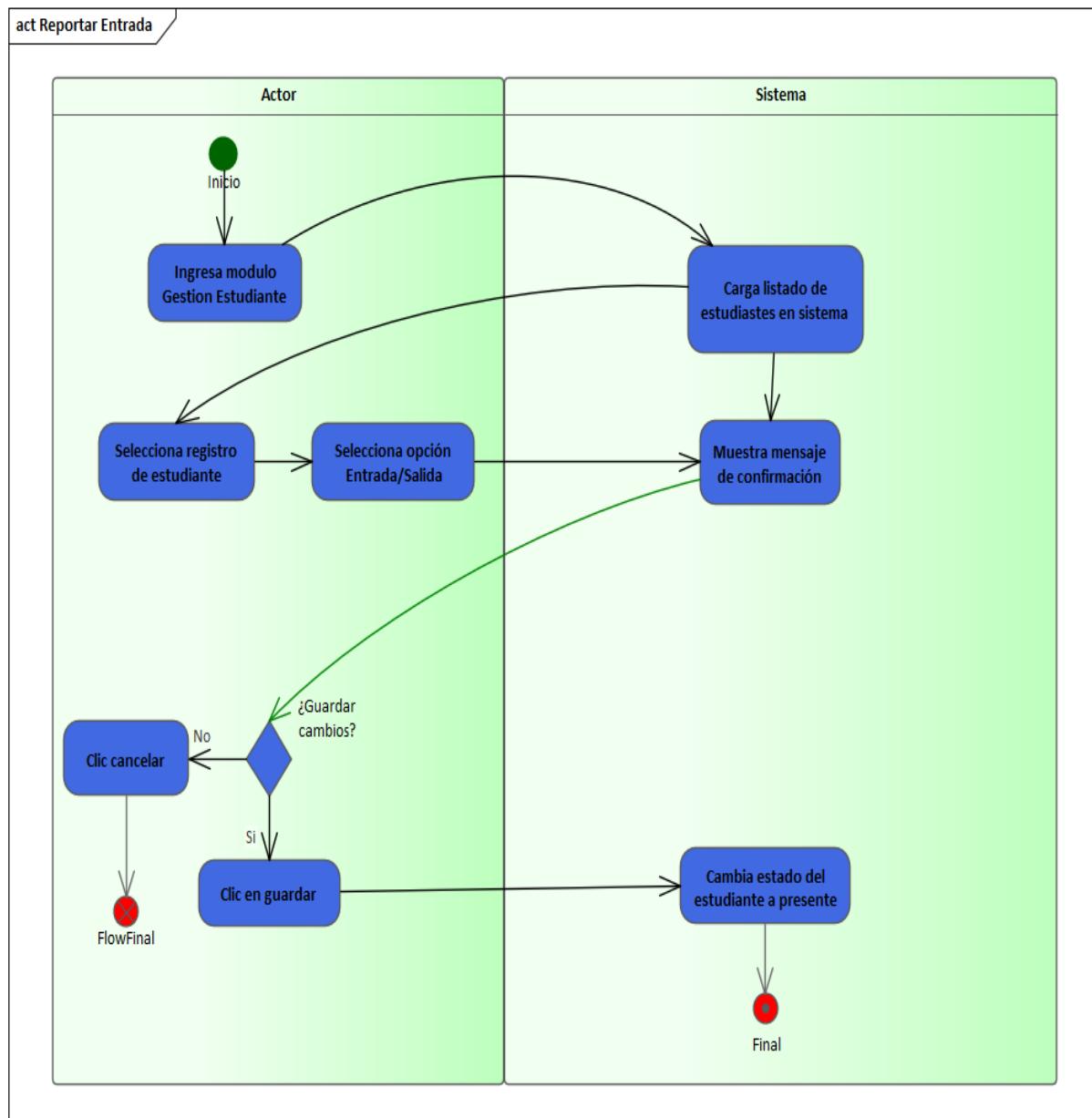


Ilustración 19 Diagrama de Actividad Reportar regreso de estudiante

Fuente: elaboración propia.

- **Historial de Cambios de cuarto e Historial de Salidas de estudiantes**

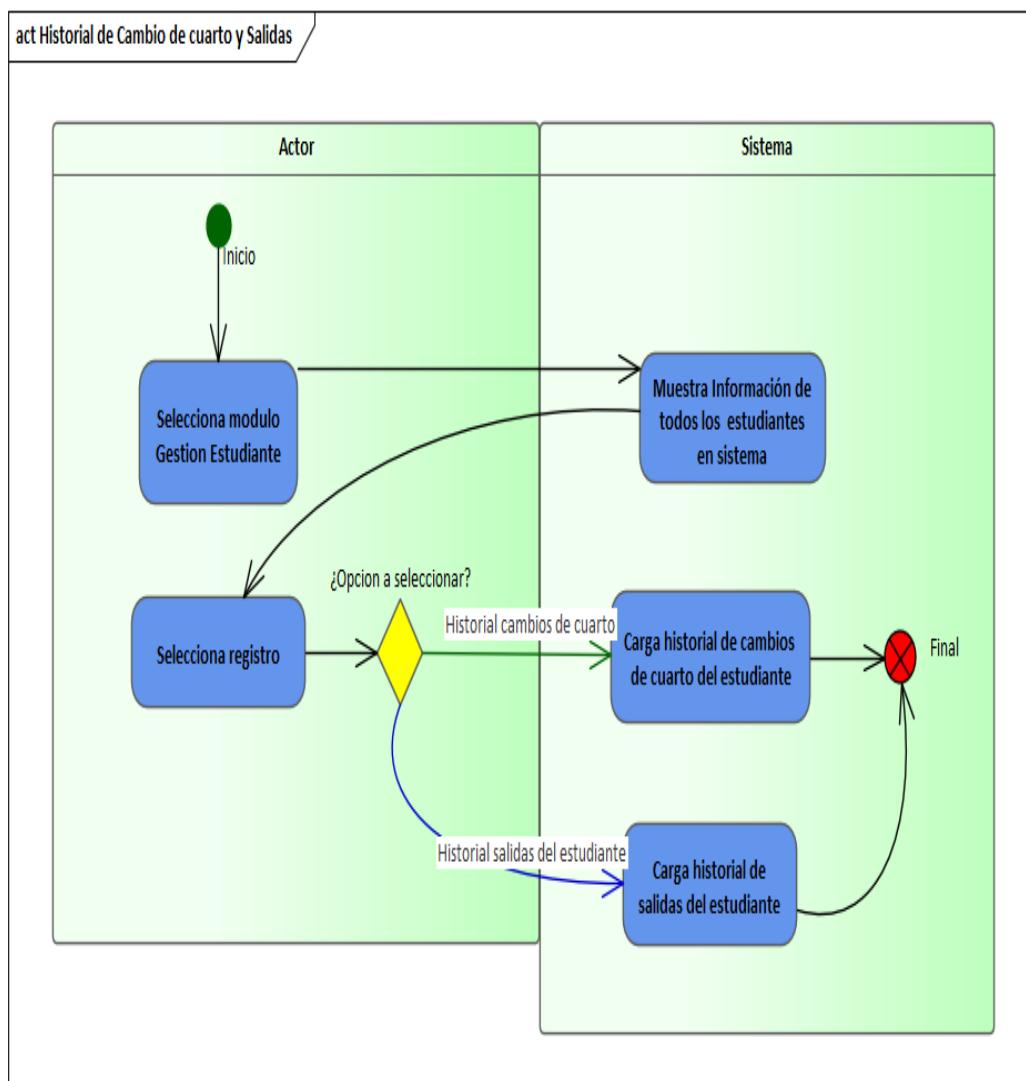


Ilustración 20 Diagrama de Actividad Historial cambio de cuarto y de salidas

Fuente: elaboración propia.

- **Visualizar estudiantes en sistema**

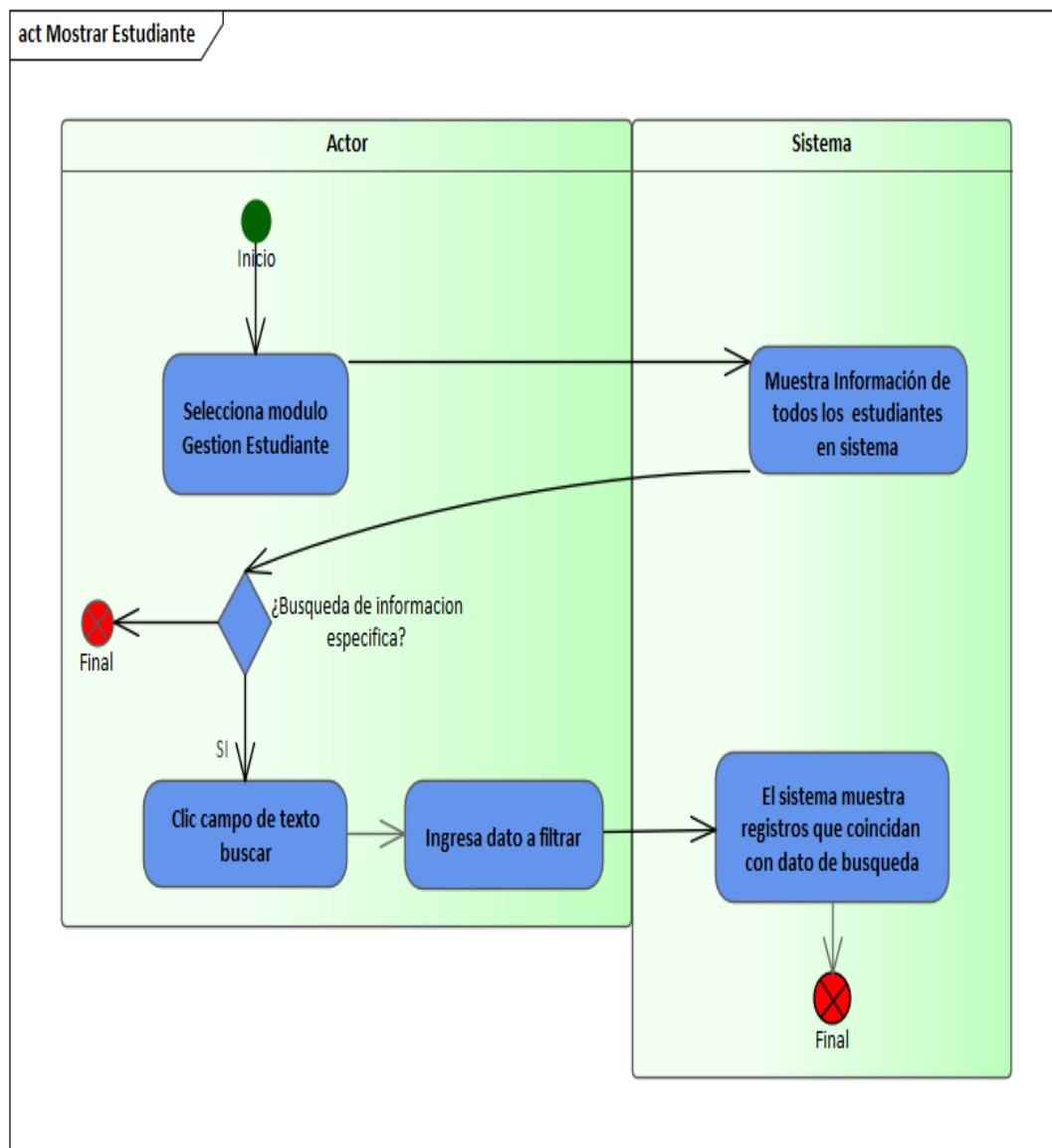


Ilustración 21 Diagrama de Actividad Visualizar estudiante en sistema

Fuente: elaboración propia.

- Asignar actividad de aseo

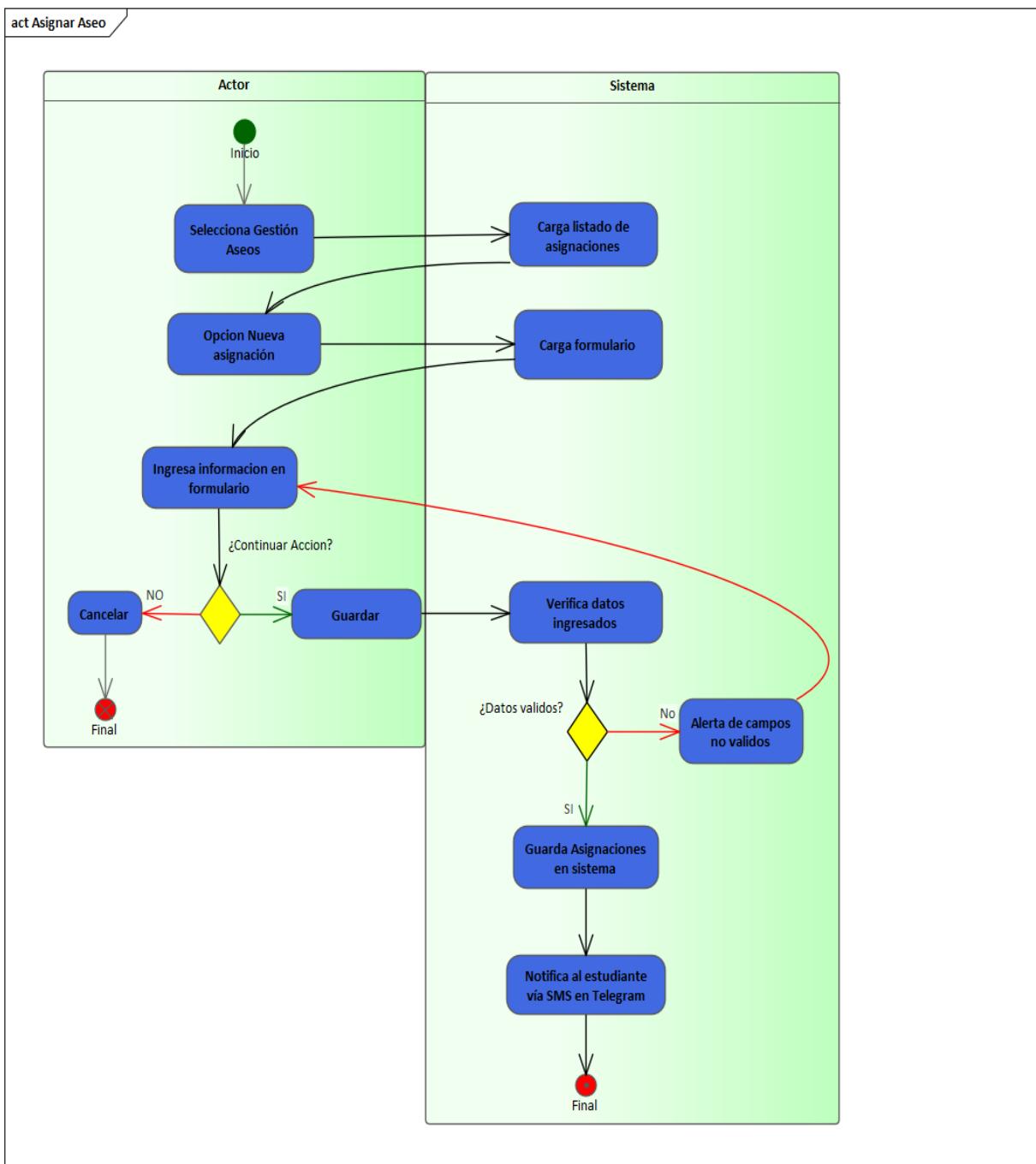


Ilustración 22 Diagrama de Actividad Asignar actividad de aseo

Fuente: elaboración propia.

- **Editar asignación de aseo**

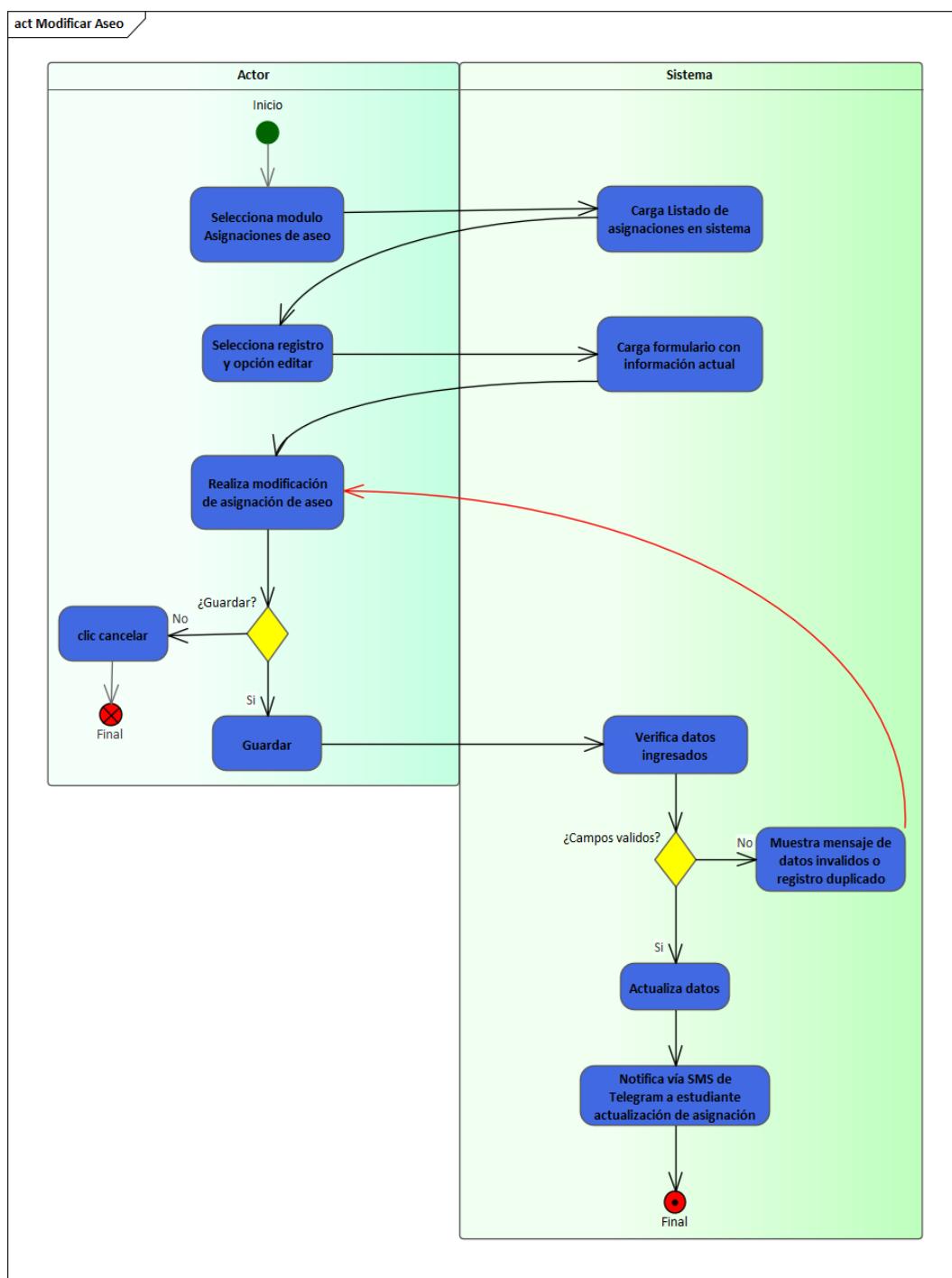


Ilustración 23 Diagrama de Actividad *Editar asignación de aseo*

Fuente: elaboración propia.

- **Eliminar asignación de aseo**

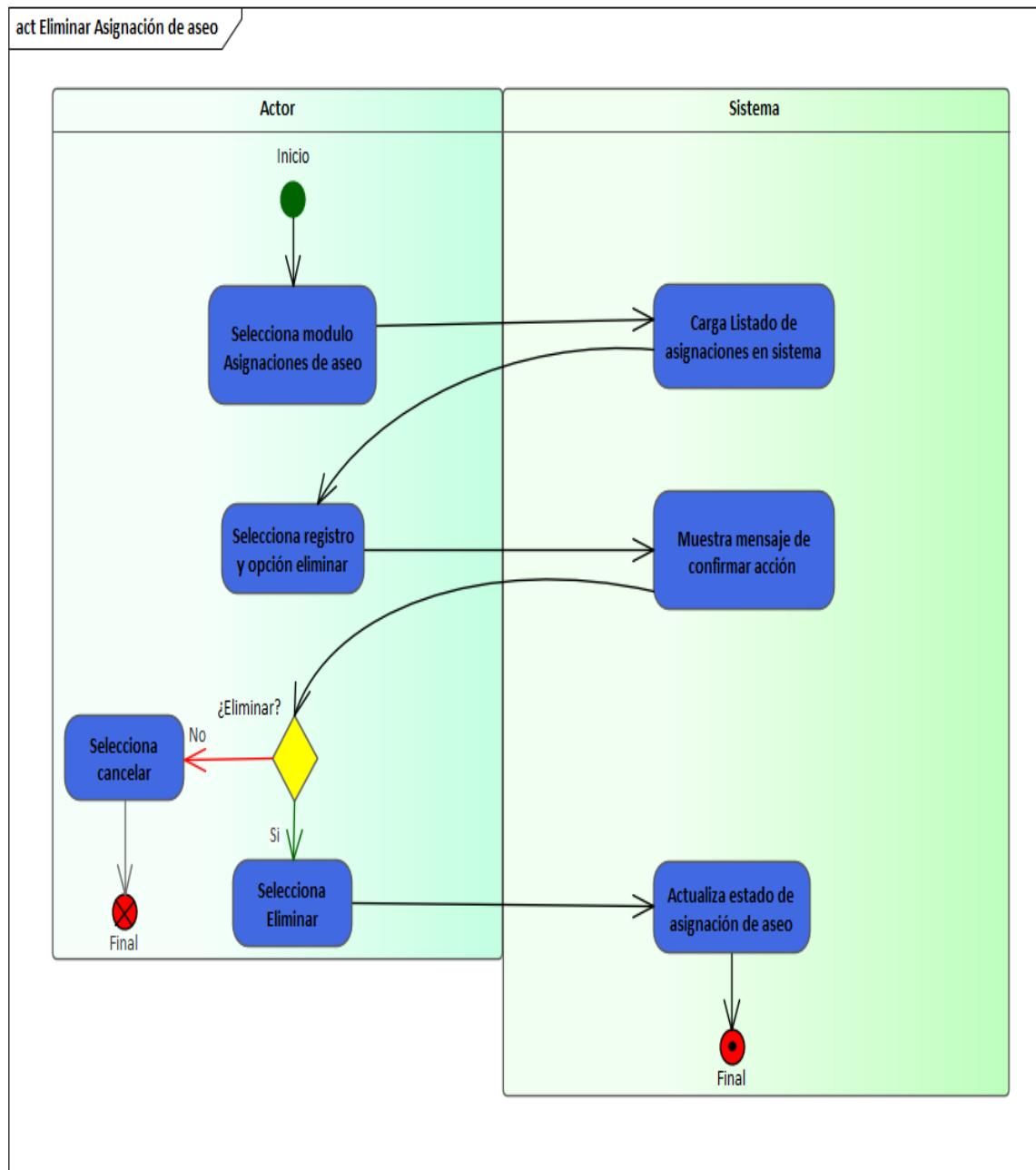


Ilustración 24 Diagrama de Actividad Eliminar asignación de aseo

Fuente: elaboración propia.

- **Calendario de asignaciones**

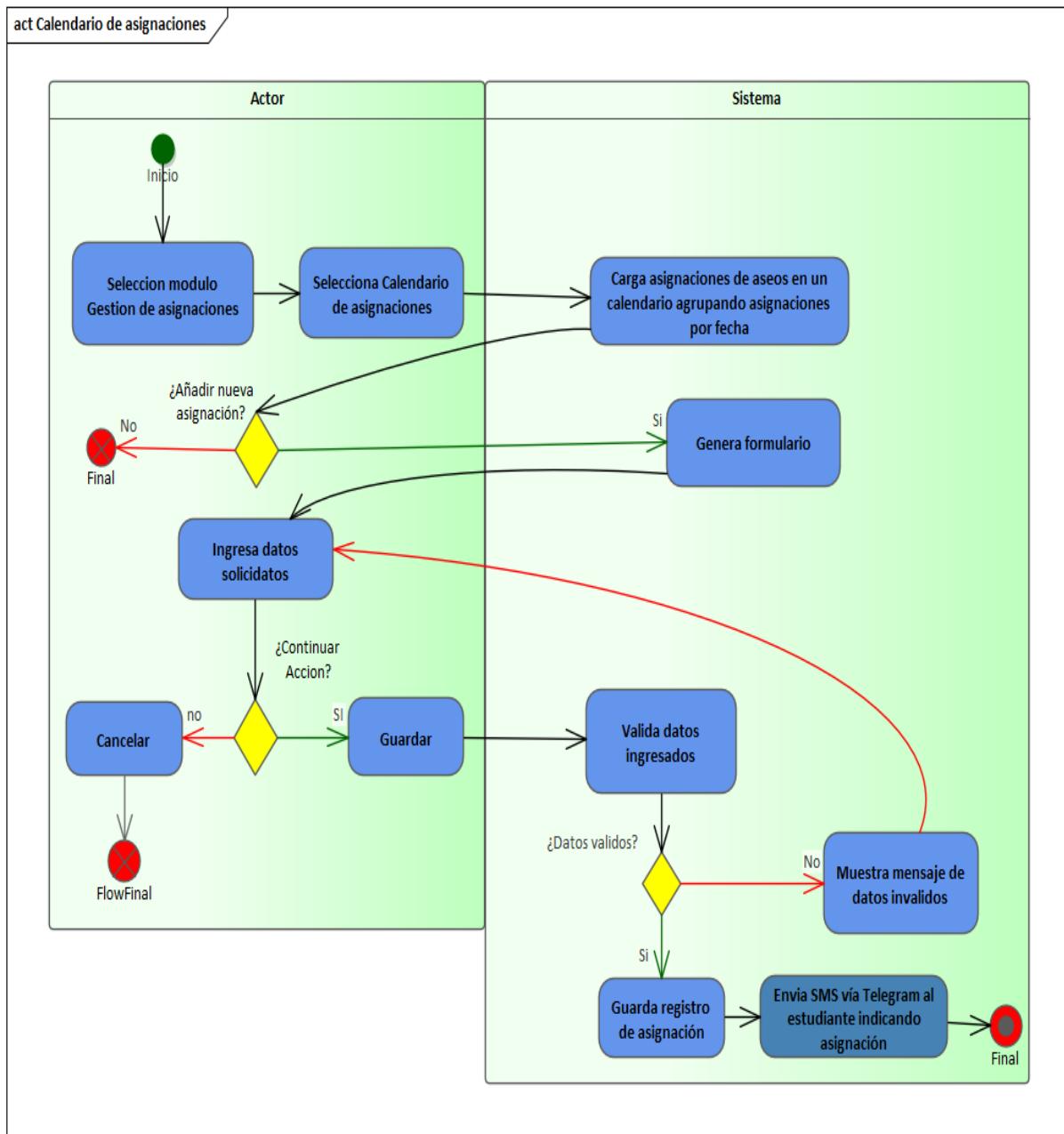


Ilustración 25 Diagrama de Actividad Calendario de asignaciones

Fuente: elaboración propia.

- Gestión Tipo de aseo

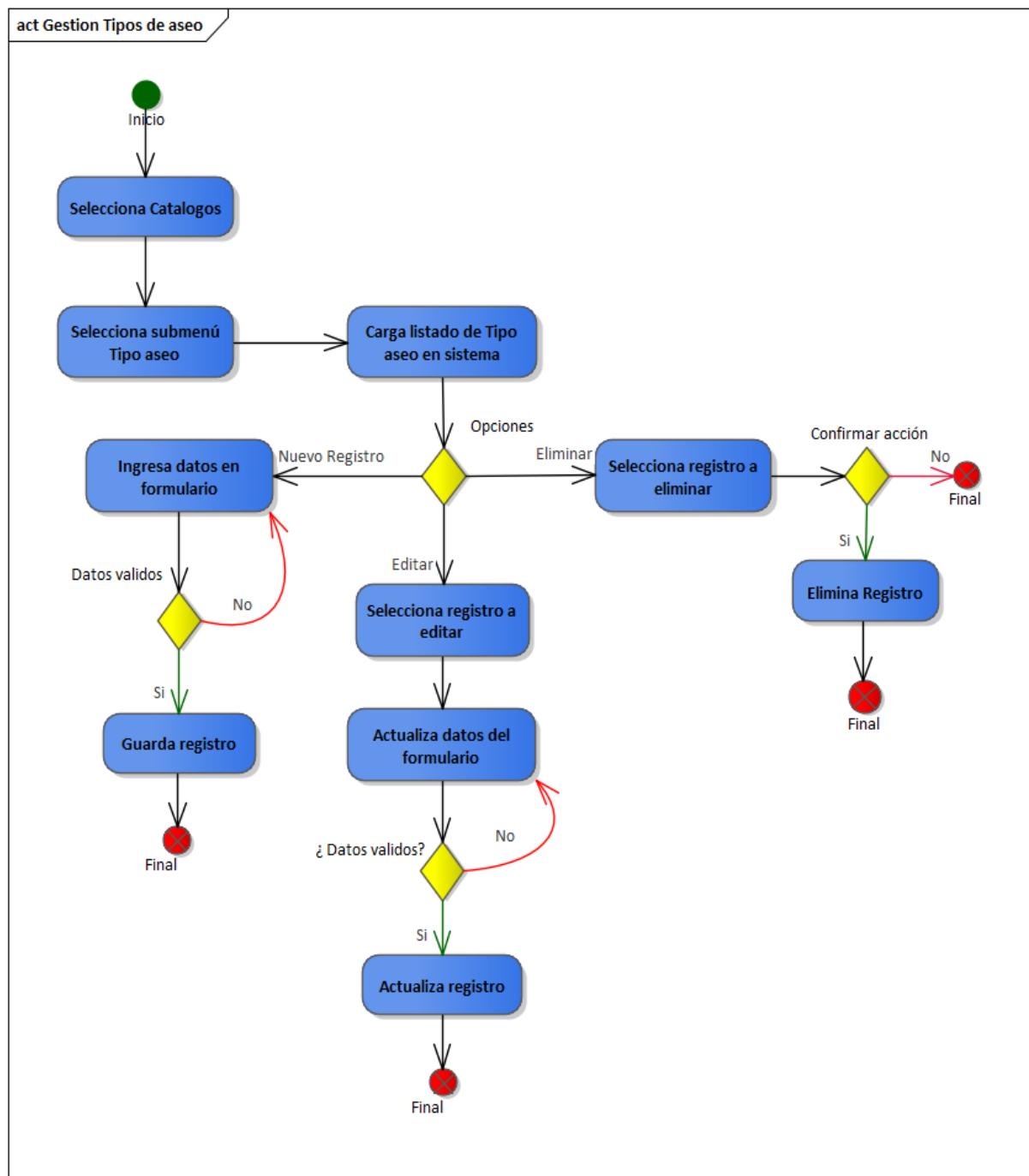


Ilustración 26 Diagrama de Actividad Gestión tipo de aseo

Fuente: elaboración propia.

- Gestión cuartos

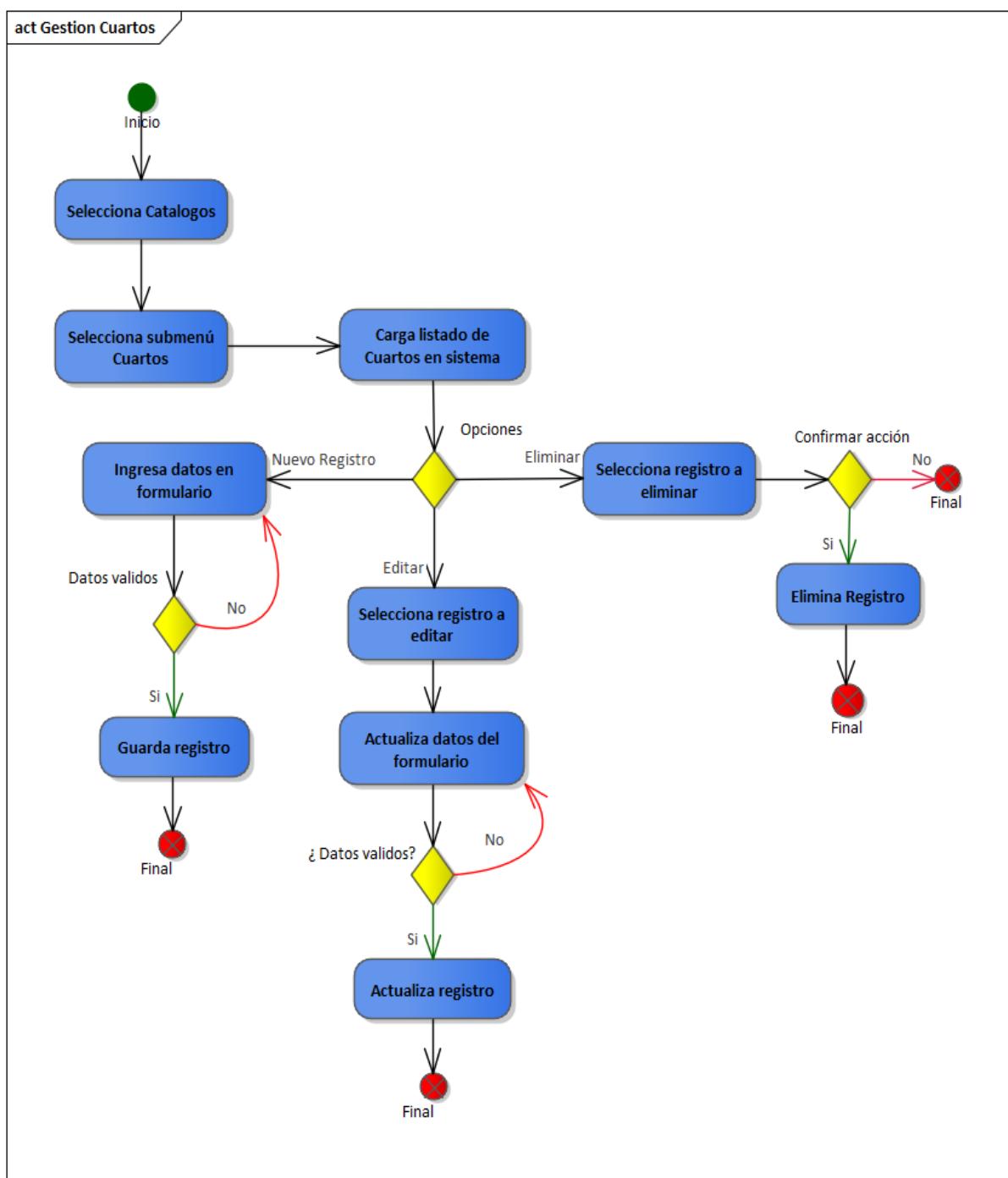


Ilustración 27 Diagrama de Actividad Gestión cuarto

Fuente: elaboración propia.

- Gestión motivos de salida

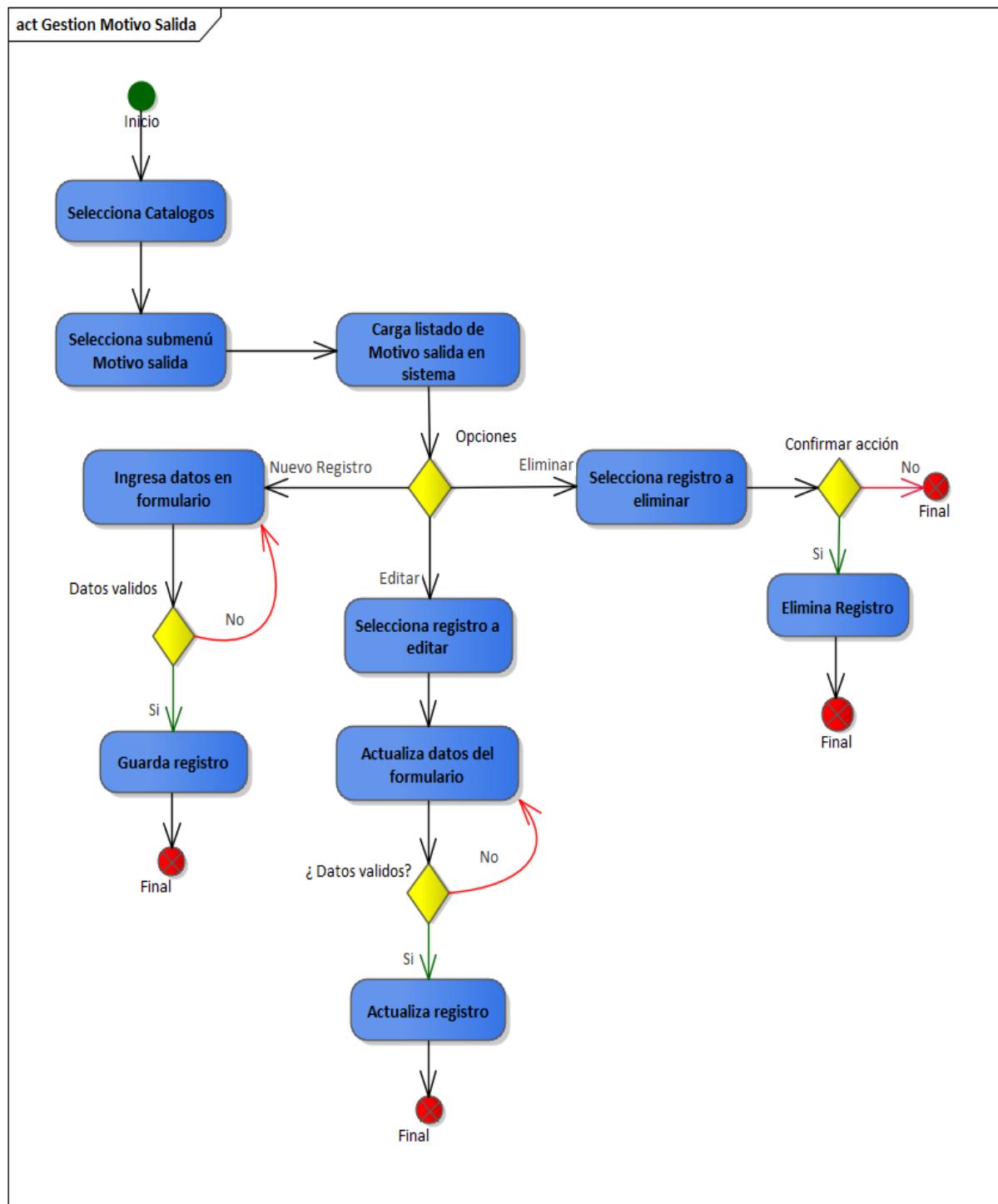


Ilustración 28 Diagrama de Actividad Gestión motivo de salida

Fuente: elaboración propia.

- Gestión carreras

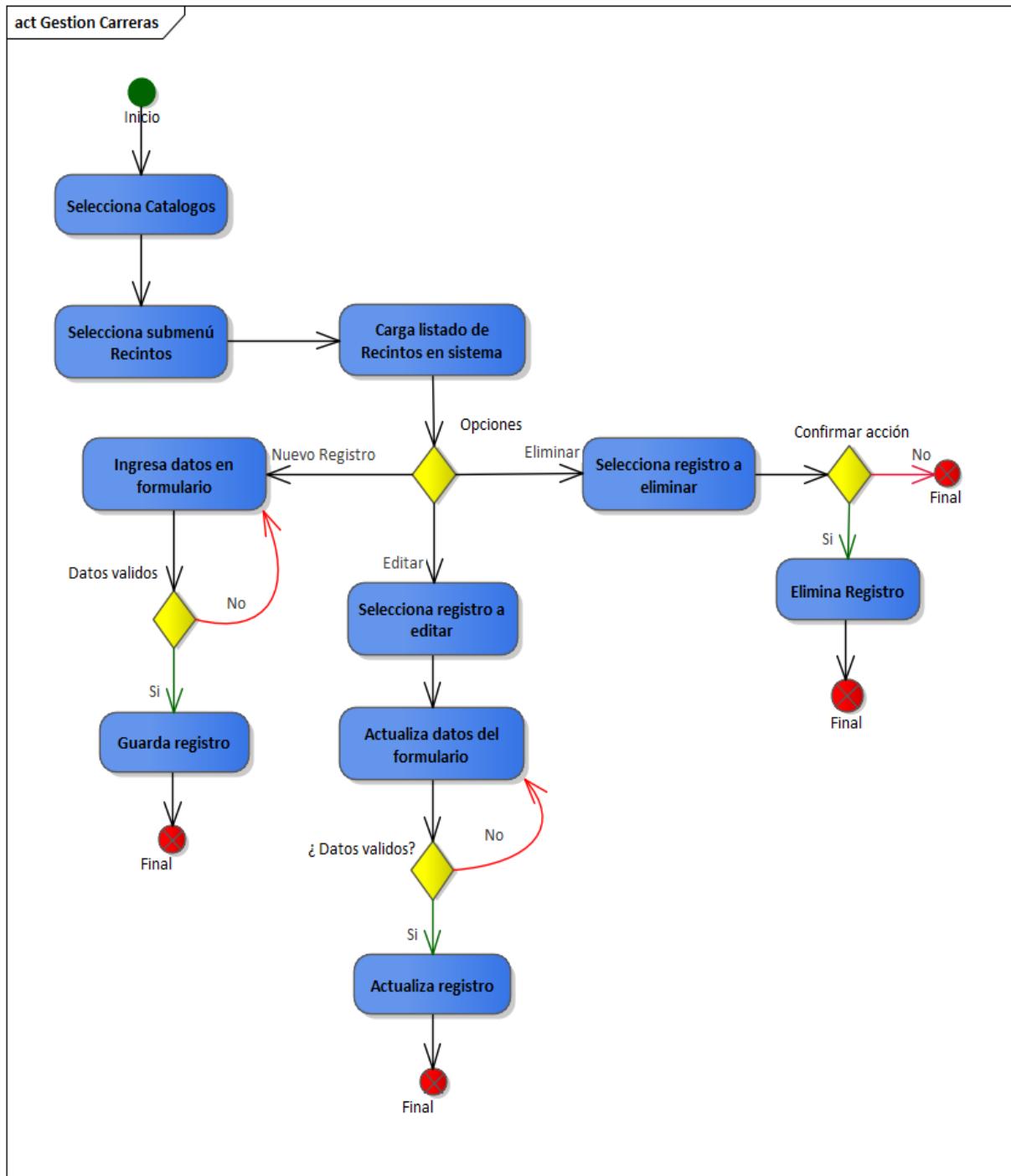


Ilustración 29 Diagrama de Actividad Gestión carreras

Fuente: elaboración propia.

- Gestión Recintos

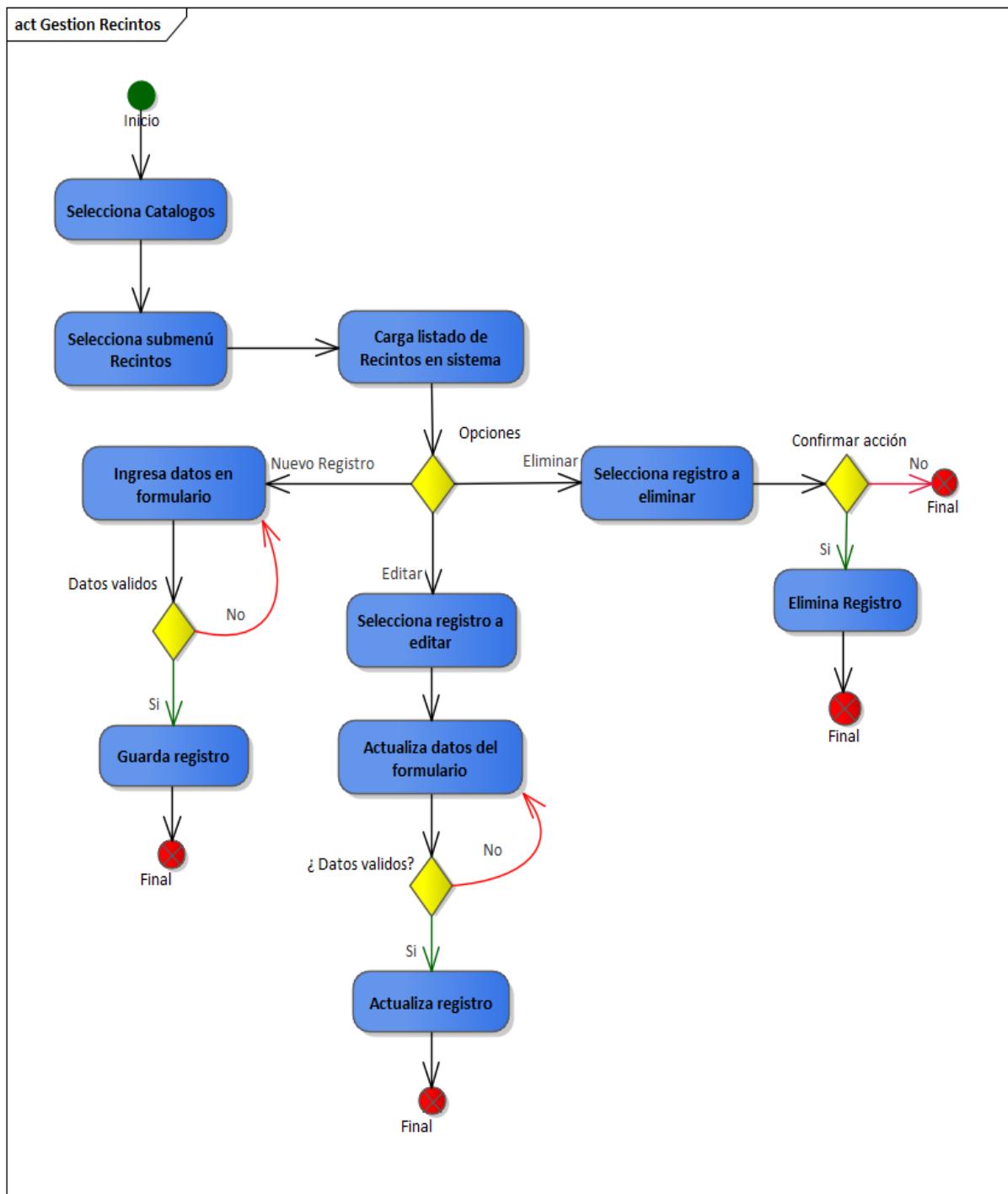


Ilustración 30 Diagrama de Actividad Gestión recintos

Fuente: elaboración propia.

7.1.6. Diagramas de secuencias

Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones. (sparxsystems, 2023).

- **Inicio de sesión**

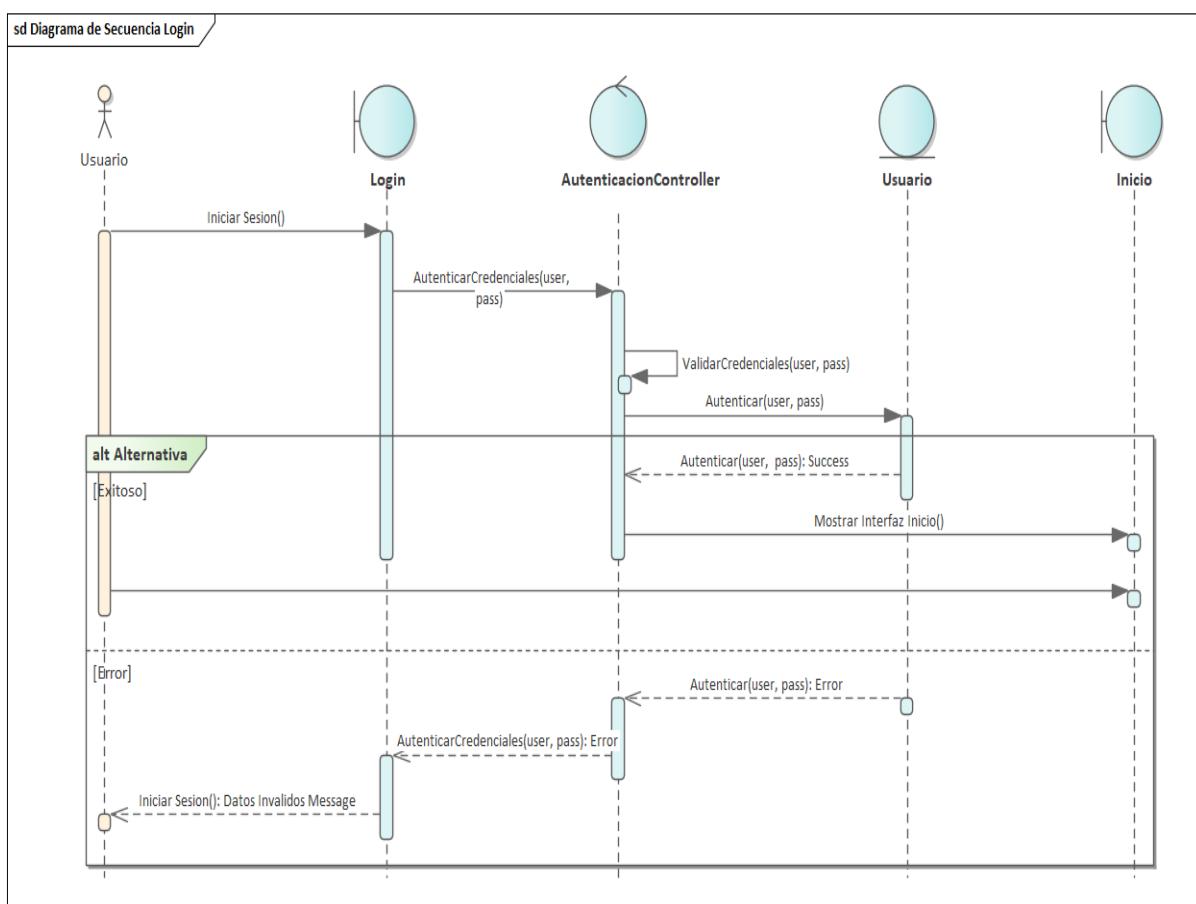


Ilustración 31 Diagrama de Secuencia Inicio de sesión

Fuente: elaboración propia.

- **Ingresar nuevo estudiante**

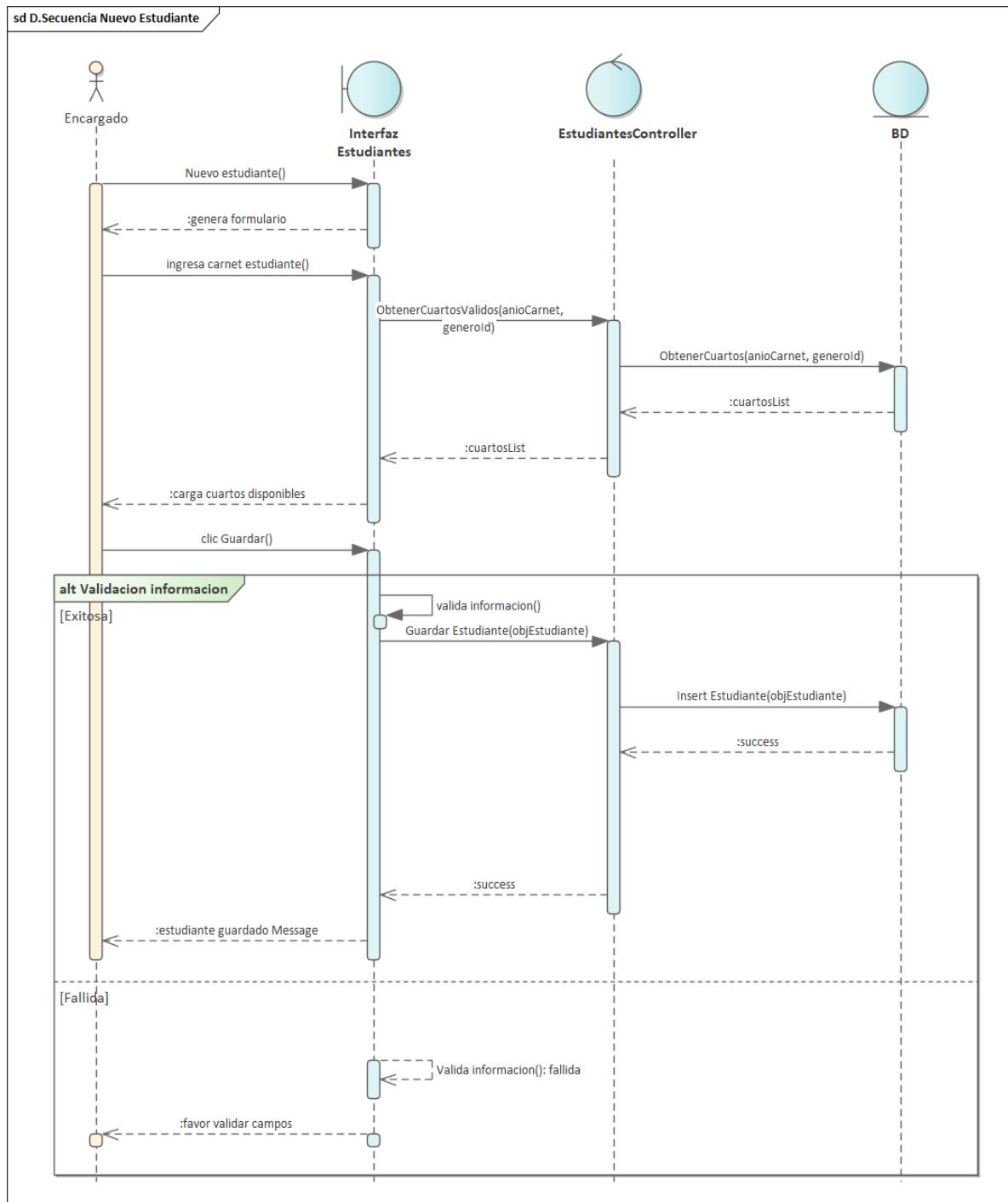


Ilustración 32 Diagrama de Secuencia Ingresar nuevo estudiante

Fuente: elaboración propia.

- **Editar estudiante**

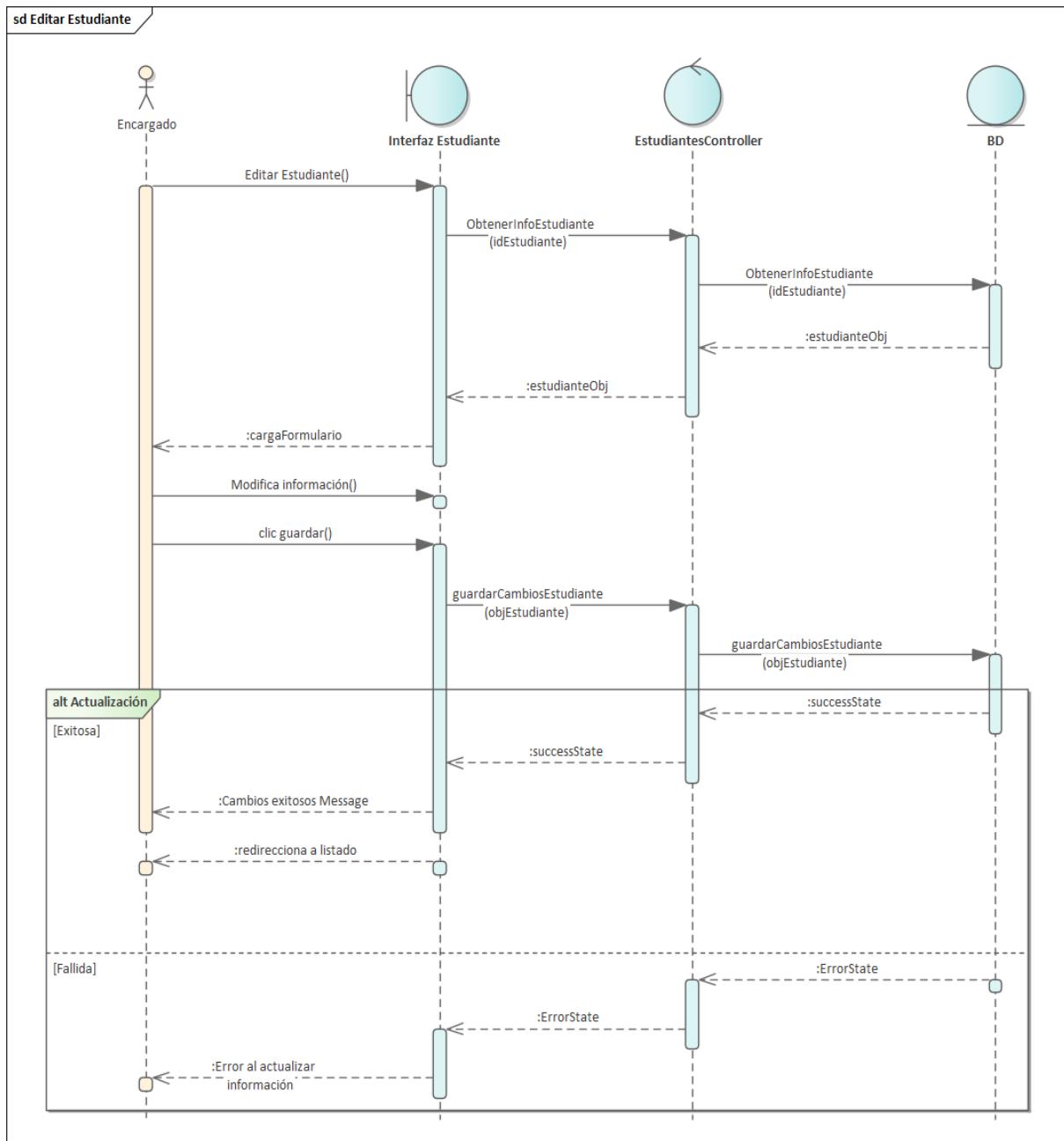


Ilustración 33 Diagrama de Secuencia Editar estudiante

Fuente: elaboración propia.

- **Eliminar estudiante**

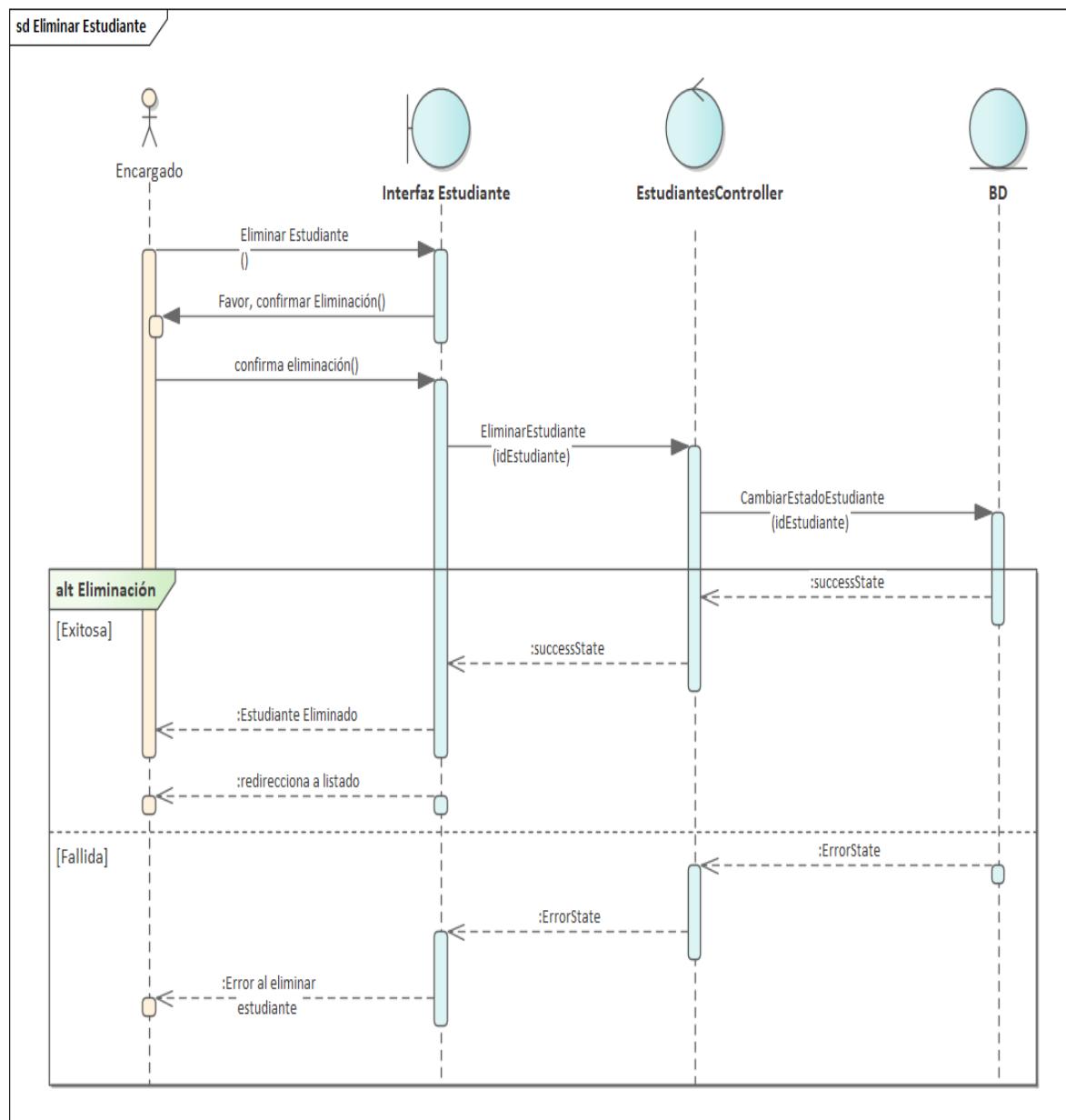


Ilustración 34 Diagrama de Secuencia Eliminar estudiante

Fuente: elaboración propia.

- **Visualizar estudiante**

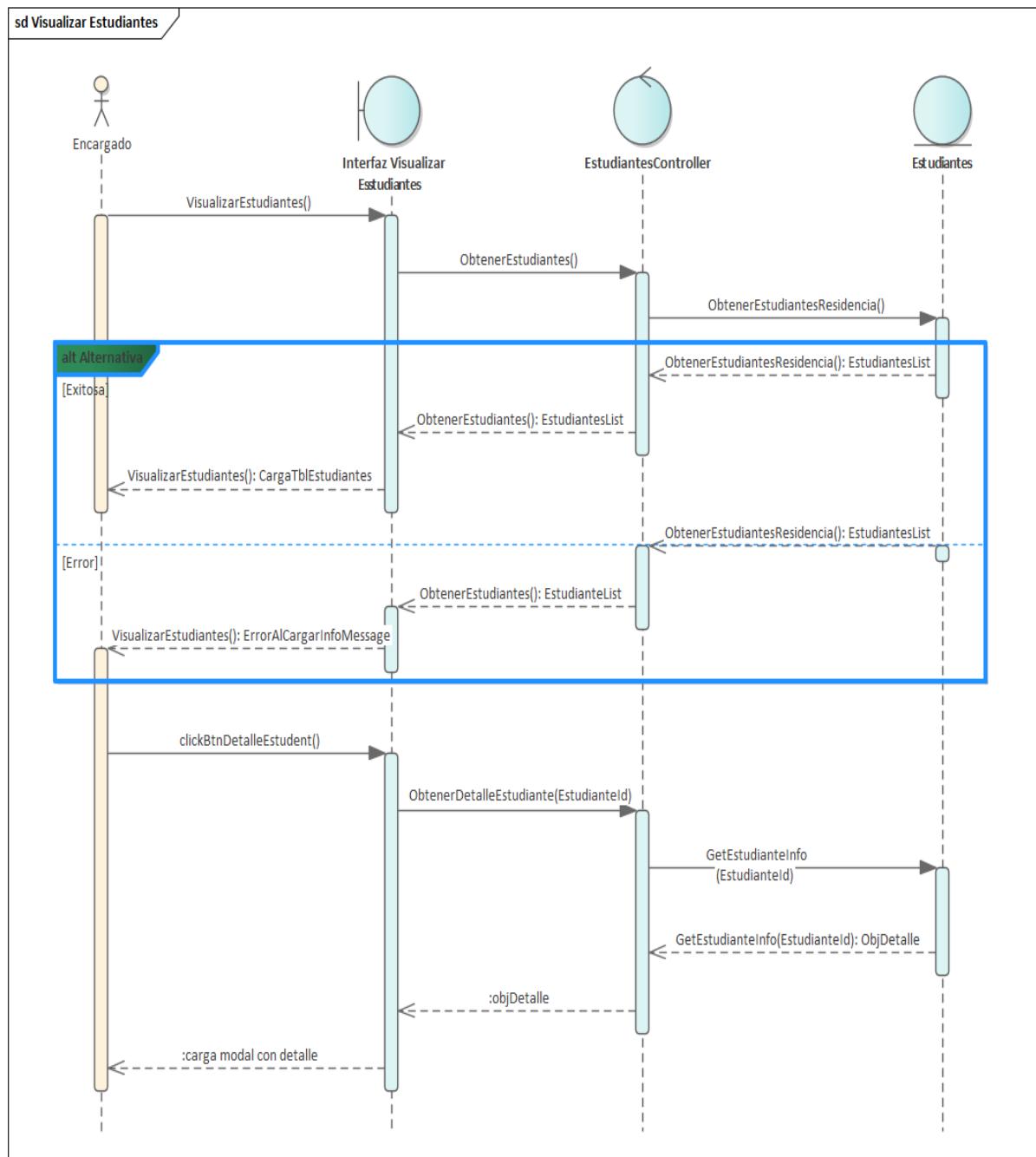


Ilustración 35 Diagrama de Secuencia Visualizar estudiante

Fuente: elaboración propia.

- Cambio de cuarto a estudiante

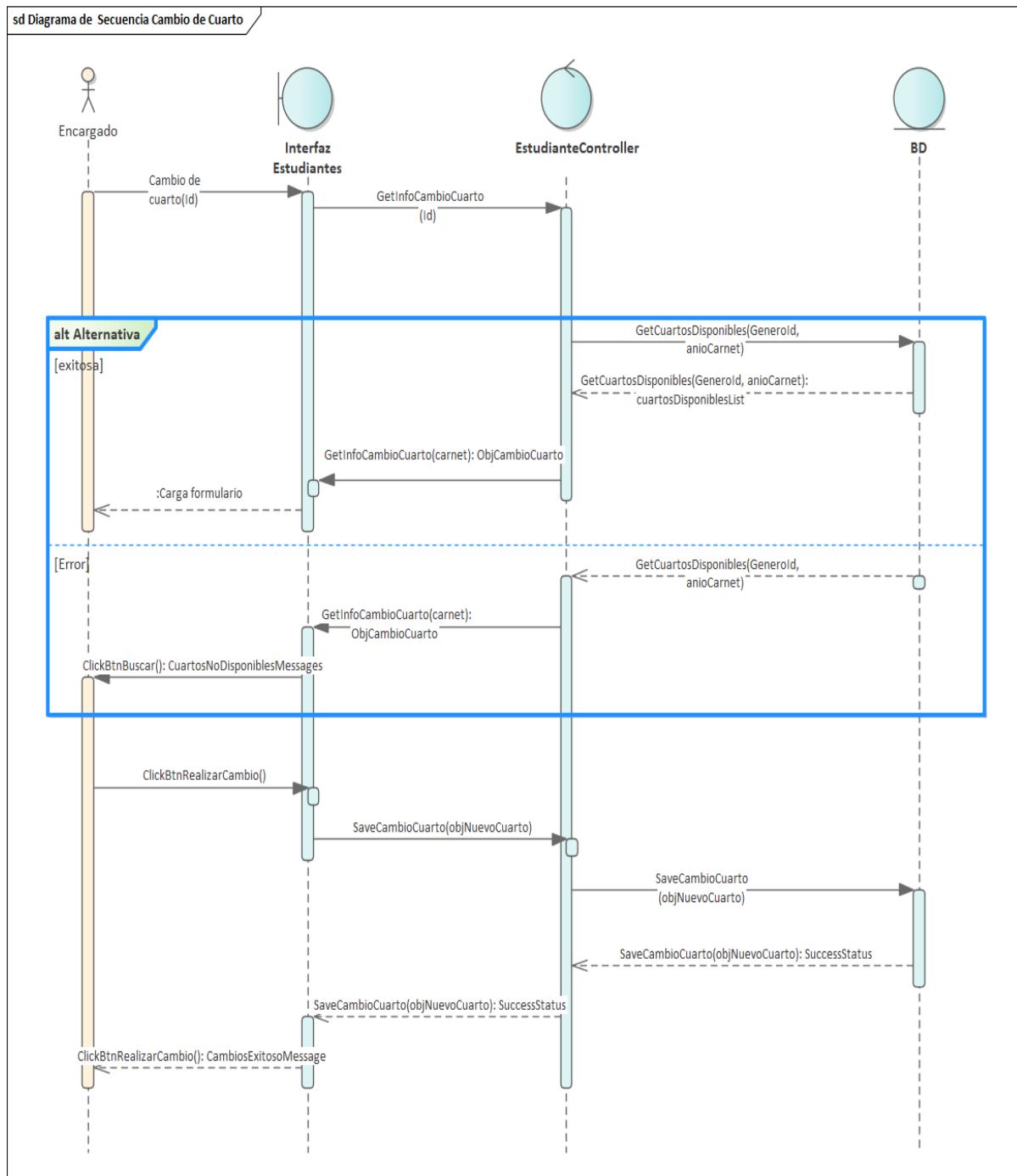


Ilustración 36 Diagrama de Secuencia Cambio de cuarto a estudiante

Fuente: elaboración propia.

- **Reportar Entrada/Salida de estudiante**

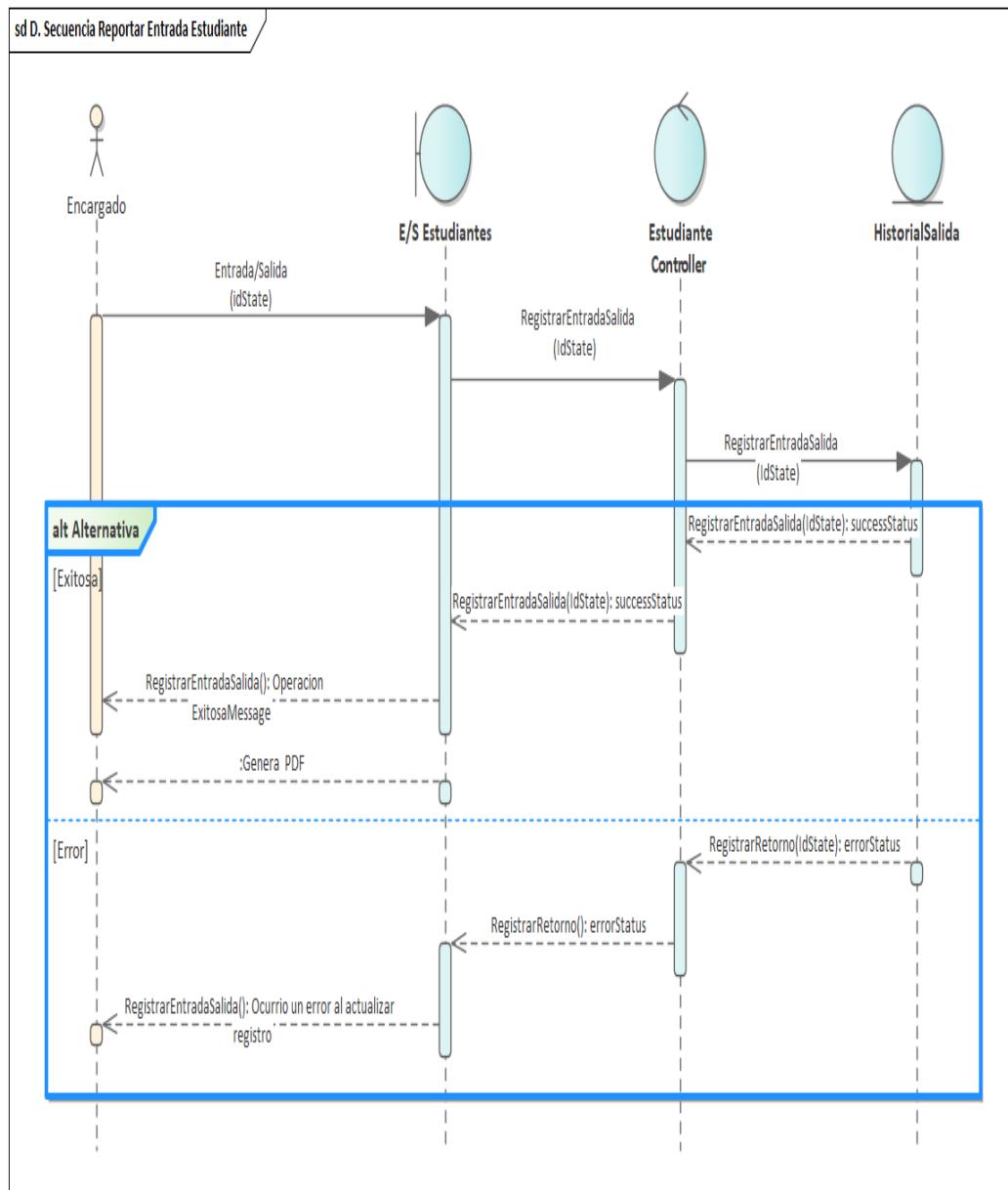


Ilustración 37 Diagrama de Secuencia Reportar entrada/Salida estudiante

Fuente: elaboración propia.

- Nueva asignación de aseo

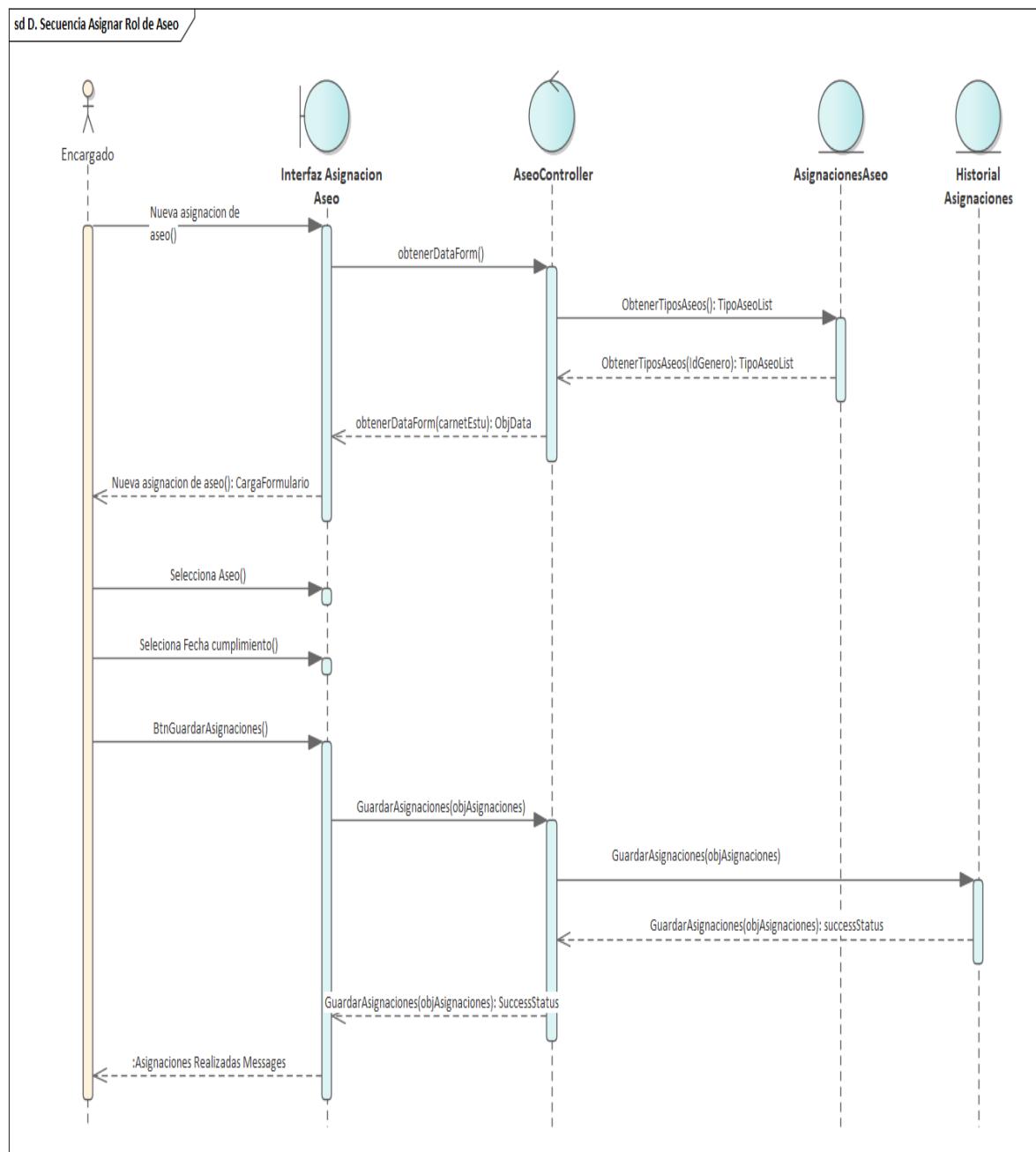


Ilustración 38 Diagrama de Secuencia Nueva asignación de aseo

Fuente: elaboración propia.

- **Editar asignación de aseo**

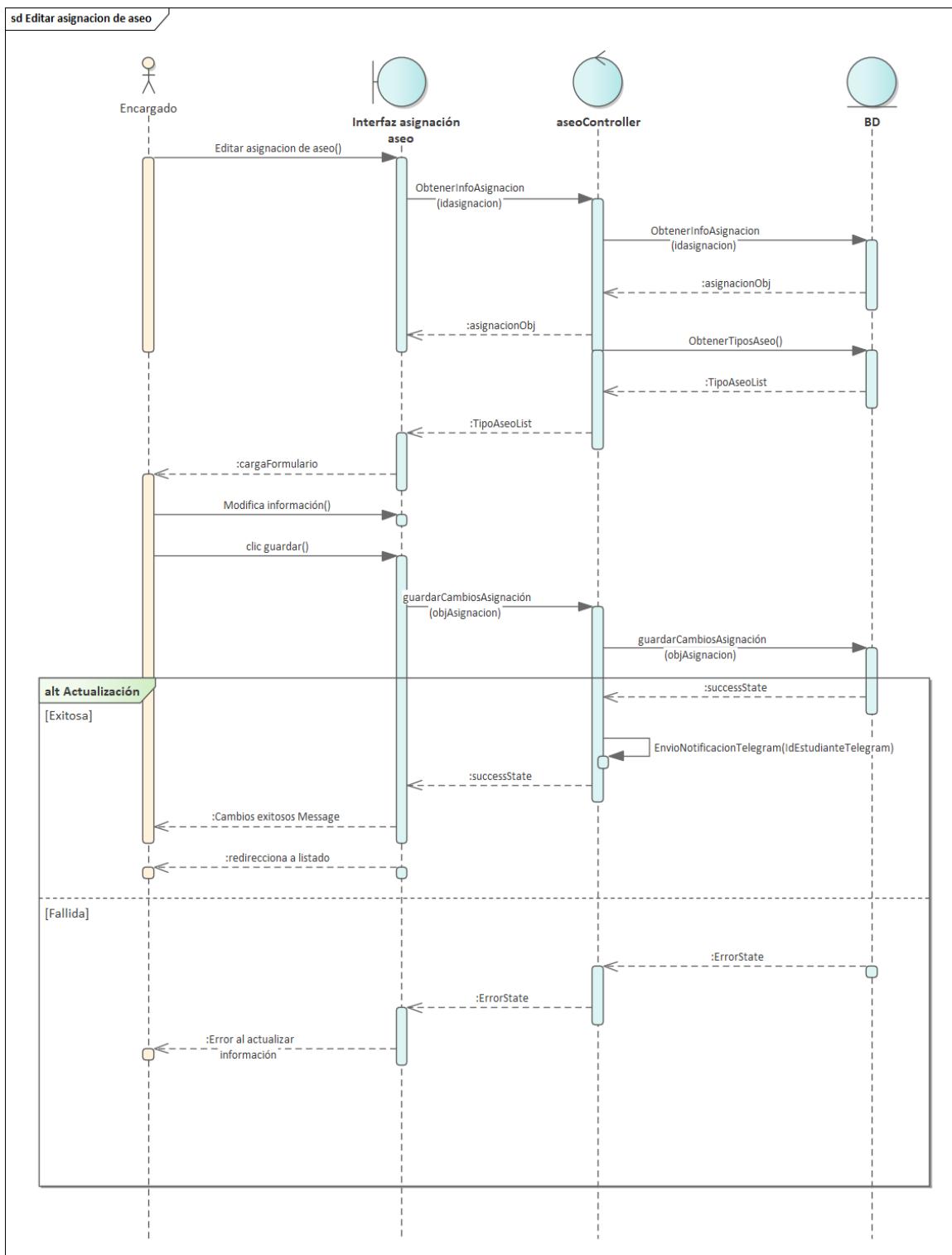


Ilustración 39 Diagrama de Secuencia Editar asignación de aseo

Fuente: elaboración propia.

- **Calendario de asignaciones**

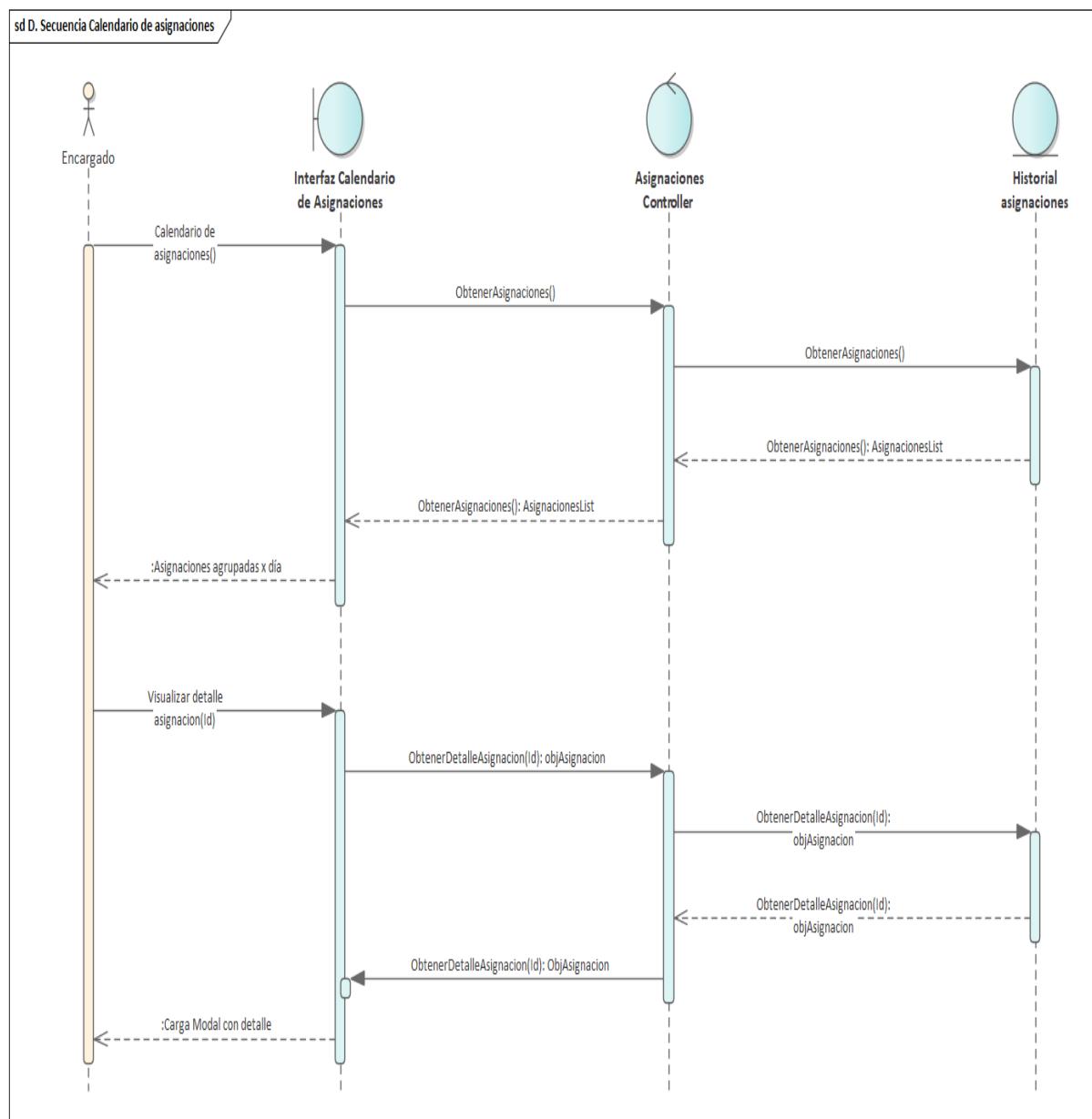


Ilustración 40 Diagrama de Secuencia Calendario de asignaciones

Fuente: elaboración propia.

- **Envío de encuestas de alimentos**

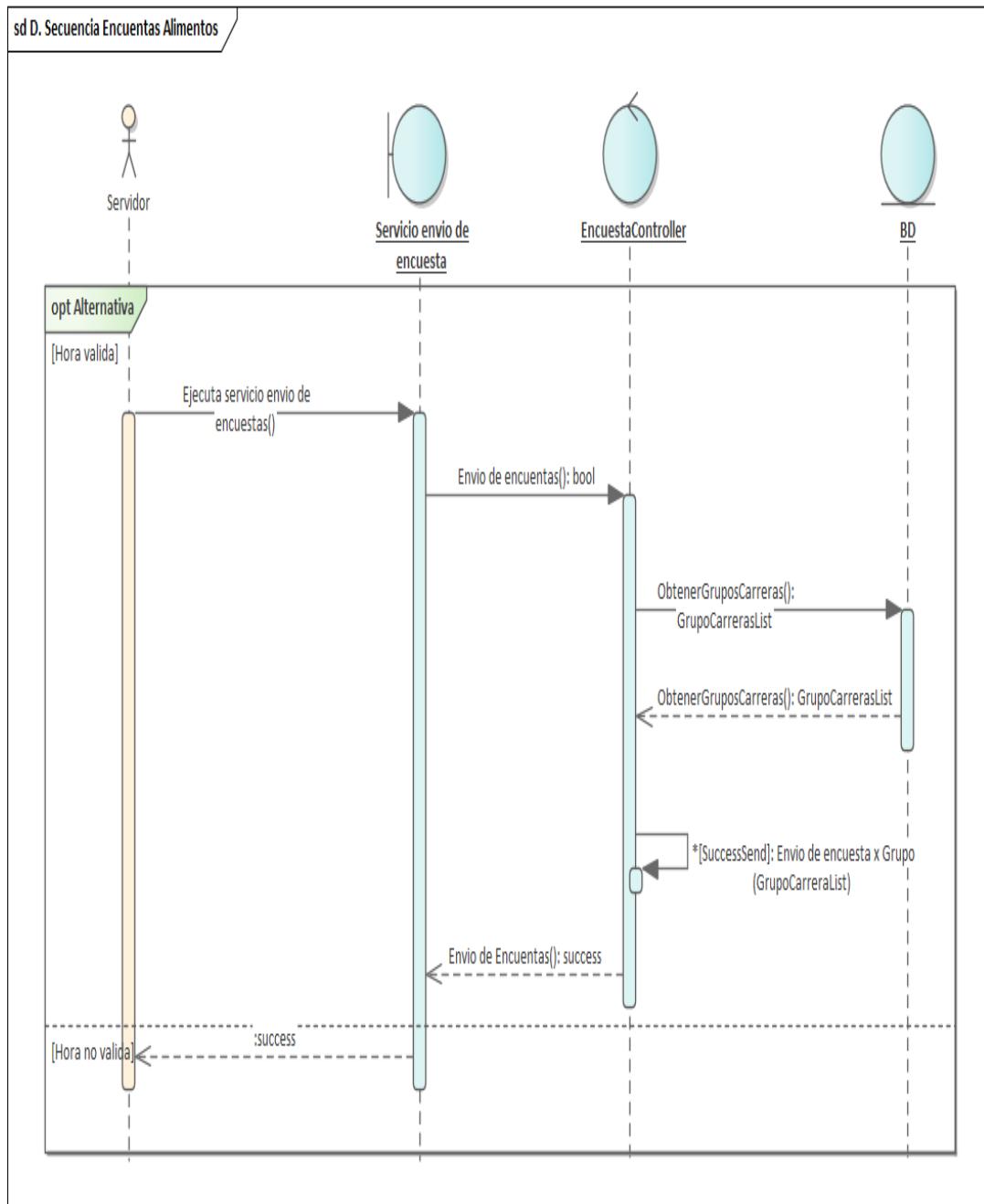


Ilustración 41 Diagrama de Secuencia Envío de encuestas

Fuente: elaboración propia.

- **Panel comedor**

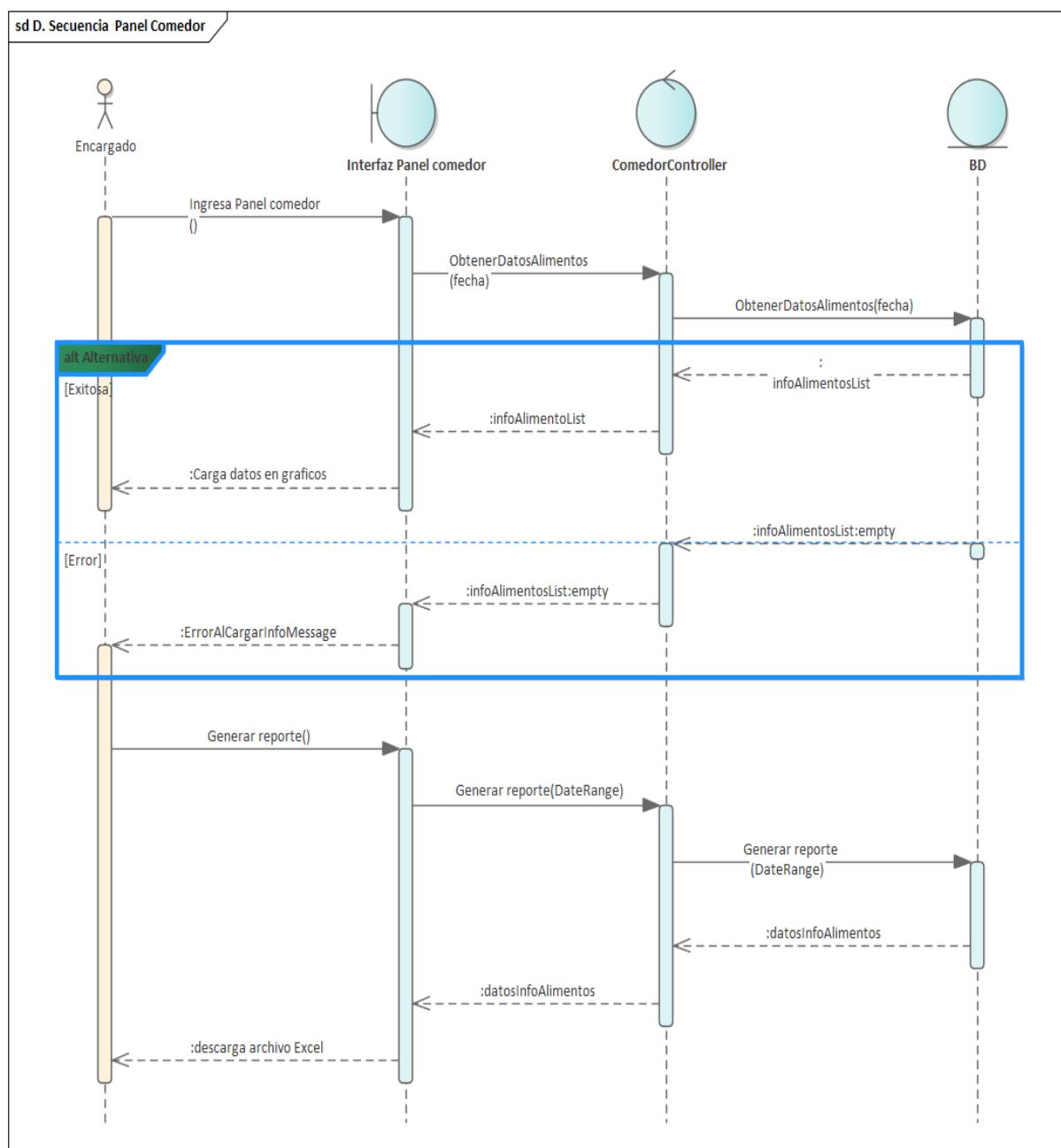


Ilustración 42 Diagrama de Secuencia Panel comededor

Fuente: elaboración propia.

- Nuevo usuario sistema

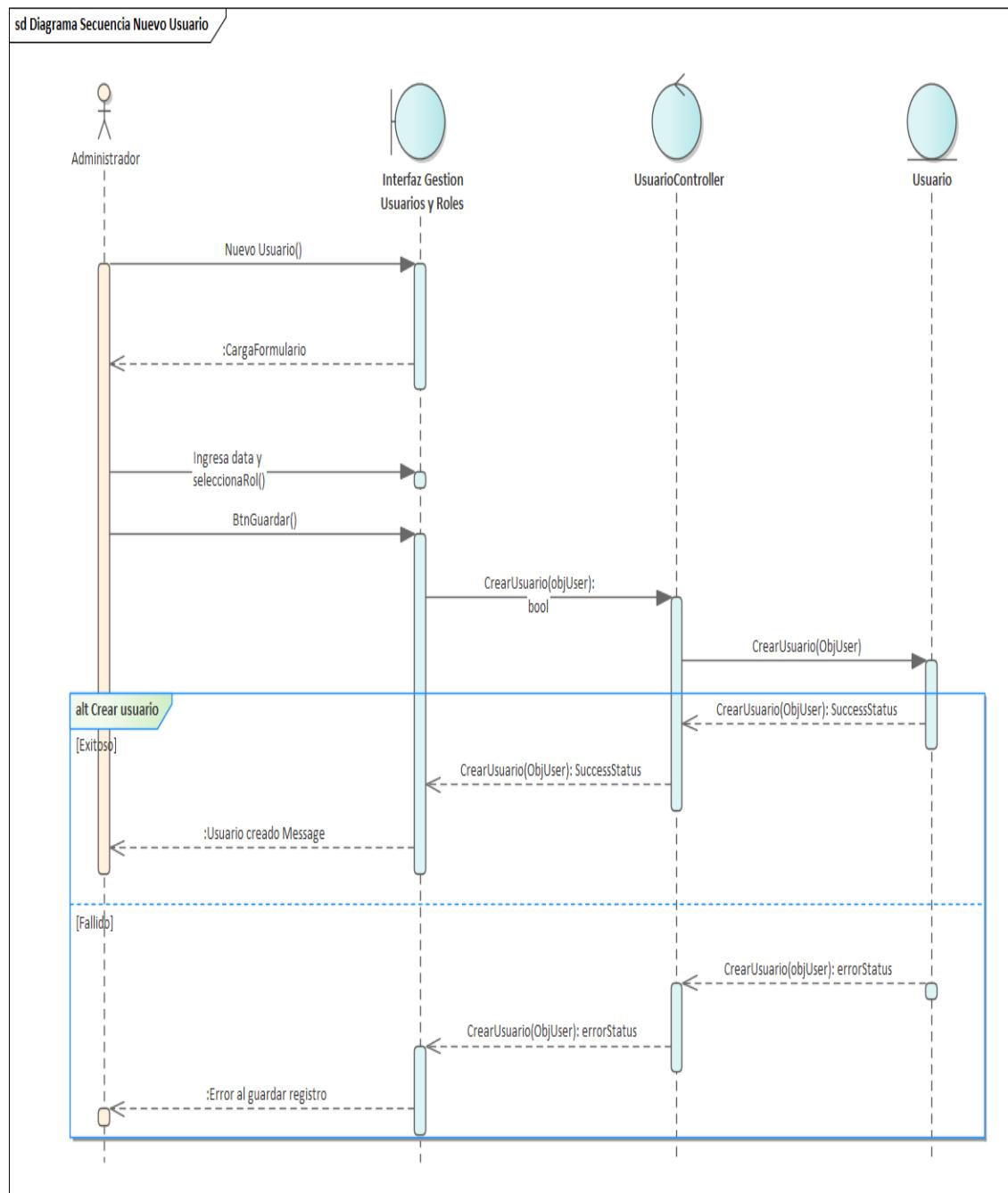


Ilustración 43 Diagrama de Secuencia Nuevo usuario

Fuente: elaboración propia.

7.2. Diseño del sistema

El diseño se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán en el proceso de codificación y la estructura que debe darse a la aplicación web, arquitectura del sistema y del entorno tecnológico.

7.2.1. Modelo de navegación

Este modelo representa mediante un esquema la estructura del hipertexto, es decir, especifica qué objetos serán visitados por el navegador a través de la aplicación y cómo se relacionan.

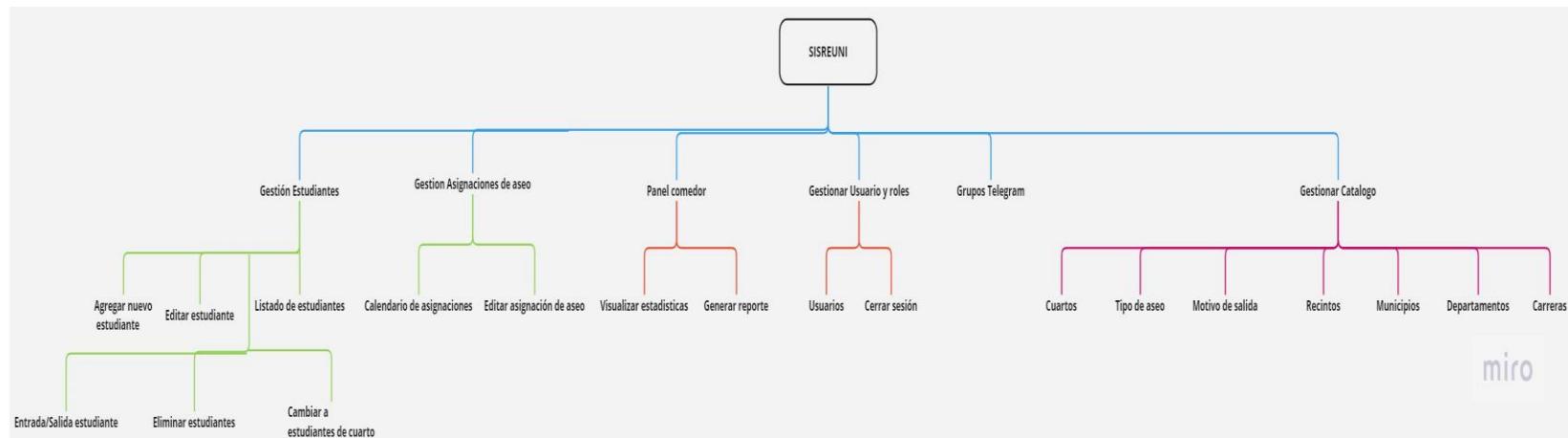


Ilustración 44 Modelo de navegación

Fuente: elaboración propia.

7.2.2. Modelo de presentación

Describe dónde y cómo los objetos de navegación y accesos primitivos serán presentados al usuario, es decir, una representación esquemática de los objetos visibles al usuario.

- **Inicio de sesión**

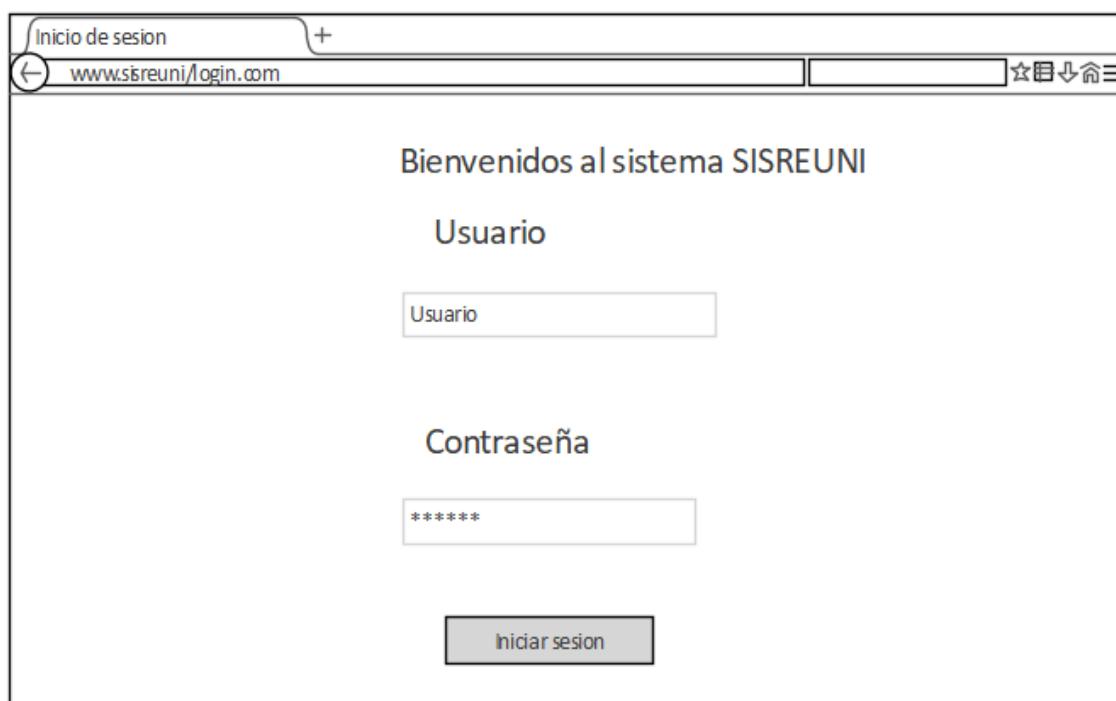


Ilustración 45 Modelo de presentación Inicio de sesión

Fuente: elaboración propia.

- Gestión estudiante

ui User Interface

Gestion estudiantes +
www.ssreuni/estudante/gestion.com

Modal

Nombre Carnet
(from Gestion Estudiante) (from Gestion Estudiante)

Genero
(from Gestion Estudiante)

Cancelar Guardar

+ Nuevo estudiante

Estudiantes en sistema

Nombre estudiante	Carnet	Carrera	Opciones
Gamalel Ordoñez	2016-000E	Computación	Opciones
Jonathan Acuña	2016-111P	Electrica	Opciones
Fernando Herrera	2012-333K	Civil	Opciones

Generación de modal al seleccionar opción de nuevo estudiante

> Editar estudiante
> Eliminar estudiante
> Ver detalle

Ilustración 46 Modelo de presentación Gestión estudiante

Fuente: elaboración propia.

- Gestión asignación de aseo

ui User Interface

Asignar actividades de aseo +
www.ssreuni/asignaciones/agregar.com

Modal

27/7/2023 Cuartos Estudiantes
Tipo de aseo Hora de aseo
Cancelar Guardar

+ Nueva asignación

Actividades asignadas

Nombre estudiante	Actividad de aseo	Fecha de realización	Opciones
Gamalel Ordoñez	Limpiar pasillos	2022-09-09	Opciones
Gamalel Ordoñez	Limpiar baños	2022-09-12	Opciones
Gamalel Ordoñez	Regar jardín	2022-09-29	Opciones

Generación de modal al seleccionar opción de nueva asignación

> Editar asignación
> Eliminar asignación
> Ver detalle

Ilustración 47 Modelo de presentación Gestión asignación de aseo

Fuente: elaboración propia.

- **Calendario de asignaciones**

Calendario de asignaciones

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9 Limpieza baño	10	11	12	13 Limpieza Jard	14
15	16	17	18 Limpieza Pasil	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Ilustración 48 Modelo de presentación Calendario de asignaciones

Fuente: elaboración propia.

- **Panel comedor**

Reporte

Rango de fechas

27/7/2023

Generar reporte

Ilustración 49 Modelo de presentación Panel comedor

Fuente: elaboración propia.

- Gestión usuarios y roles

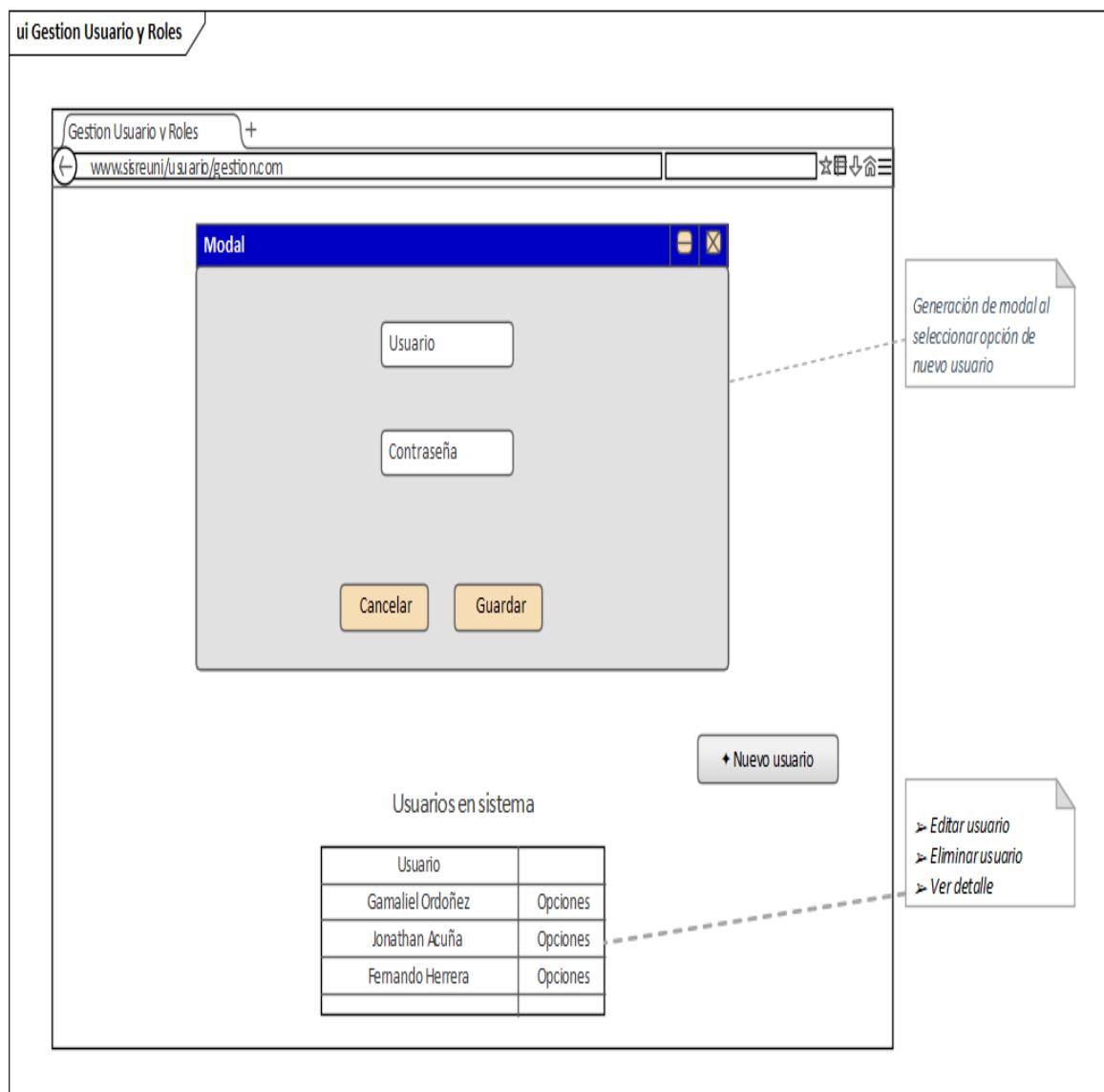


Ilustración 50 Modelo de presentación Gestión Usuarios y roles

Fuente: elaboración propia.

7.2.3. Modelo arquitectónico

El sistema está diseñado bajo la arquitectura MVC, que permite separar en capas la vista, controladores y modelos para brindar una manera más ágil al momento de mantener el código a futuro.

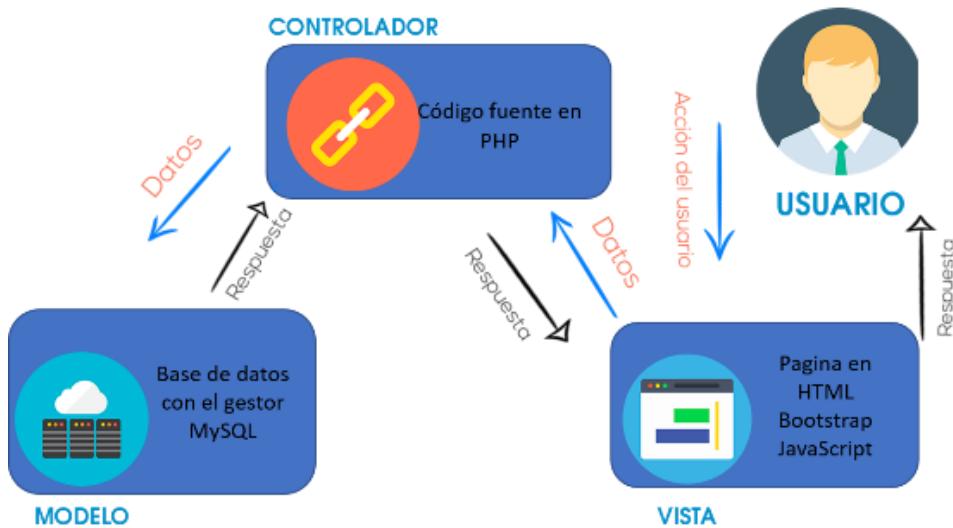


Ilustración 51 Modelo Arquitectónico SISREUNI

Fuente: elaboración propia.

7.2.4. Diagrama de componentes

Los Diagramas de Componentes ilustran las piezas del software, controladores embebidos, etc. que conformarán un sistema eventualmente un componente puede comprender una gran porción de un sistema.

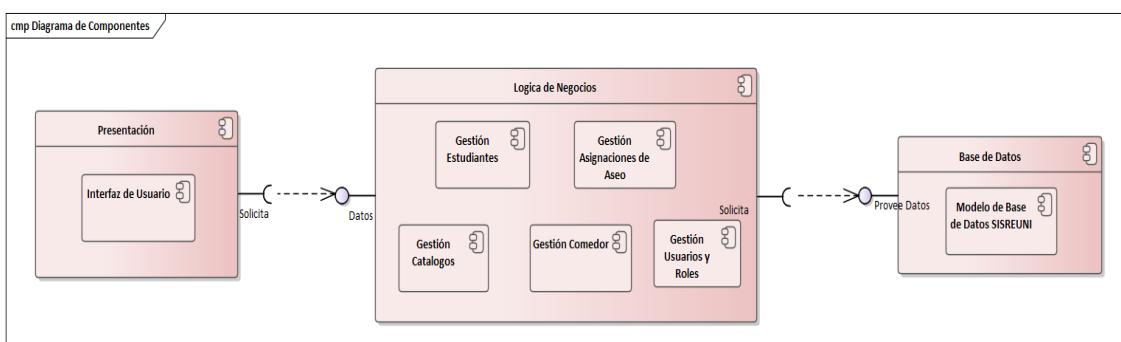


Ilustración 52 Diagrama de componentes SISREUNI

Fuente: elaboración propia.

7.3. Codificación

En esta etapa se desarrolla el código fuente del sistema, traduciendo los modelos a un lenguaje de programación.

7.3.1. Base de datos

Se utilizó el gestor de base de datos mySQLi para realizar la base de datos que utilizará el Servidor.

7.3.2. Tecnologías de desarrollo del código

Para el desarrollo del código fuente se utilizó el IDE Visual Studio Code con el lenguaje de programación PHP en su versión 8, utilizando la arquitectura MVC. Para los estilos e interfaces se utilizó la librería Bootstrap, HTML5 y CSS3, así como código de la librería JQuery de JavaScript, al igual el uso de AJAX para validaciones asíncronas y otras funcionalidades, también se implementó el SDK de la app de mensajería Telegram para el envío de notificación.

7.3.3. Aspectos de seguridad

- **Seguridad para las consultas**

Para la seguridad en las consultas y evitar el ataque de SQL Injection, se utilizó la base de datos mySQLi, que permite realizar consulta preparadas dificultando estos tipos de ataques, además se utiliza la función mysqli_real_escape_string para limpiar la data que proviene de la interfaz.

```
$numero      = $_POST['numero'];
$id_cuarto   = $_POST['id_cuarto'];
$tipo_cuarto  = $_POST['tipo_cuarto'];
$ubicacion   = clean(mysqli_real_escape_string($link,$_POST['ubicacion']));
$ultimo_cuarto = ultimo_cuarto($link);
```

Ilustración 53 Seguridad consultas SQL

Fuente: elaboración propia.

7.4. Pruebas

En esta fase los elementos ya codificados se validaron mediante pruebas funcionales para garantizar que el sistema se comporte correctamente y cumplan con los requisitos solicitados antes de ser entregado al usuario final.

Tabla 41 Pruebas de verificación

Nombre de la prueba	Resultado
Caso de Prueba N° 1: Registrar Estudiante	Satisfactorio
Caso de Prueba N° 2: Editar Estudiante	Satisfactorio
Caso de Prueba N° 3: Eliminar Estudiante	Satisfactorio
Caso de Prueba N° 4: Registrar asignación de aseo	Satisfactorio
Caso de Prueba N° 5: Editar asignación de aseo	Satisfactorio
Caso de Prueba N° 6: Eliminar asignación de aseo	Satisfactorio
Caso de Prueba N° 7: Registrar Tipo de aseo	Satisfactorio
Caso de Prueba N° 8: Editar Tipo de aseo	Satisfactorio
Caso de Prueba N° 9: Eliminar Tipo de aseo	Satisfactorio
Caso de Prueba N° 10: Registrar Cuarto	Satisfactorio
Caso de Prueba N° 11: Editar Cuarto	Satisfactorio
Caso de Prueba N° 12: Eliminar Cuarto	Satisfactorio
Caso de Prueba N° 13: Registrar Motivo de salida	Satisfactorio
Caso de Prueba N° 14: Editar Motivo de salida	Satisfactorio
Caso de Prueba N° 15: Eliminar Motivo de salida	Satisfactorio
Caso de Prueba N° 16: Registrar Recinto	Satisfactorio
Caso de Prueba N° 17: Editar Recinto	Satisfactorio
Caso de Prueba N° 18: Eliminar Recinto	Satisfactorio
Caso de Prueba N° 19: Registrar Carrera	Satisfactorio
Caso de Prueba N° 20: Editar Carrera	Satisfactorio
Caso de Prueba N° 21: Eliminar Carrera	Satisfactorio
Caso de Prueba N° 22: Registrar Usuario	Satisfactorio
Caso de Prueba N° 23: Editar Usuario	Satisfactorio
Caso de Prueba N° 24: Eliminar Usuario	Satisfactorio
Caso de Prueba N° 25: Registrar Grupo de Telegram	Satisfactorio
Caso de Prueba N° 26: Editar Grupo de Telegram	Satisfactorio
Caso de Prueba N° 27: Cambiar estado Grupo de Telegram	Satisfactorio
Caso de Prueba N° 28: Enviar encuesta de almuerzo	Satisfactorio
Caso de Prueba N° 29: Enviar encuesta de cena y desayuno	Satisfactorio
Caso de Prueba N° 30: Cerrar encuesta	Satisfactorio
Caso de Prueba N° 31: Cambiar estudiante de cuarto	Satisfactorio
Caso de Prueba N° 32 Registrar salida de estudiante	Satisfactorio
Caso de Prueba N° 33: Registrar retorno de estudiante	Satisfactorio

Caso de Prueba N° 34: Generar reporte de solicitud de comidas

Satisfactorio

Fuentes: elaboración propia

7.4.1. Detalle de pruebas

En las pruebas de verificación detalladas en la tabla anterior, en el presente documento únicamente se hará énfasis en dos de ellas el caso de prueba 1 y el caso de prueba 4, esto con el fin resumir este apartado extenso.

Tabla 42 Caso de prueba N°.1 Registrar estudiante

Caso de Prueba N° 1: Registrar Estudiante	
Identificador	P_R_Estudiante
Función Probar	Registro de Estudiante
Objetivo	Verificar que los usuarios puedan registrar estudiantes correctamente.
Criterios de Éxito	El sistema presenta mensaje de éxito al guardar o mensaje de validación, la lista en pantalla se actualiza con el estudiante recién registrado.
Criterios de Falla	El sistema no retorna mensaje de validación ni de guardado, tampoco actualiza la pantalla mostrando la nueva información digitada.
Perfil de Usuario	Encargado
Flujo del Caso de Prueba	<ul style="list-style-type: none">○ Se selecciona Gestión estudiante○ Se selecciona la opción Nuevo estudiante○ El sistema muestra el formulario○ Se digita la información a registrar○ Se hace clic al botón guardar
Resultados Obtenidos	Registro Guardado Correctamente en la Base de Datos

Fuente: elaboración propia

Ilustración 54 Prueba Interfaz nuevo estudiante

Fuente: elaboración propia.

Lista de estudiantes						
+ Nuevo estudiante						
Show <select>10</select> entries Search:						
#	No Carnet	Nombres y apellidos	Carrera	Estado	Opciones	
9	2023-0730J	Jennifer Valeria soza López	Ingeniería Química	Presente	Options	
8	2016-0091U	Alejandro Fernando Hsha Hzhs	Ingeniería Electrónica	Presente	Options	

Ilustración 55 Prueba Interfaz listado nuevo registro

Fuente: elaboración propia.

Tabla 43 Detalle caso de prueba N° 1 nueva asignación de aseo

Caso de Prueba N° 4: Registrar Asignación de aseo	
Identificador	P_R_asignacionAseo
Función Probar	Registro de asignación de aseo
Objetivo	Verificar que los usuarios puedan asignar correctamente un aseo a estudiantes correctamente.
Criterios de Éxito	El sistema presenta mensaje de éxito al guardar o mensaje de validación, la lista en pantalla se actualiza con la asignación de aseo recién registrado.
Criterios de Falla	El sistema no retorna mensaje de validación ni de guardado, tampoco actualiza la pantalla mostrando la nueva información digitada.
Perfil de Usuario	Encargado

Flujo del Caso de Prueba	<ul style="list-style-type: none"> ○ Iniciar Sesión dentro del Sistema ○ Se selecciona Gestión asignación de aseo ○ Se selecciona la opción Nueva asignación ○ El sistema muestra el formulario ○ Se digita la información a registrar ○ Se hace clic al botón guardar
Fecha	20/07/23
Resultados Obtenidos	Registro Guardado Correctamente en la Base de Datos

Fuente: elaboración propia

Ilustración 56 Prueba Interfaz nueva asignación

Fuente: elaboración propia.

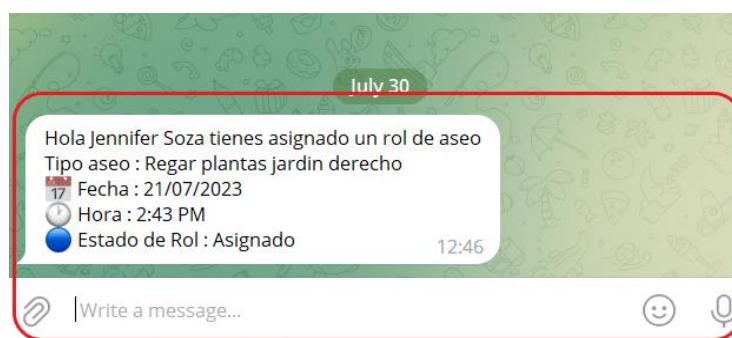


Ilustración 57 Prueba notificación nueva asignación

Fuente: elaboración propia.

7.5. Implementación

En esta etapa una vez finalizado el desarrollo del sistema se entregó a los usuarios finales el sistema, brindando un manual de usuario que se incorporó dentro del sistema para su fácil acceso. Así como una introducción y capacitaciones para el manejo de las funcionalidades del sistema.

Cabe destacar que el sistema se alojó en un Host gratuito como lo es Hosting24 capaz de soportar las tecnologías de PHP y mySQLi, esto permite mantener el sistema en funcionamiento y que los usuarios puedan acceder cuando necesiten a las funcionalidades. A continuación, se muestra un poco la configuración del Host.

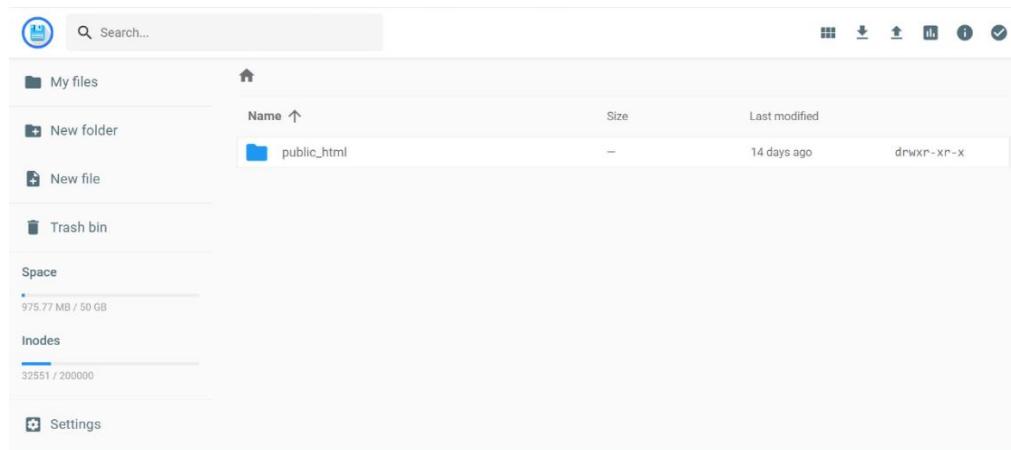


Ilustración 58 Carpeta contenedora de código fuente de SISREUNI

Fuente: elaboración propia.

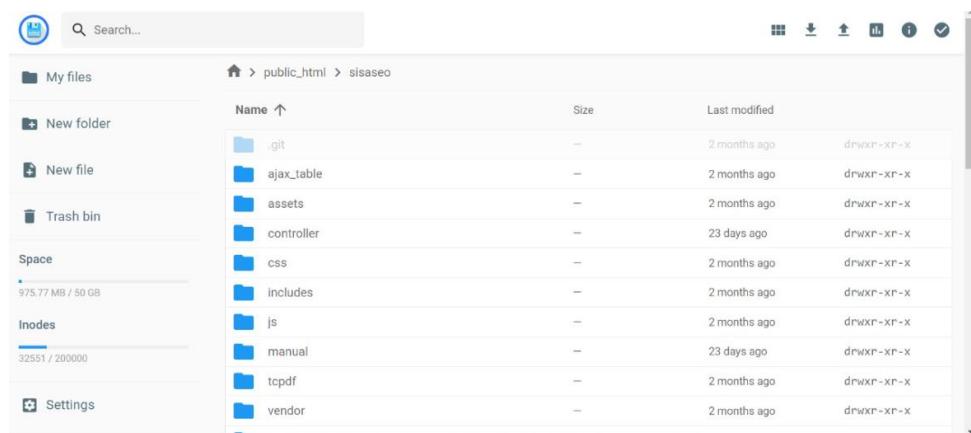


Ilustración 59 Estructura del sistema desplegado en el Hosting

Fuente: elaboración propia.

8. Conclusiones

El presente trabajo monográfico tuvo como objetivo el desarrollo de un Sistema Web que permita a la administración de la residencia estudiantil de la UNI, realizar procesos más eficientes, ahorro de tiempo y capacidad de realización, fácil acceso a la información, concluyendo de manera exitosa con los objetivos y funcionalidades definidas durante el análisis y diseño del proyecto.

Para la exitosa conclusión del desarrollo se utilizó la metodología UWE UML, lo que permitió estructurar el sistema web creado, obteniendo que se cumplió con:

- El análisis y diseño del sistema (objetivo específico 1 y 2) al identificar los requerimientos y reglas de negocio de la residencia estudiantil de la UNI, identificación de usuarios en los procesos y modelado del sistema a través de UML utilizando la herramienta Enterprise Architect.
- La codificación del sistema (objetivo específico 3), se llevó a cabo de manera ágil debido a las herramientas utilizadas como el lenguaje de programación PHP y gestor de base de datos mySQLi, y la implementación de notificaciones a través de SDK de Telegram.
- Se implementó el sistema web en la nube (objetivo específico 4) alojándolo en los servidores de la plataforma Hosting24.
- Al concluir la codificación se realizaron pruebas (objetivo específico 5) para verificar el correcto comportamiento del sistema y aceptación por parte de los usuarios.

9. Recomendaciones

- Incentivar a los estudiantes con beca de residencia a descargar la app de Telegram para que puedan recibir notificaciones instantáneas de asignaciones de aseos y encuestas de solicitud de alimentos.
- Gestionar con la Dirección de Bienestar estudiantil exponer un API privado o un usuario con permisos de lectura para poder poblar al sistema con información de años anteriores.
- Gestionar con la Dirección de Bienestar estudiantil adquirir un Hosting más completo o poder implantar el sistema en sus servidores.
- Informar a cada una de las personas involucradas las características del sistema, las ventajas del uso de este, y comunicar los cambios que se harán en el modelo de trabajo con la adquisición del nuevo sistema.
- Para posteriores incorporaciones de funcionalidades se recomienda implementar un filtro en la interfaz de inicio para permitir visualizar cantidad de estudiantes por año.
- Desarrollar funcionalidad para poder incorporar el uso de un scanner de código de barra para obtener información del estudiante al escanear el carné estudiantil y no ingresar dicha información de manera manual.

10. Anexos

10.1. Carta de solicitud de autorización para realizar sistema

Managua 19 de Enero del 2021.

Ing. Isaac Ríos Medina
Director DBE
UNI
Su despacho

Estimado Ing. Ríos,

Reciba un cordial saludo. A través de la presente me dirijo a usted con el objetivo de solicitar su apoyo para que los siguientes estudiantes:

Br. Jonathan Alejandro Acuña Barrios (Carné 2016-0091U) y
Br. Joaquín Gamaliel Ordoñez Díaz (Carné 2016-0019U), estudiantes de quinto año de la carrera de Ingeniería en Computación, puedan desarrollar la propuesta del tema monográfico titulado **Desarrollo del Sistema WEB para el control y Monitoreo de los Roles de Aseos de los estudiantes Internos de la universidad nacional de ingeniería UNI**.

Para el buen desarrollo de este proyecto se necesitaría que personas conocedoras del proceso de gestión de control de los estudiantes internos UNI le brinden información sobre cómo se realiza el proceso y validen los resultados generados por los jóvenes (como sucede con cualquier proyecto informático).

Este proyecto es realizado con fines académicos, y la información que ustedes provean será manejada siguiendo bases éticas y de acuerdo a los lineamientos institucionales.

De antemano le agradezco el apoyo que pueda brindar a los Br. Acuña y Ordoñez para desarrollar este proyecto que puede tener mucha utilidad para el control Interno de los estudiantes. Quedo a la orden para cualquier consulta al respecto.



Atentamente
MSc. Ing. Luis Eduardo Chávez Mairena
Jefe Dpto. Arquitectura y Sistema de Aplicación
Facultad de electrotecnia y computación
Universidad Nacional de Ingeniería

Cc. Parte Interesada

Ilustración 60 Carta de solicitud sistema

10.2. Carta de aceptación del sistema



UNIVERSIDAD NACIONAL DE INGENIERÍA
RECTORÍA
DIRECCIÓN DE BIENESTAR ESTUDIANTIL



Managua, 22 de marzo del 2021.

MSc. Ing. Luis Eduardo Chávez Mairena
Jefe Dpto. Arquitectura y Sistema de Aplicación
Facultad de Electrotecnia y Computación.
Universidad Nacional de Ingeniería

Estimado MSc. Ing. Chávez.

Reciba cordiales saludos de mi parte esperando que se encuentre gozando de buena salud y éxitos en sus funciones diarias.

Por medio de la presente me dirijo usted para informarle que estamos aceptando la propuesta emitida por usted de los siguientes bachilleres de la carrera de Ingeniería en computación:

Jonathan Alejandro Acuña Barrios (carne 2016-0091U)

Joaquín Gamaliel Ordoñez Díaz (carne 2016-0019U)

Del tema monográfico titulado **DESARROLLO DEL SISTEMA WED PARA EL CONTROL Y MONITOREO DE LOS ROLES DE ASEOS DE LOS ESTUDIANTES INTERNOS DE LA UNIVERSIDAD NACIONAL DE INGENIERÍA.**

- Adjunto copia de la carta emitida por ustedes.

Sin más a que hacer mención me despido de usted deseándole muchos éxitos en sus funciones cotidianas.

Atte.

J P 2101
Ing. Jorge Isaac Ríos Medina
Director DBE



C.c. Archivo.
Lic. Humberto Pérez Responsable de Vivienda Estudiantil
Br. Jonathan Acuña Barrios Estudiante
Br. Joaquin Ordoñez Diaz Estudiante

Recibido 1.221
23.4.21
2:00 pm

25/03/2021
1:44 pm
K

Ilustración 61 Carta de aceptación Dirección de Bienestar Estudiantil

11. Bibliografía

- Aguilera Trujillo, C. A., Miranda Alvarado, G. A., & Varela Mercado, M. O. (2016). *Servicio Web para el control de los procesos de la Jornada Universitaria de Desarrollo Científico (JUDC) basada en la arquitectura Api Rest en el Departamento de Computación*. Managua, Nicaragua.
- Alor-Hernández. (2014). . *Frameworks, Methodologies, and Tools for Developing*. Mexico: IGI Global.
- Alvarez, M. A. (02 de enero de 2014). *desarrolloweb*. Obtenido de <https://desarrolloweb.com/articulos/que-es-mvc.html>
- Alvarez, M. A. (28 de julio de 2020). *Desarrollo Web*. Obtenido de Desarrollo Web: <https://desarrolloweb.com/articulos/que-es-mvc.html>
- AMAZON. (19 de 07 de 2023). *AWS Amazon*. Obtenido de AWS Amazon: <https://aws.amazon.com/es/what-is/sdk/#:~:text=Un%20kit%20de%20desarrollo%20de,o%20lenguaje%20de%20programaci%C3%B3n%20espec%C3%ADficos>.
- Angel, M. A. (2013). Introducción a PHP. En M. A. Angel. IT CAMPUS ACADEMY.
- Arias Cortez, M. F., Lopez Almendarez, M. L., & Navarrete Sequeria, S. A. (2018). *Sistema de Planificación de Locales para el Recinto*. Managua. Obtenido de SISTEMA DE PLANIFICACIÓN DE LOCALES PARA EL RECINTO.
- Arias, A. (2015). Aprende a Programar Ajax y Jquery.
- Cantarero, J. A., Guido, L. D., & Rivera, A. S. (2016). Sistema de información web para la gestión de relación entre Plasencia Cigars y sus clientes. En *Sistema de información web para la gestión de relación entre Plasencia Cigars y sus clientes* (pág. 193). Esteli.
- developers google. (05 de 12 de 2021). *developers.google*. Obtenido de developers.google: <https://developers.google.com/search/docs/advanced/security/https?hl=es>
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y JavaScript*. Barcelona: Marcombo, S.A. .
- Gobierno Digital. (06 de marzo de 2020). *Pruebas de Interfaces y Contenidos*. Obtenido de Pruebas de Interfaces y Contenidos: <https://www.guiadigital.gob.cl/articulo/pruebas-de-interfaces-y-contenidos.html>
- Google. (29 de septiembre de 2016). *sites.google*. Obtenido de sites.google: <https://sites.google.com/site/latecnologiasusformasdeuso/seguridad-informatica>

- Guru, R. (18 de 11 de 2019). *Resource Guru*. Obtenido de Resource Guru:
<https://resourceguruapp.com/>
- Juárez Balmaceda , S. M., & Contrera Mendoza, T. V. (2018). *Sistema Web de Control de Ventas al por Mayor de Productos Agroquímicos para SOLIAGROSA* . Managua.
- kinetic software. (15 de febrero de 2022). *Student Accommodation Management Software*. Obtenido de Student Accommodation Management Software:
<https://kineticsoftware.com/student-accommodation-management-software/>
- Lancker, L. V. (2014). jQuery el framework JavaScript de la web 2.0. ENI.
- Maza, M. A. (2012). JavaScript. Innovacion Y Cualificaciones.
- microsoft. (06 de noviembre de 2021). *microsoft*. Obtenido de microsoft:
<https://docs.microsoft.com/es-es/dotnet/framework/data/adonet/ef/security-considerations>
- Olivas, W. M. (2020). *SISTEMA DE INFORMACIÓN WEB DE GESTIÓN DE TICKETS QUE APOYE LOS PROCESOS DEL ÁREA DE BASE DE DATOS DE LA DIRECCIÓN GENERAL DE INGRESOS*. Managua.
- Otto, M. (05 de 12 de 2017). *Bootstrap*. Obtenido de <https://getbootstrap.com/>
- Redeszone. (19 de 07 de 2023). *Redeszone*. Obtenido de Redeszone:
<https://www.redeszone.net/tutoriales/servidores/cron-crontab-linux-programar-tareas/>
- RoomPact. (15 de febrero de 2022). *RoomPact*. Obtenido de RoomPact:
<https://www.roompact.com/>
- S.A, D. (22 de 12 de 2019). *Dasit*. Obtenido de Dasit : <https://www.dasit.es/que-es-la-seguridad-informatica/?cn-reloaded=1>
- Solano, E. J., Miranda, E. J., & Dávila, P. D. (2020). *DESARROLLO DE UN SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE PROCESOS PRODUCTIVOS DE LA GRANJA AVÍCOLA MA'S GESTIÓN DE PROCESOS PRODUCTIVOS DE LA GRANJA AVÍCOLA MA'S*. Juigalpa.
- sparxsystems. (06 de 07 de 2023). *sparxsystems*. Obtenido de sparxsystems:
http://www.sparxsystems.com.ar/resources/tutorial/uml2_activitydiagram.php
- Zamitz, C. A. (26 de 01 de 2020). *UNAM*. Obtenido de profesores.fi-b.unam.mx/carlos/aydoo/toc.html