

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA E INFORMÁTICA
CONCEPCION - CHILE



**“SISTEMA DE GESTIÓN DE CARGA DOCENTE PARA EL
DEPARTAMENTO DE ELECTRÓNICA E INFORMÁTICA
DE LA UNIVERSIDAD TÉCNICA FEDERICO SANTA
MARÍA, SEDE CONCEPCIÓN”**

MATÍAS ENRIQUE CÓRDOVA CARREÑO

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO EN INFORMÁTICA**

Profesor Guía: Cristian Lara Valenzuela

Noviembre - 2023

DEDICATORIA

Dedico esta memoria a mis padres, pareja quienes siempre estuvieron ahí para apoyarme en todo este proceso académico, en los momentos de incertidumbre y en todos estos años empujándome a ser un buen profesional haciendo lo que me gusta y a nunca rendirse.

AGRADECIMIENTOS

En esta memoria quiero agradecer a todos los docentes que pasaron por mi camino tanto en este camino de educación superior como en mi camino de educación básica, ya que sin ninguno de todos esos docentes esta memoria no podría ser como es ahora. Además de agradecer a todos mis compañeros que me estuvieron en los procesos académicos, ya que sin esas experiencias junto a ellos tampoco sería la persona que desarrollo esta memoria. Y sin duda agradecer a los que me enseñaron el mundo de la informática ya que sin esas acciones quizá esto no sería así.

RESUMEN

Esta memoria desarrolla el análisis y diseño de un sistema informático destinado a cubrir la necesidad de los encargados de la gestión de la carga académica de los docentes al momento de impartir asignaturas en la UTFSM. El sistema informático abarca un proceso realizado de manera manual, además de lograr ser un proceso más amigable e intuitivo para los nuevos usuarios de estos procesos. La implementación de este sistema informático sería una mejora considerable tanto en la experiencia del proceso como en el almacenamiento de los datos generado en este proceso.

Palabras Clave—Carga Académica; Docente; Diseño; Web; Universidad

ABSTRACT

This report develops the analysis and design of a computer system intended to cover the needs of those in charge of managing the academic load of teachers when teaching subjects at UTFSM. The computer system covers a process carried out manually, in addition to being a more friendly and intuitive process for new users of these processes. The implementation of this computer system would be a considerable improvement both in the experience of the process and in the storage of the data generated in this process.

Keywords— Academic Load; Teacher; Design; Web; University.

GLOSARIO

UTFSM: Universidad Técnica Federico Santa María.

DEI: Departamento de Electrónica e Informática.

MVC: Modelo – Vista – Controlador.

INDICE DE CONTENIDOS

RESUMEN	4
ABSTRACT	4
INDICE DE FIGURAS	9
INDICE DE TABLAS	11
INTRODUCCIÓN	13
2.1 Conceptos previos.....	15
2.2 Tecnologías	15
2.2.1 Base de datos	15
2.2.2 Framework	15
2.2.3 Arquitectura	16
CAPÍTULO 3: PROPUESTA DE SOLUCION.....	17
3.1 IDENTIFICACIÓN DE PROCESOS	17
3.2 DIAGRAMA DE CASOS DE USO	18
3.3 CASOS DE USO	19
3.2.1 Añadir docente	19
3.2.2 Añadir carga del docente	20
3.2.3 Quitar carga del docente	22
3.2.4 Modificar carga del docente	23
3.2.5 Calculo Horas límite de cada docente	25
3.2.6 Generación informe del docente	26
3.2.7 Generación reporte semestral	27
3.2.8 Edición de información del docente	29
3.2.9 Añadir Asignatura	30
3.2.10 Edición de la asignatura	32
3.2.11 Iniciar sesión	33

3.2.12	Registrar usuarios al sistema	35
3.2.13	Editar Permisos	37
3.2.14	Iniciar periodo	38
3.2.15	Crear paralelo	39
3.3	ARQUITECTURA DEL SISTEMA	40
3.4	DISEÑO DE DATOS.....	41
3.4.1	Modelo de datos	41
3.4.1.1	Tabla departamento	42
3.4.1.2	Tabla paralelo	43
3.4.1.3	Tabla asignatura	44
3.4.1.4	Tabla modalidad.....	45
3.4.1.5	Tabla carrera	46
3.4.1.6	Tabla asig_periodo_par	47
3.4.1.7	Tabla periodo	48
3.4.1.8	Tabla horario.....	49
3.4.1.9	Tabla bloque	50
3.4.1.10	Tabla dia	51
3.4.1.11	Tabla profe_tipo.....	52
3.4.1.12	Tabla sala	53
3.4.1.13	Tabla profesor	54
3.4.1.14	Tabla usuario.....	55
3.4.1.15	Tabla usuario_carrera	56
3.5	INTERFAZ DEL SISTEMA	57
3.5.1	Inicio de sesión.....	57
3.5.2	Menú.....	58
3.5.3	Docente	58
3.5.4	Asignaturas	63

3.5.5	Generación reporte	67
3.5.6	Sistema	69
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN		73
4.1	Validaciones.....	73
CAPÍTULO 5: CONCLUSIONES		76
	Referencias bibliográficas	77

INDICE DE FIGURAS

Figura 1: Diagrama MVC.....	16
Figura 2: Diagrama de casos de uso.	18
Figura 3: Diagrama de secuencia Caso de uso 1.	19
Figura 4: Diagrama de secuencia Caso de uso 2.	21
Figura 5: Diagrama de secuencia Caso de uso 3.	22
Figura 6: Diagrama de secuencia Caso de uso 4.	24
Figura 7: Diagrama de secuencia Caso de uso 5.	25
Figura 8: Diagrama de secuencia Caso de uso 6.	26
Figura 9: Diagrama de secuencia Caso de uso 7.	28
Figura 10: Diagrama de secuencia Caso de uso 8.	29
Figura 11: Diagrama de secuencia Caso de uso 9.	31
Figura 12: Diagrama de secuencia Caso de uso 10.	32
Figura 13: Diagrama de secuencia Caso de uso 11.	34
Figura 14: Diagrama de secuencia Caso de uso 12.	36
Figura 15: Diagrama de secuencia Caso de uso 13.	37
Figura 16: Diagrama de secuencia Caso de uso 14.	38
Figura 17: Diagrama de secuencia Caso de uso 15.	39
Figura 18: Arquitectura Sistema.....	40
Figura 19: Modelo de datos.....	41
Figura 20: Inicio sesión mockup.....	57
Figura 21: Menú mockup.....	58
Figura 22: Docentes mockup	58
Figura 23: Docentes nuevo docente mockup.....	59
Figura 24: Docentes editar mockup	60
Figura 25: Docentes carga mockup	61
Figura 26: Docentes añadir carga mockup	61

Figura 27: Docentes informe generado mockup	62
Figura 28: Asignaturas mockup.....	63
Figura 29: Asignaturas añadir mockup	64
Figura 30: Asignaturas editar mockup	64
Figura 31: Asignaturas paralelos mockup.....	65
Figura 32: Asignaturas paralelo añadir mockup.....	66
Figura 33: Reportes mockup	67
Figura 34: Reportes generar modal mockup.....	67
Figura 35: Reportes generado mockup.....	68
Figura 36: Sistema mockup.....	69
Figura 37: Sistema usuarios mockup	69
Figura 38: Sistema usuarios nuevo mockup	70
Figura 39: Sistema usuarios permisos mockup	70
Figura 40: Periodo alerta nuevo	71
Figura 41: Sistema periodo iniciar	72
Figura 42: Imagen petición tabla asignatura	74
Figura 43: Imagen petición tabla asignatura 2	75

INDICE DE TABLAS

Tabla 1: Caso de uso extendido 1.....	19
Tabla 2: Caso de uso extendido 2.	20
Tabla 3: Caso de uso extendido 3.....	22
Tabla 4: Caso de uso extendido 4.....	23
Tabla 5: Caso de uso extendido 5.....	25
Tabla 6: Caso de uso extendido 6.....	26
Tabla 7: Caso de uso extendido 7.....	27
Tabla 8: Caso de uso extendido 8.....	29
Tabla 9: Caso de uso extendido 9.....	30
Tabla 10: Caso de uso extendido 10.	32
Tabla 11: Caso de uso extendido 11.	33
Tabla 12: Caso de uso extendido 12.	35
Tabla 13: Caso de uso extendido 13.	37
Tabla 14: Caso de uso extendido 14.	38
Tabla 15: Caso de uso extendido 15.	39
Tabla 16: Tabla departamento.	42
Tabla 17: Tabla departamento ejemplo.	42
Tabla 18: Tabla paralelo.....	43
Tabla 19: Tabla paralelo ejemplo.	43
Tabla 20: Tabla asignatura.....	44
Tabla 21: Tabla asignatura ejemplo.	44
Tabla 22: Tabla modalidad.....	45
Tabla 23: Tabla modalidad ejemplo.....	45
Tabla 24: Tabla carrera.....	46
Tabla 25: Tabla carrera ejemplo.....	46
Tabla 26: Tabla asig_periodo_par.....	47

Tabla 27: Tabla asig_periodo_par ejemplo.....	47
Tabla 28: Tabla periodo	48
Tabla 29: Tabla periodo ejemplo.....	48
Tabla 30: Tabla horario	49
Tabla 31: Tabla horario ejemplo.....	50
Tabla 32: Tabla bloque.....	50
Tabla 33: Tabla bloque ejemplo	50
Tabla 34: Tabla día.....	51
Tabla 35: Tabla día ejemplo	51
Tabla 36: Tabla profe_tipo	52
Tabla 37: Tabla profe_tipo ejemplo.....	52
Tabla 38: Tabla sala.....	53
Tabla 39: Tabla sala ejemplo	53
Tabla 40: Tabla profesor	54
Tabla 41: Tabla profesor ejemplo.....	54
Tabla 42: Tabla usuario.....	55
Tabla 43: Tabla usuario ejemplo	55
Tabla 44: Tabla usuario_carrera	56
Tabla 45: Tabla usuario_carrera ejemplo	56

INTRODUCCIÓN

El presente trabajo de título se enfoca en el análisis y diseño de un sistema computacional que esta destinado a el área ejecutiva de un departamento universitario, específicamente al departamento de electrónica e informática. Este sistema tiene el fin de solucionar un problema de transformación digital existente en un proceso ocurrido en el área ejecutiva del departamento. La idea de crear un sistema que ayude a la gestión de la carga de los docentes nace tras la necesidad de optimizar el tiempo de aprendizaje y la realización del proceso, el cual es realizado de manera manual rellenando tablas en una hoja de calculo universal la cual es enviada a cada usuario que realice el proceso.

El sistema de gestión de la carga docente cubrirá lo que actualmente se realiza de manera manual a través del llenado de tablas en hojas de cálculo. Además, el sistema al ser un sistema computacional se le dará permisos a distintos usuarios, los cuales podrán ayudar en el proceso de gestión de la carga docente así evitando el envío de la hoja de cálculo mediante correo electrónico y mantener los datos en una base de datos local y no en un solo archivo así evitando riesgos de perdida.

El presente análisis busca hacer un sistema computacional de calidad el cual logre resolver el problema, tomando en cuenta los requerimientos solicitados por los usuarios de este proceso.

El diseño de la solución nos dará a conocer a detalle la arquitectura del sistema, la arquitectura de los datos que almacenara el sistema, lo que permitiría crear la base de datos que corresponda, además del funcionamiento de los componentes del sistema para conocer al detalle como estos interactuaran. Además, el diseño describirá como se verán las vistas con las cuales los usuarios interactúen las cuales deben ser intuitivas y que lleve poco tiempo aprender el proceso de gestión de la carga docente.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

Actualmente en la UTFSM sede Concepcion al momento de realizar la gestión de la carga académica de cada docente este proceso se realiza a través de llenar tablas en hojas de cálculo, por lo que si el usuario encargado desea compartir la información este debe enviar estos archivos por correo electrónico existiendo algún riesgo de pérdida o de conflicto de versiones con futuras modificaciones de este.

Especificando en el proceso, este se realiza de manera digital, llenando una tabla en una hoja de cálculo que son creadas por cada semestre, esta información queda respaldada solo en estos archivos.

Este proceso es realizado por cada departamento en la universidad, ya que cada departamento está encargado de la carga académica realizada por cada profesor en sus asignaturas. Cada departamento tiene una persona a cargo de realizar esta tarea y muchas veces se solicita ayuda a distintas personas si esto es necesario por lo que es necesario que ambas personas tengan el mismo archivo en el que se están almacenando los dato. Lo anterior hace que este proceso sea engorroso e ineficaz, ya que muchas podrían existir confusiones de versiones del archivo y hacer el proceso aún más engorroso.

Durante el primer semestre del año 2023 un alumno de técnico en informática de la UTFSM ayudo a la mejora de este proceso, desarrollando macros y automatizando un poco el proceso de la gestión de la carga de cada docente, pero aún el proceso se continúa haciendo en hojas de cálculo.

Como objetivos para esta solución son:

Objetivo general

- Diseñar y analizar una plataforma para la gestión de carga académica de cada docente de la universidad.

Objetivos específicos

- Diseñar una plataforma amigable e intuitiva.
- Optimizar el tiempo de aprendizaje para la gestión de carga docente.
- Evitar los conflictos de versiones en el informe de gestión de carga docente.

Esta solución puede ser muy útil para todos los departamentos de la UTFSM. Como primera instancia esta puede ser desplegada por cada departamento, pero su alcance puede ser a nivel de universidad y no solo de departamento.

CAPÍTULO 2: MARCO CONCEPTUAL

2.1 Conceptos previos

Antes de entrar en detalles de las tecnologías que se utilizaron en esta memoria se debe tener mas conocimiento sobre el funcionamiento del proceso estudiado.

El proceso de la gestión de la carga académica es un proceso realizado por cada departamento, este proceso se realiza por cada docente que realice actividades dentro del departamento (docentes de planta, docentes externos al departamento, docentes part-time). Cada docente tiene un máximo de horas que puede realizar dependiendo de su tipo de contrato (planta, part-time), además que cada docente puede realizar clases fuera de su departamento asignado como, por ejemplo, un profesor que imparta la asignatura de programación en una carrera que no sea parte del DEI (Departamento de electrónica e informática). En un caso como el anterior el departamento al cual corresponde la carrera es el que debe hacer esa gestión de la carga académica teniendo en cuenta que el docente puede tener asignadas asignaturas en otros departamentos por lo que no podría exceder de un límite de horas. Este proceso lo hace un encargado de cada departamento el cual muchas veces puede necesitar ayuda.

Luego de realizar toda la gestión, completando la tabla de la hoja de cálculo con la cual se trabaja actualmente esta es enviada para gestionar los pagos que se deben efectuar a cada docente dependiendo de las horas correspondientes trabajadas.

2.2 Tecnologías

Para las tecnologías usadas en esta memoria se consideró la experiencia con ellas anteriormente, buenas prácticas de desarrollo, código libre y escalable. A continuación, se detallarán las tecnologías utilizadas.

2.2.1 Base de datos

En el área de los datos, es decir, el motor de base de datos que se utilizara para el sistema, se selecciono “**MySQL**”, ya que es un motor de base de datos confiable por su historial se usos exitosos en proyectos anteriores, tener una gran documentación, tener una fácil integración con el framework seleccionado, su nivel de escalabilidad y ser de código abierto.

Tal como dice la corporación “Digital Ocean”, “**MySQL**” es un sistema de gestión de bases de datos relacional que al tener un gran nivel de escalabilidad este podrá ser utilizado al momento de implementar el sistema a un nivel mayor como puede ser a nivel de todas las sedes de la UTFSM. (Digital Ocean, 2023)

2.2.2 Framework

Al momento de seleccionar el framework, la opción elegida fue el framework “**Django**”, ya que es un framework de código libre, tiene muy buena documentación, es escalable para futuros requerimientos, además de tener una rápida forma de desarrollar y desplegar los proyectos, por lo que es una buena elección para un sistema como este.

Django es un framework de alto nivel, el cual es desarrollado en “Python”, esto facilita su desarrollo y mantenimiento, ya que este es un lenguaje con una comunidad muy grande y con amplia documentación. Además de trabajar muy bien con la arquitectura MVC(Modelo-Vista-Controlador). (Django Software Foundation, s.f.)

Ademas de utilizar “**CSS**” para el desarrollo de el diseño del sistema. CSS es un lenguaje de estilo de hojas que permite al autor y a los usuarios adjuntar estilo a documentos como HTML. Separando el estilo de la pagina de la informacion que esta contenga, asi dando un mejor orden y mayor capacidad para desarrollar una experiencia de usuario y interfaz de usuario mejor. (Mora, 2012)

2.2.3 Arquitectura

Para la arquitectura se eligió la arquitectura **MVC**, ya que al trabajar con un patrón de diseño como este permite en el desarrollo y en el mantenimiento del sistema mantener un desarrollo organizado y mantenible, facilitando la modificación de componentes o creación de nuevos sin comprometer componentes ya existentes.

La arquitectura Modelo-Vista-Controlador es un patrón de diseño que busca organizar la aplicación en 3 componentes: el modelo, la vista y el controlador.

- **Modelo:** El modelo representa la estructura de los datos y la lógica de negocio del sistema, por lo que es el responsable de manipular los datos y establecer la lógica con la cual se harán las validaciones.
- **Vista:** La vista se encarga de la presentación de la información al usuario y de recibir las interacciones del usuario con el sistema. Esta muestra los datos entregados desde el modelo y le entrega a el controlador los datos entregados por el usuario.
- **Controlador:** El controlador actúa como intermediaria entre la vista y el modelo, recibiendo las entradas del usuario y entregándoselas al modelo para que este le envíe los nuevos datos a la vista. Controlando así el flujo de los datos.

(Pantoja, 2004)

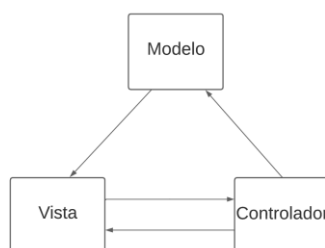


Figura 1: Diagrama MVC

Fuente:Elaboracion propia.

CAPÍTULO 3: PROPUESTA DE SOLUCION

3.1 IDENTIFICACIÓN DE PROCESOS

La identificación de procesos es un componente crucial para el análisis de un sistema informático, ya que los procesos en el contexto del análisis para un sistema se pueden definir como una serie de actividades las cuales se compactan en un proceso, el cual hará el sistema.

Añadir docente: Añadir un docente al sistema para gestionar su carga laboral.

Añadir carga del docente: Añadir carga laboral al docente seleccionado.

Quitar carga del docente: Quitar carga que tenga el docente.

Modificar carga del docente: Modificar el tipo de carga que tiene el docente.

Calculo Horas límite de cada docente: Hacer el cálculo de las horas que está cumpliendo el docente y si llega a su límite.

Generación informe del docente: Se generará una planilla para el docente seleccionado el cual contendrá la información de la carga de cada docente

Generación reporte semestral: Se generará una planilla con la información de todos los docentes, indicando sus cargas académicas.

Edición de información del docente: Se editará la información adicional del docente

Añadir Asignatura: Añadir una asignatura al sistema

Edición de la asignatura: Se editarán los datos de la asignatura

Iniciar sesión: Se iniciará sesión en el sistema.

Registrar usuarios al sistema: Se registrarán los datos del usuario en el sistema.

Editar Permisos: Se editarán los permisos que tienen los usuarios

Iniciar periodo: Se iniciará el periodo semestral.

Crear paralelo: Se creará un nuevo paralelo en el periodo actual.

Cada proceso identificado fue identificado mediante la solicitud de requerimientos que tiene el usuario del sistema. Por lo que se tuvo que conversar con personas las cuales han tenido experiencia en el proceso de la gestión de la carga de los docentes. Luego de varias conversaciones juntos a distintas personas con esta experiencia se llegó a identificar los 15 procesos mostrados anteriormente.

Siguiendo el con la propuesta de la solución, a continuación, se mostrará un diagrama de casos de usos y luego se detallará cada caso de uso.

3.2 DIAGRAMA DE CASOS DE USO

En la figura número 2 se pueden ver un diagrama de casos de uso del sistema de carga docente, en el cual podemos ver los nombres de los casos de uso que se identificaron y los distintos tipos de usuarios que existirán en el sistema. Los distintos usuarios tienen distintos permisos y relaciones con los casos de uso identificados.

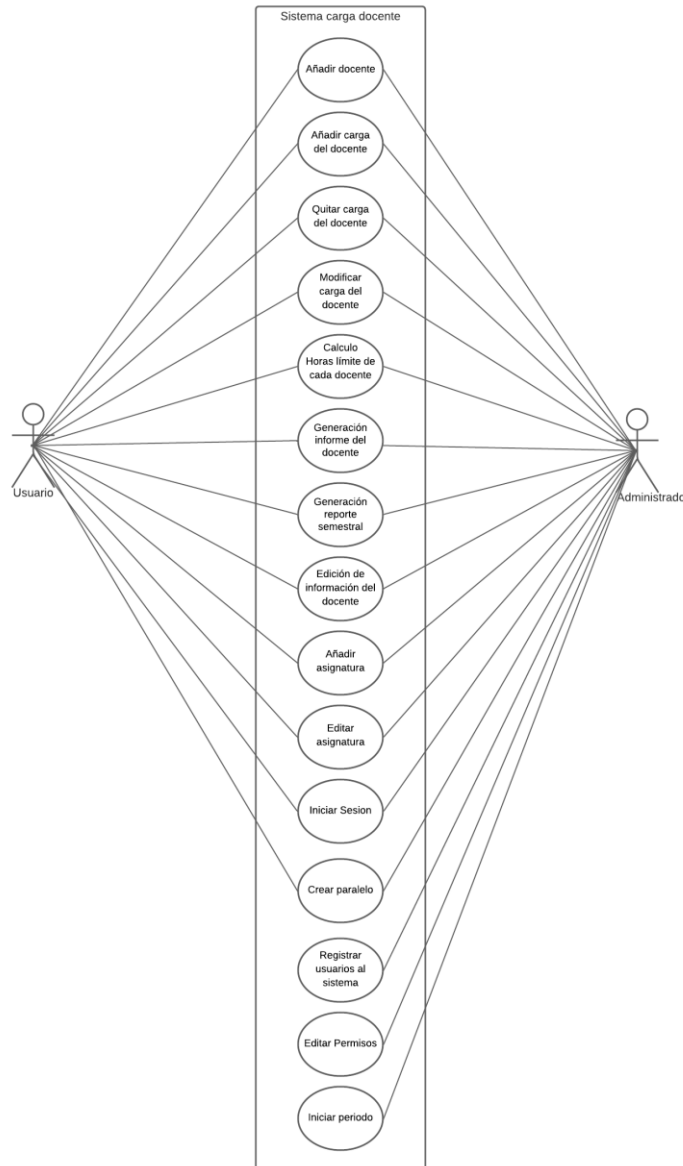


Figura 2: Diagrama de casos de uso.
Fuente:Elaboracion propia.

3.3 CASOS DE USO

En esta sección se detallará cada caso de uso mediante su respectiva tabla, con el propósito de detallar el flujo de los eventos de cada caso de uso, descripción, y actores.

3.2.1 Añadir docente

Tabla 1: Caso de uso extendido 1.

Fuente: Elaboración Propia.

Nombre: Añadir docente
Descripción: Añadir un docente al sistema
Actores: Usuario
Precondiciones: Debe estar ingresado como usuario
Requisitos no funcionales: Ninguno
Flujo de eventos: 1. El usuario presionara "Añadir docente", en la vista "docentes". 2. Se desplegará un modal para agregar un nuevo docente. 3. Si los datos del formulario no están completos, se mostrará una alerta. 4. Si los datos del formulario son correctos, se añadirá el nuevo docente al sistema.
Post-condiciones: Se añadirá el nuevo docente al sistema al sistema.

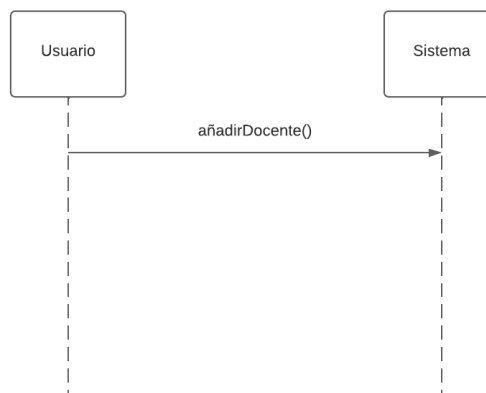


Figura 3: Diagrama de secuencia Caso de uso 1.

Fuente:Elaboracion propia.

El respectivo caso de uso corresponde a adición de un nuevo docente al sistema para luego ser utilizados para añadirle su respectiva carga académica.

En la figura número 3 podemos ver el flujo que mantiene el caso de uso mediante un diagrama de secuencia.

3.2.2 Añadir carga del docente

Tabla 2: Caso de uso extendido 2.

Fuente: Elaboración Propia.

Nombre: Añadir carga del docente
Descripción: Añadir carga laboral al docente seleccionado
Actores: Usuario
Precondiciones: Debe existir algún docente en el sistema
Requisitos no funcionales: Ninguno
Flujo de eventos: 1. El usuario presionara "agregar carga" en el docente que le desea añadir carga 2. Se desplegará una vista para agregar carga al docente seleccionado 3. Se presionará "Guardar" para guardar la carga docente añadida 4. En caso de que sobrepase la carga limite, no se podrá añadir más carga al docente seleccionado
Post-condiciones: Los datos guardados se enviarán al sistema y al docente se le será añadido la cantidad de carga ingresada

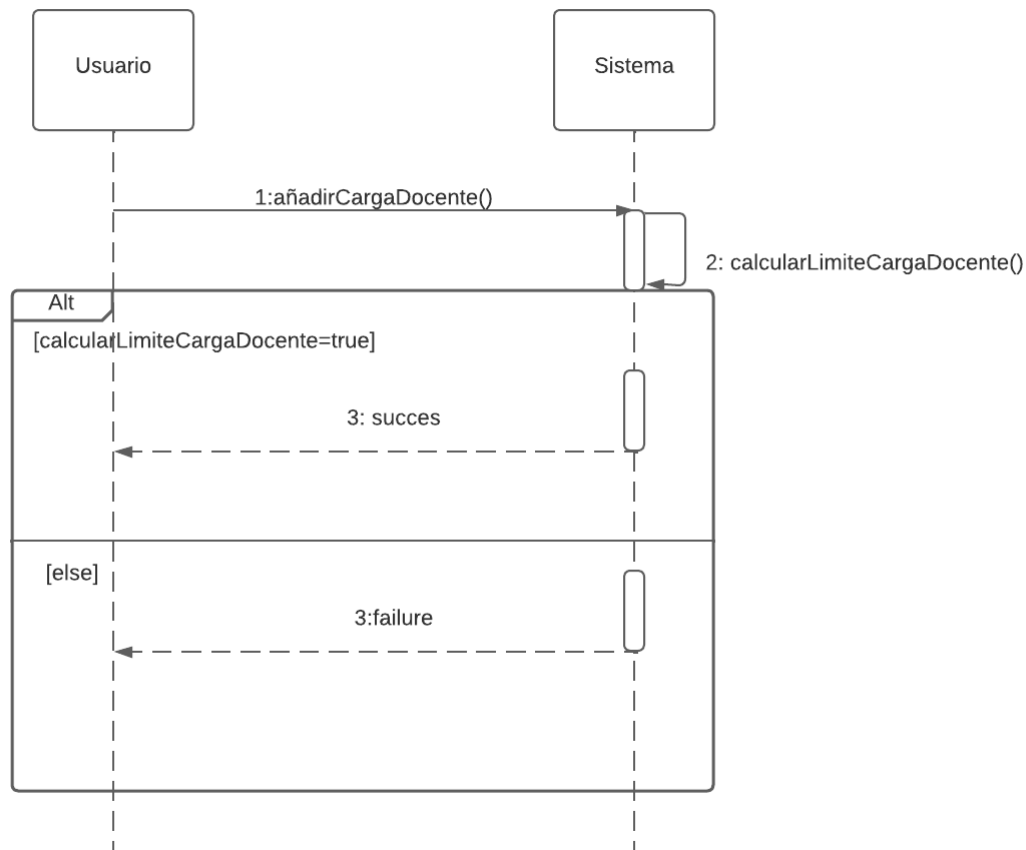


Figura 4: Diagrama de secuencia Caso de uso 2.

Fuente:Elaboracion propia.

El respectivo caso de uso corresponde al caso de uso principal dentro del sistema, ya que este es el que añade la carga academica dependiendo del docente seleccionado y si esque este mismo ya contiene la horas limites.

En la figura número 4 podemos ver el flujo del caso de uso, evidenciando que este puede salir exitoso o fallando dependiedo si el calculo del limite de la carga del docente sea verdadero. Osea que el docente ya este en el limite o no.

3.2.3 Quitar carga del docente

Tabla 3: Caso de uso extendido 3.

Fuente: Elaboración Propia.

Nombre: Quitar carga del docente
Descripción: Quitar carga que tenga el docente
Actores: Usuario
Precondiciones: El docente debe contener carga
Requisitos no funcionales: Ninguno
Flujo de eventos: 1. El usuario presionara "editar carga" en el docente que desee editar la carga que tiene 2. Se desplegará una vista con la carga y la opción de quitar la carga existente 3. Se quitará la carga del docente 4. Se guardará en el sistema la modificación de la carga del docente
Post-condiciones: La modificación de la carga del docente seleccionado se actualizará en el sistema

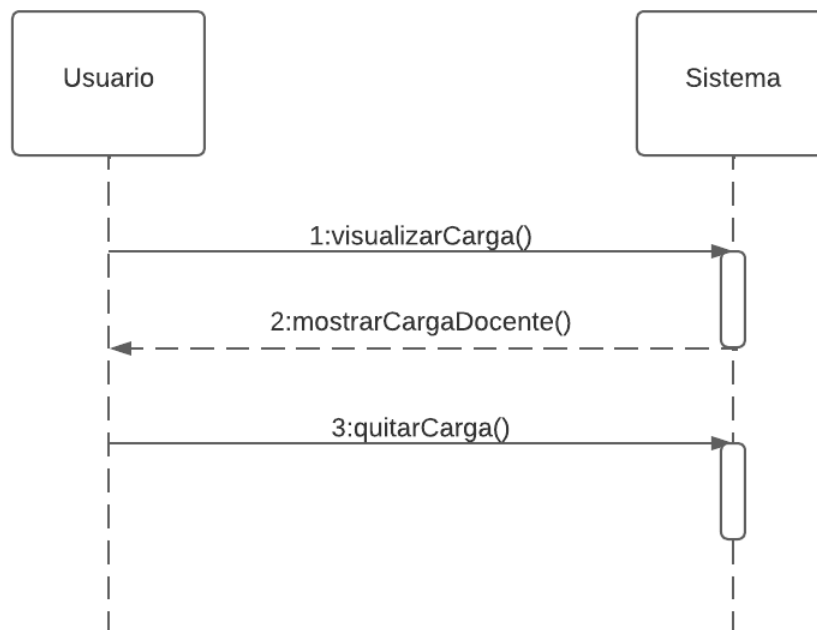


Figura 5: Diagrama de secuencia Caso de uso 3.

Fuente:Elaboracion propia.

El correspondiente caso de uso trata sobre la eliminación de carga a su respectivo docente, mediante tal como dice en la figura número 5, primero la visualización de las cargas que tiene y luego la eliminación de la carga seleccionada.

3.2.4 Modificar carga del docente

Tabla 4: Caso de uso extendido 4.
Fuente: Elaboración Propia.

Nombre: Modificar carga del docente
Descripción: Modificar la carga que contiene el docente
Actores: Usuario
Precondiciones: El docente debe contener carga
Requisitos no funcionales: Ninguno
Flujo de eventos: 1. El usuario presionara "editar carga" en el docente que desee editar la carga que tiene 2. Se desplegará una vista con la carga y la opción de editar la carga existente 3. Se guardará en el sistema la modificación de la carga del docente
Post-condiciones: La modificación de la carga del docente seleccionado se actualizará en el sistema

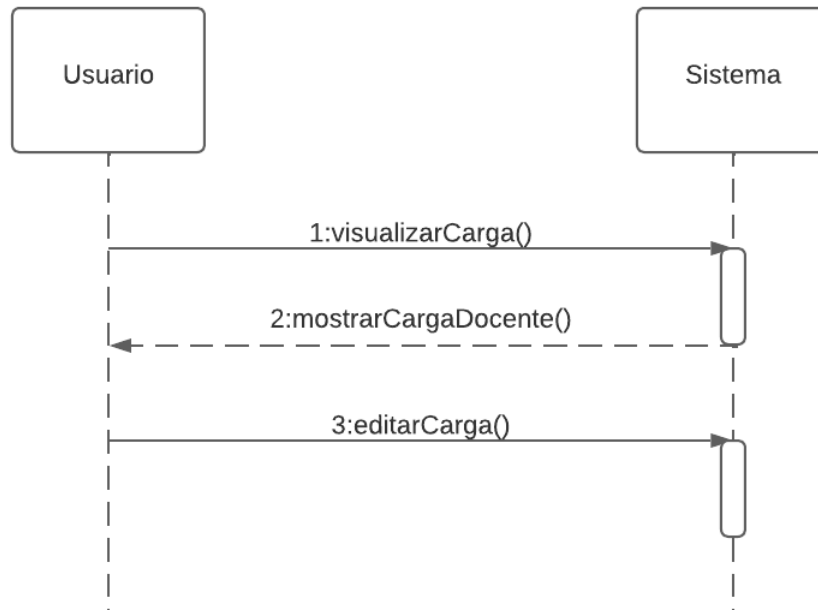


Figura 6: Diagrama de secuencia Caso de uso 4.
Fuente:Elaboracion propia.

La correspondiente figura número 6 tiene como flujo la visualización de la carga que contiene el docente y para terminar el flujo el usuario envía la edición de la carga del docente seleccionado.

Este diagrama de secuencia corresponde al caso de uso 4, el cual modifica la carga del docente seleccionado para así mantener la información actualizada si es que fuera necesario.

3.2.5 Calculo Horas límite de cada docente

Tabla 5: Caso de uso extendido 5.

Fuente: Elaboración Propia.

Nombre: Calculo Horas límite de cada docente
Descripción: Hacer el cálculo de las horas que está cumpliendo el docente y si llego a su límite
Actores: Usuario
Precondiciones: Debe existir un docente
Requisitos no funcionales: Ninguno
Flujo de eventos: 1. El sistema identifica si el docente llego al límite de horas 2. En el caso que la respuesta sea si, el sistema envía una alerta diciendo que el docente ya llego a su límite de horas 3. En el caso que la respuesta sea no, el sistema no envía ninguna alerta y permite al usuario agregar carga al docente
Post-condiciones: Se muestra si el docente llego al límite de sus horas laborales

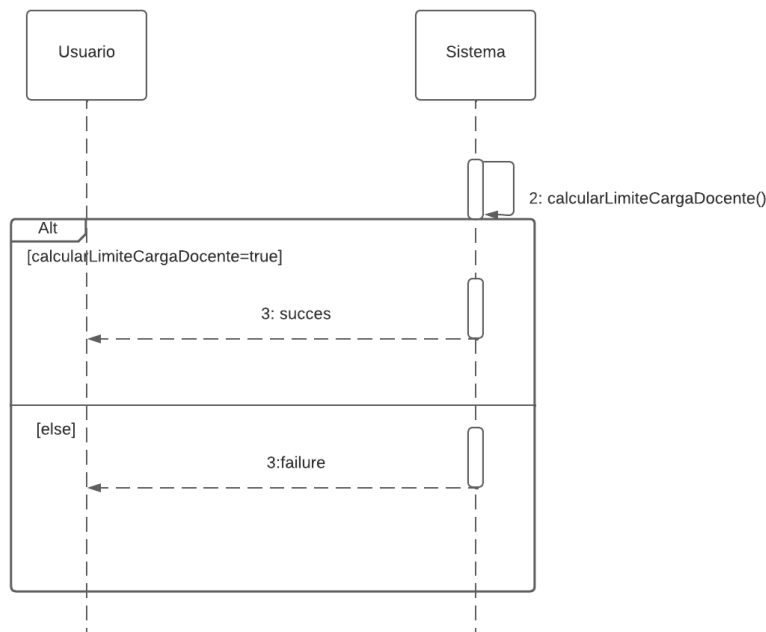


Figura 7: Diagrama de secuencia Caso de uso 5.

Fuente:Elaboracion propia.

El caso de uso correspondiente en este punto es el caso de uso número 5 el cual corresponde al calculo de las horas limites que tiene el docente. En este caso de uso el

sistema es el actor principal, ya que este calcula la cantidad de horas que tiene el docente dependiendo de la cantidad de carga que este tenga asignada. Esto se puede verificar con la figura número 7, en la secuencia evidenciada en el diagrama se puede ver como este tiene los 2 casos, si el docente ya esta en el limite de horas o no.

3.2.6 Generación informe del docente

Tabla 6: Caso de uso extendido 6.

Fuente: Elaboración Propia.

Nombre: Generación informe del docente
Descripción: Se generará una planilla para el docente seleccionado el cual contendrá la información de la carga de cada docente
Actores: Usuario
Precondiciones: Debe existir un docente
Requisitos no funcionales: Ninguno
Flujo de eventos: 1. El usuario presionara "ver planilla", en el docente seleccionado 2. El sistema genera la plantilla con la información detallada del docente seleccionado 3. El sistema mostrara una planilla con la información detallada del docente seleccionado
Post-condiciones: Se muestra la información detallada del docente, mediante una planilla

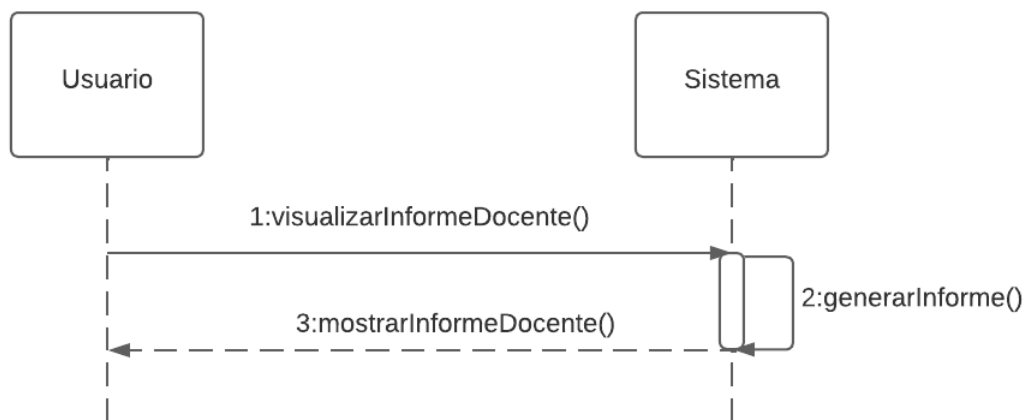


Figura 8: Diagrama de secuencia Caso de uso 6.

Fuente:Elaboracion propia.

El correspondiente caso de uso es el caso de uso destinado a la generación y visualización de un informe del docente seleccionado conteniendo información sobre sus cargas actuales y sus datos.

Tal cual como se ve en la figura número 8, se puede ver que el usuario solicita el informe, el sistema lo genera y luego se lo muestra.

3.2.7 Generación reporte semestral

Tabla 7: Caso de uso extendido 7.

Fuente: Elaboración Propia.

Nombre: Generación reporte semestral
Descripción: Se generará una planilla con la información de todos los docentes, indicando sus cargas académicas.
Actores: Usuario
Precondiciones: Debe existir un docente con carga asignada
Requisitos no funcionales: Ninguno
Flujo de eventos: 1. El usuario selecciona el semestre para la generación del informe 2. El usuario presionara "generar reporte" 3. El sistema generara un reporte con la información detallada del semestre seleccionado.
Post-condiciones: Se muestra la información detallada del semestre, mediante un reporte.

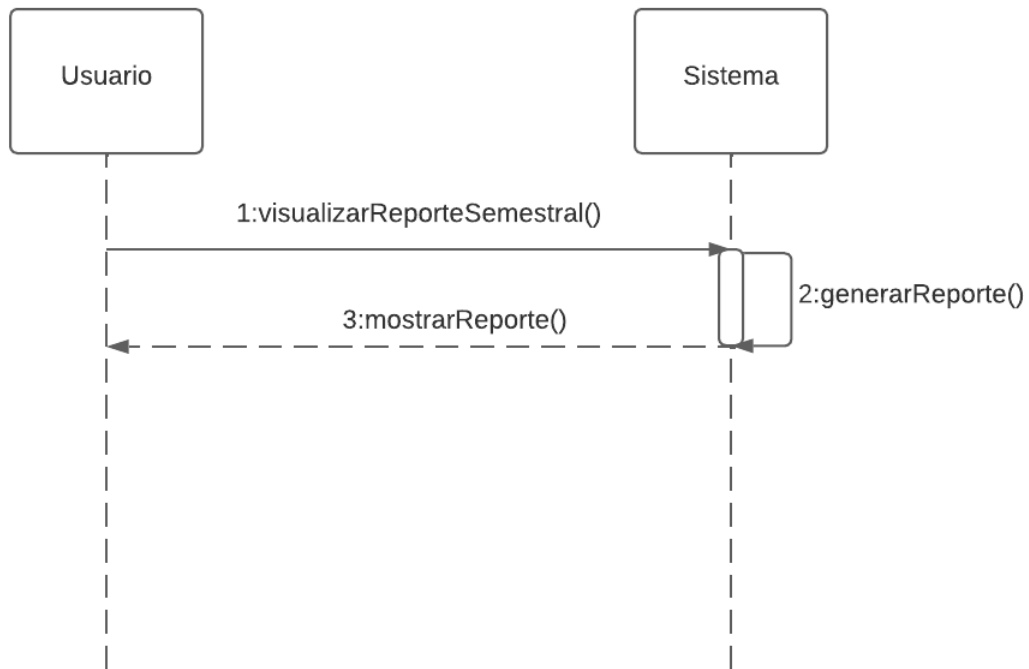


Figura 9: Diagrama de secuencia Caso de uso 7

Fuente:Elaboracion propia.

El caso de uso correspondiente es el caso de uso encargado de la visualización y generación del reporte semestral. Este reporte semestral es un reporte que contiene la información de todos los docentes junto a su carga asignada, este reporte se genera dependiendo del semestre seleccionado.

En la figura número 9 podemos ver un diagrama de secuencia en el cual se muestra como se genera el reporte y se le muestra al usuario.

3.2.8 Edición de información del docente

Tabla 8: Caso de uso extendido 8.

Fuente: Elaboración Propia.

Nombre: Edición de información del docente
Descripción: Se editará la información adicional del docente
Actores: Usuario
Precondiciones: Debe existir un docente
Requisitos no funcionales: Ninguno
Flujo de eventos: 1. El usuario presionara "editar docente" en el docente deseado 2. El sistema mostrará la información actual del docente y permitirá editarla 3. El usuario editara la información que quiera actualizar 4. El usuario presionara "Guardar" 5. El sistema guardara la nueva información del docente
Post-condiciones: Se guardará la nueva información del docente seleccionado

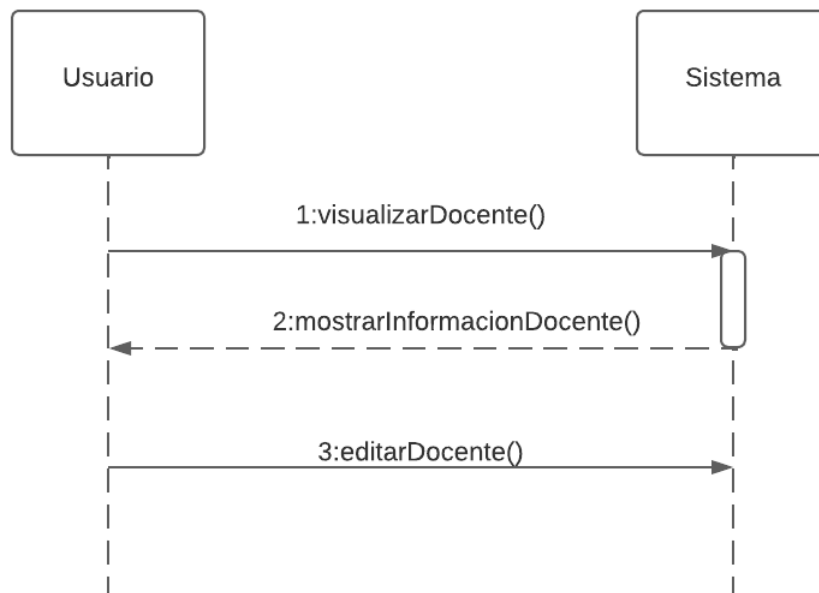


Figura 10: Diagrama de secuencia Caso de uso 8.

Fuente:Elaboracion propia.

Este caso de uso corresponde a la edicion de la informacion del docente, esto tiene la finalidad de mantener la informacion de cada docente actualizada y lista para la generacion de un futuro reporte.

En la figura número 10 se puede ver la secuencia que demuestra el transcurso de los hechos, comenzando con la solicitud de la visualización de los datos y luego el envío de los datos editados del docente seleccionado.

3.2.9 Añadir Asignatura

Tabla 9: Caso de uso extendido 9.

Fuente: Elaboración Propia.

Nombre: Añadir Asignatura
Descripción: Añadir una asignatura al sistema
Actores: Usuario
Precondiciones: Ninguna
Requisitos no funcionales: Ninguno
Flujo de eventos: 1. El usuario presionara "Añadir asignatura", en la vista "asignaturas". 2. Se desplegará un modal para agregar una nueva asignatura. 3. Si los datos del formulario no están completos, se mostrará una alerta. 4. Si los datos del formulario son correctos, se añadirá la nueva asignatura al sistema.
Post-condiciones: Se agrega la nueva asignatura al sistema.

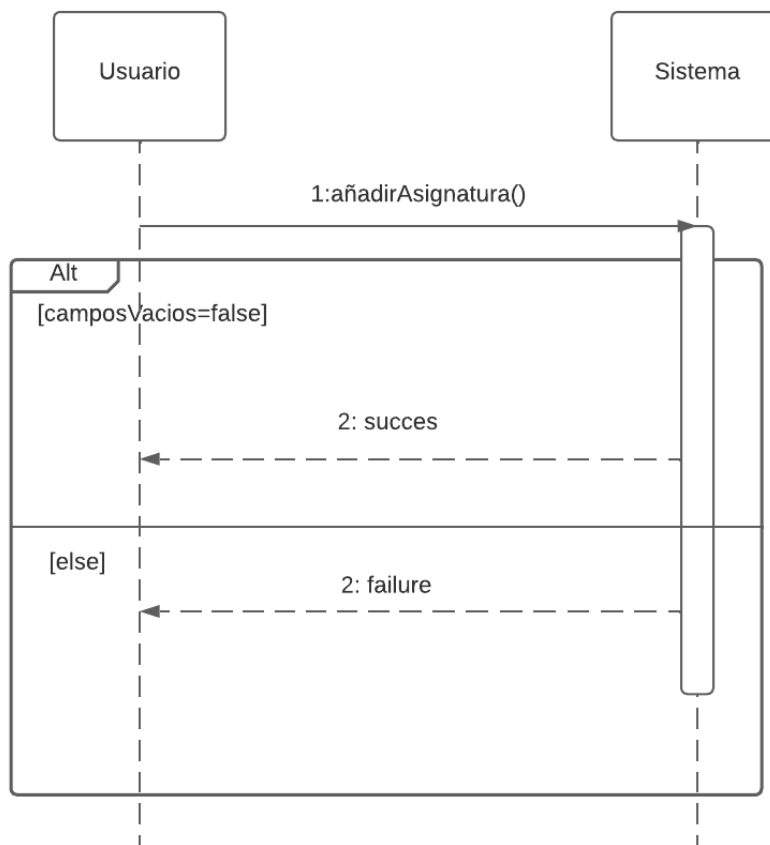


Figura 11: Diagrama de secuencia Caso de uso 9.
Fuente:Elaboracion propia.

Este caso de uso corresponde a la adición de una nueva asignatura al sistema, esta se realiza mediante un modal. Tal cual como se ve en la figura número 11, se puede ver que el usuario envía los datos al añadir la asignatura y el sistema responde con éxito o no si es que los campos están vacíos o son válidos.

3.2.10 Edición de la asignatura

Tabla 10: Caso de uso extendido 10.

Fuente: Elaboración Propia.

Nombre: Edición de la asignatura
Descripción: Se editarán los datos de la asignatura
Actores: Usuario
Precondiciones: Debe existir una asignatura
Requisitos no funcionales: Ninguno
Flujo de eventos: 1. El usuario presionara "editar asignatura" en la asignatura deseada. 2. El sistema mostrará la información actual de la asignatura y permitirá editarla. 3. El usuario editara la información que quiera actualizar. 4. El usuario presionara "Guardar". 5. El sistema guardara la nueva información de la asignatura.
Post-condiciones: Se guardará la nueva información de la asignatura seleccionada.

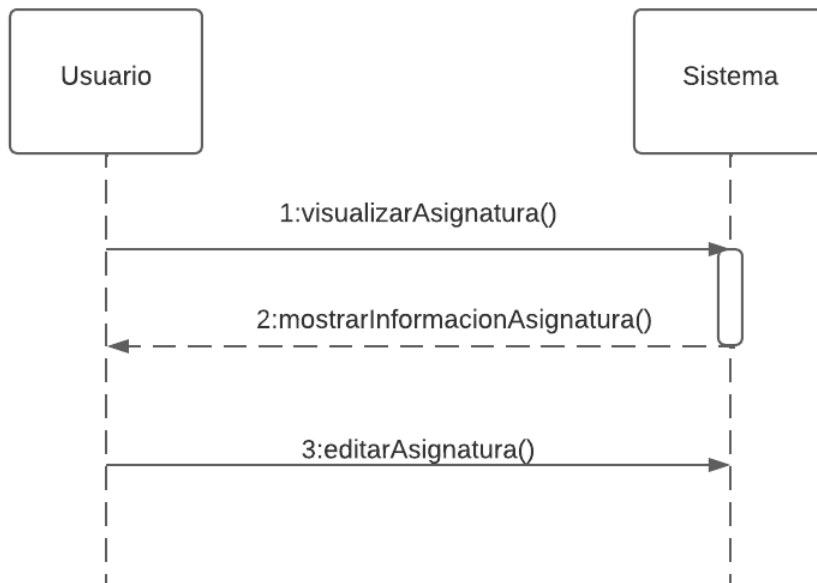


Figura 12: Diagrama de secuencia Caso de uso 10.

Fuente:Elaboracion propia.

El presente caso de uso corresponde a la modificación de una asignatura la cual ya debe existir. Según la figura número 12 el usuario solicita al sistema la visualización de la

asignatura con sus datos existentes y luego le envía al sistema los datos los cuales se realizó la modificación.

3.2.11 Iniciar sesión

Tabla 11: Caso de uso extendido 11.

Fuente: Elaboración Propia.

Nombre: Iniciar sesión.
Descripción: Se iniciará sesión en el sistema.
Actores: Usuario.
Precondiciones: Los datos del usuario deben estar ingresados en el sistema.
Requisitos no funcionales: Ninguno.
Flujo de eventos: 1. El usuario ingresara su usuario. 2. El usuario ingresara su contraseña. 3. Si sus datos son los correctos, inicia sesión en el sistema. 4. Si sus datos son incorrectos, no inicia sesión en el sistema.
Post-condiciones: El usuario inicia sesión en el sistema.

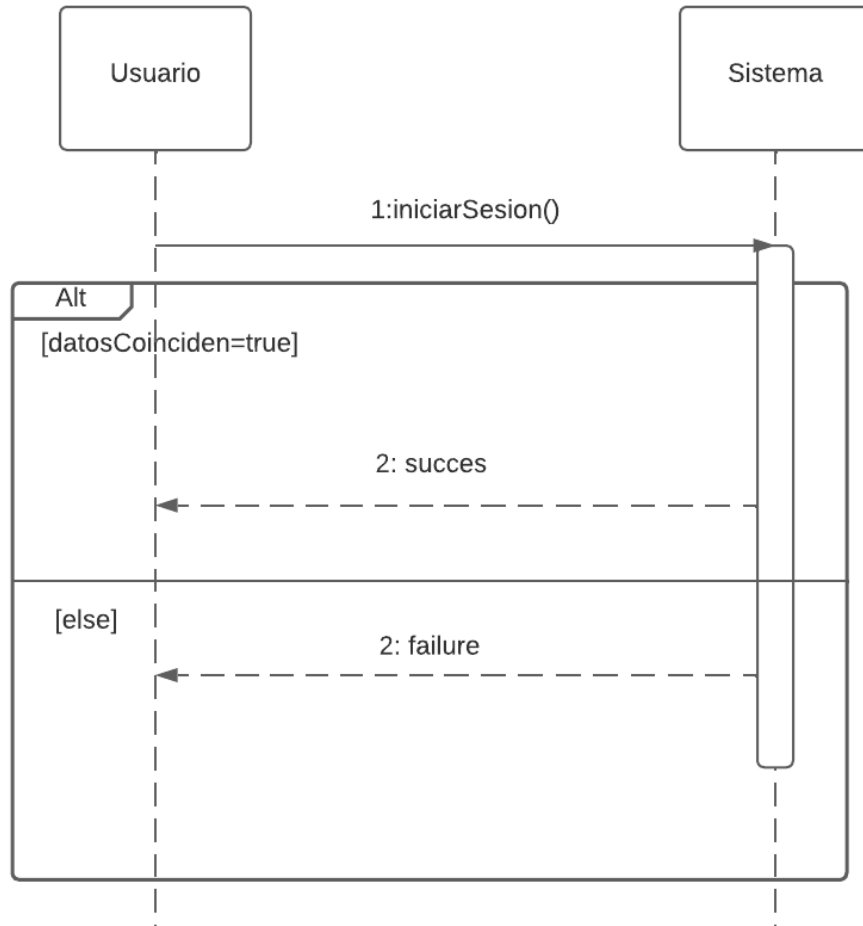


Figura 13: Diagrama de secuencia Caso de uso 11.

Fuente:Elaboracion propia.

El caso de uso correspondiente a la figura número 13 es el caso de uso del inicio de sesión. En el diagrama de secuencia se puede visualizar como el usuario solicita el inicio de sesión, el cual será con su usuario y su contraseña, luego el sistema valida si los datos coinciden y le permite el inicio de sesión.

3.2.12 Registrar usuarios al sistema

Tabla 12: Caso de uso extendido 12.

Fuente: Elaboración Propia.

Nombre: Registrar usuarios al sistema
Descripción: Se registrarán los datos del usuario en el sistema.
Actores: Administrador.
Precondiciones: Ninguno
Requisitos no funcionales: Ninguno.
Flujo de eventos: 1. El administrador entra a la vista “usuarios” desde la vista “sistema”. 2. El administrador presiona “agregar usuario”. 3. Se desplegará un modal para agregar un nuevo usuario. 4. El administrador ingresa los datos. 5. Si los datos están completos, se registra el nuevo usuario. 6. Si los datos están incompletos, no se registra el nuevo usuario.
Post-condiciones: Se registra un nuevo usuario.

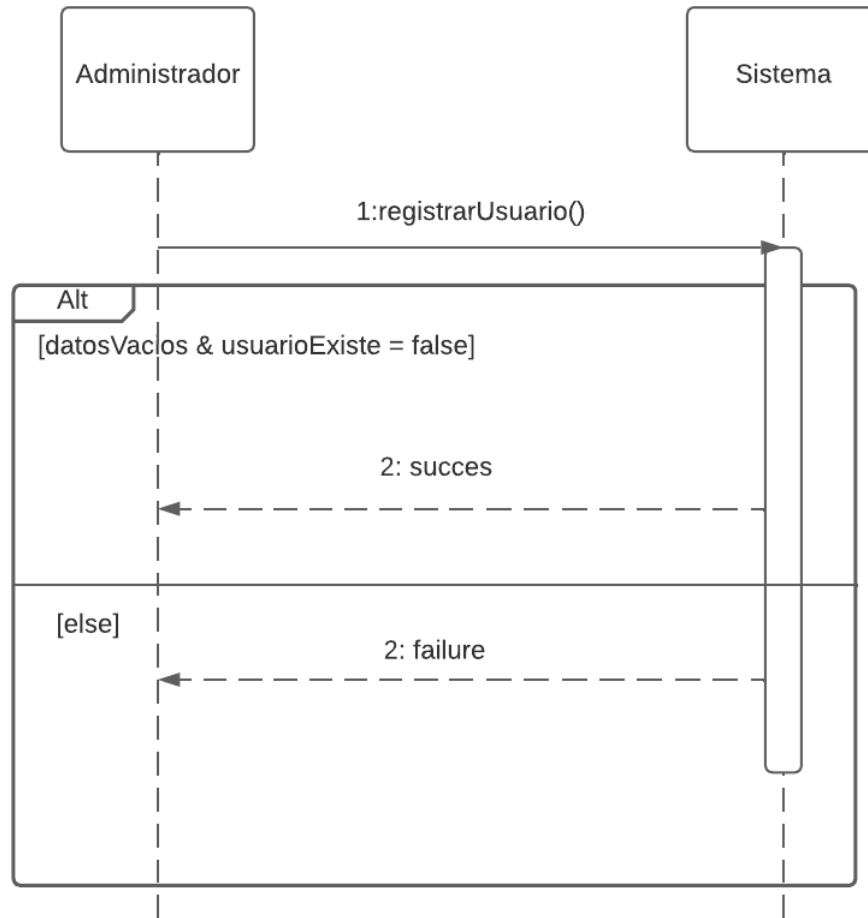


Figura 14: Diagrama de secuencia Caso de uso 12.

Fuente:Elaboracion propia.

El correspondiente caso de uso describe el registro de nuevos usuarios hacia el sistema, este registro debe ser realizado por un administrador. Tal cual se ve en la figura número 14, se puede ver como el administrador envía los datos de registro hacia el sistema y el sistema le envía la respuesta si el usuario ha sido registrado correctamente o no.

3.2.13 Editar Permisos

Tabla 13: Caso de uso extendido 13.

Fuente: Elaboración Propia.

Nombre: Editar Permisos
Descripción: Se editarán los permisos que tienen los usuarios
Actores: Administrador.
Precondiciones: Ninguno
Requisitos no funcionales: Ninguno.
Flujo de eventos: 1. El administrador entra a la vista “usuarios” desde la vista “Sistema”. 2. El administrador presiona la opción “dar permisos” en el respectivo usuario. 3. El sistema mostrara una alerta. 4. El administrador presionara “confirmar” si desea dar los permisos. 5. El administrador presionara “cancelar” si no desea dar los permisos.
Post-condiciones: Se editaran los permisos del usuario seleccionado.

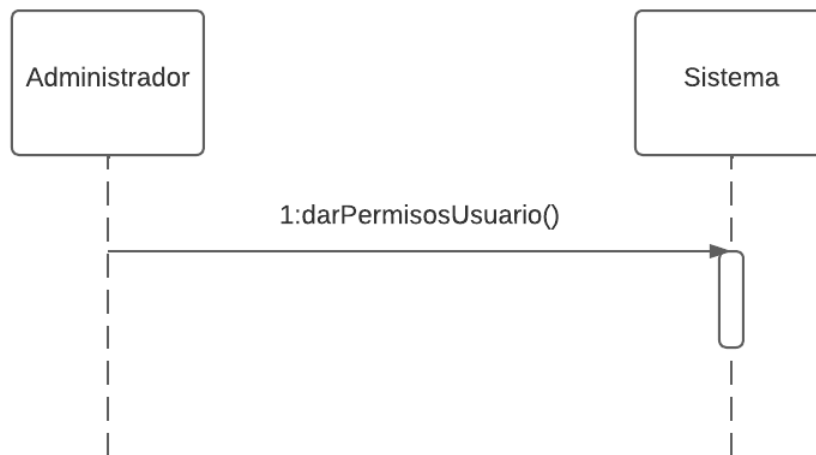


Figura 15: Diagrama de secuencia Caso de uso 13.

Fuente:Elaboracion propia.

El caso de uso corresponde a la modificación de los permisos de los usuarios existentes. Tal como se ve en la figura número 15, el administrador del sistema es el que hace la modificación de los permisos hacia un usuario ya existente.

3.2.14 Iniciar periodo

Tabla 14: Caso de uso extendido 14.

Fuente:Elaboracion propia.

Nombre: Iniciar periodo
Descripción: Se iniciará el periodo semestral
Actores: Administrador.
Precondiciones: Ninguno
Requisitos no funcionales: Ninguno.
Flujo de eventos: 1. El administrador entra a la vista “Sistema”. 2. El administrador entra a la vista “Periodo”. 3. El administrador presiona “Iniciar nuevo periodo” 4. El sistema mostrara una alerta. 5. El administrador ingresa los datos y presiona “confirmar” 6. El sistema inicia un nuevo periodo
Post-condiciones: El sistema inicia un nuevo periodo con los datos enviado.

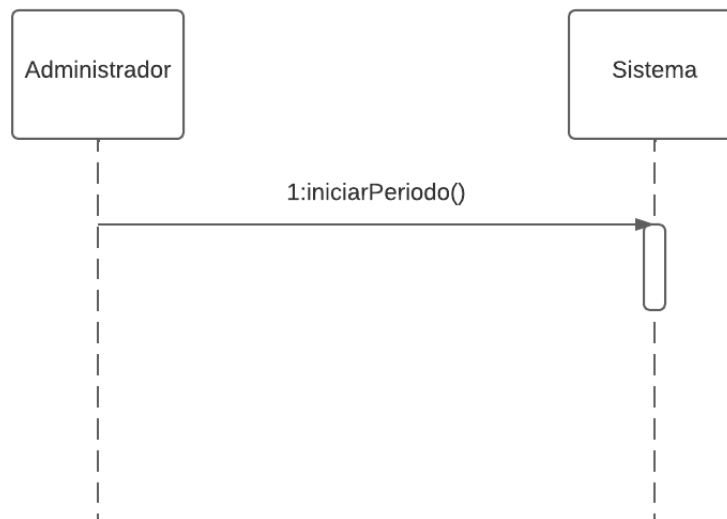


Figura 16: Diagrama de secuencia Caso de uso 14.

Fuente:Elaboracion propia.

El correspondiente caso de uso corresponde al inicio de un nuevo perío electivo de clases, osea, un semestre. Por ejemplo el semestre 2023-1 que corresponde al primer semestre del año 2023.

3.2.15 Crear paralelo

Tabla 15: Caso de uso extendido 15.

Fuente:Elaboracion propia.

Nombre: Crear paralelo
Descripción: Se creará un nuevo paralelo en el periodo actual
Actores: Usuario
Precondiciones: Haber iniciado un periodo
Requisitos no funcionales: Ninguno.
Flujo de eventos: 1. El usuario entra a la vista “Asignaturas” 2. El usuario presiona la asignatura a crearle paralelos. 3. El sistema mostrara una ventana con los paralelos. 4. El usuario presionara “Crear paralelo”. 5. El sistema mostrara una ventana para ingresar los datos. 6. El usuario ingresara los datos y presionara “crear”
Post-condiciones: Se creara un nuevo paralelo para la asignatura

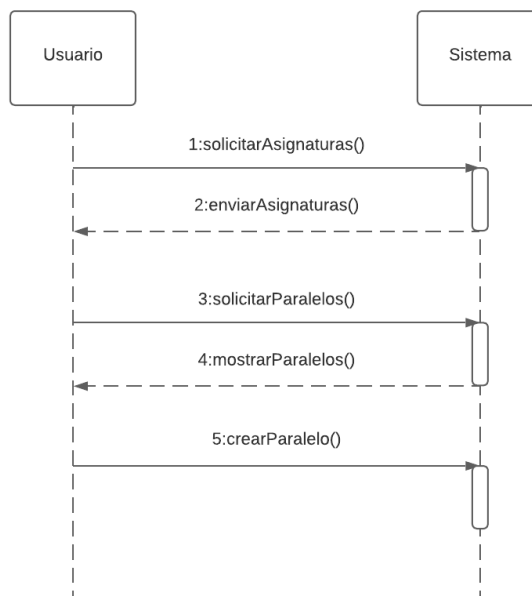


Figura 17: Diagrama de secuencia Caso de uso 15.

Fuente:Elaboracion propia.

El caso de uso 15 corresponde a la creación de los paralelos que contienen cada asignatura en el periodo, ya que cada asignatura puede tener más de un paralelo impartido en los distintos periodos.

3.3 ARQUITECTURA DEL SISTEMA

A continuación, se detallará la arquitectura la cual se propone para el desarrollo del sistema.

Tal como se puede ver en la figura número 16 la arquitectura del sistema se basará en el framework Django, implementando el patrón de diseño MVC, con la cual el usuario tendrá interacción con la vista y entregará los datos y solicitudes a través del controlador, el controlador solicitará o enviará los datos al modelo, para que estos datos sean ingresados en la base de datos o el modelo envíe los respectivos datos a la vista solicitados por el controlador. Se utilizará Django para el desarrollo del sistema y un gestor de base de datos MySQL para la base de datos. Para la interfaz con la cual interactuará el usuario será desarrollada con HTML y con CSS para diseñar una página más intuitiva y amigable.

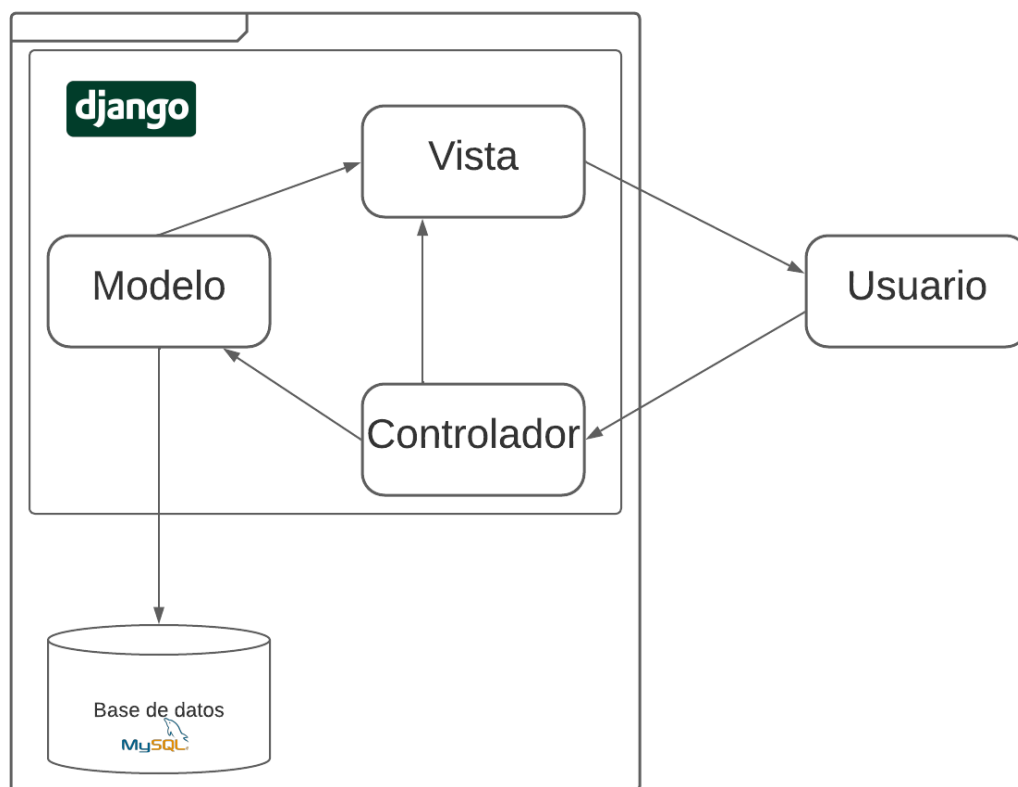


Figura 18: Arquitectura Sistema
Fuente:Elaboración propia.

3.4 DISEÑO DE DATOS

En esta sección se describirá el modelo de datos relacional propuesto para el sistema, detallando cada tabla para entenderlas de una mejor manera y más detallada

3.4.1 Modelo de datos

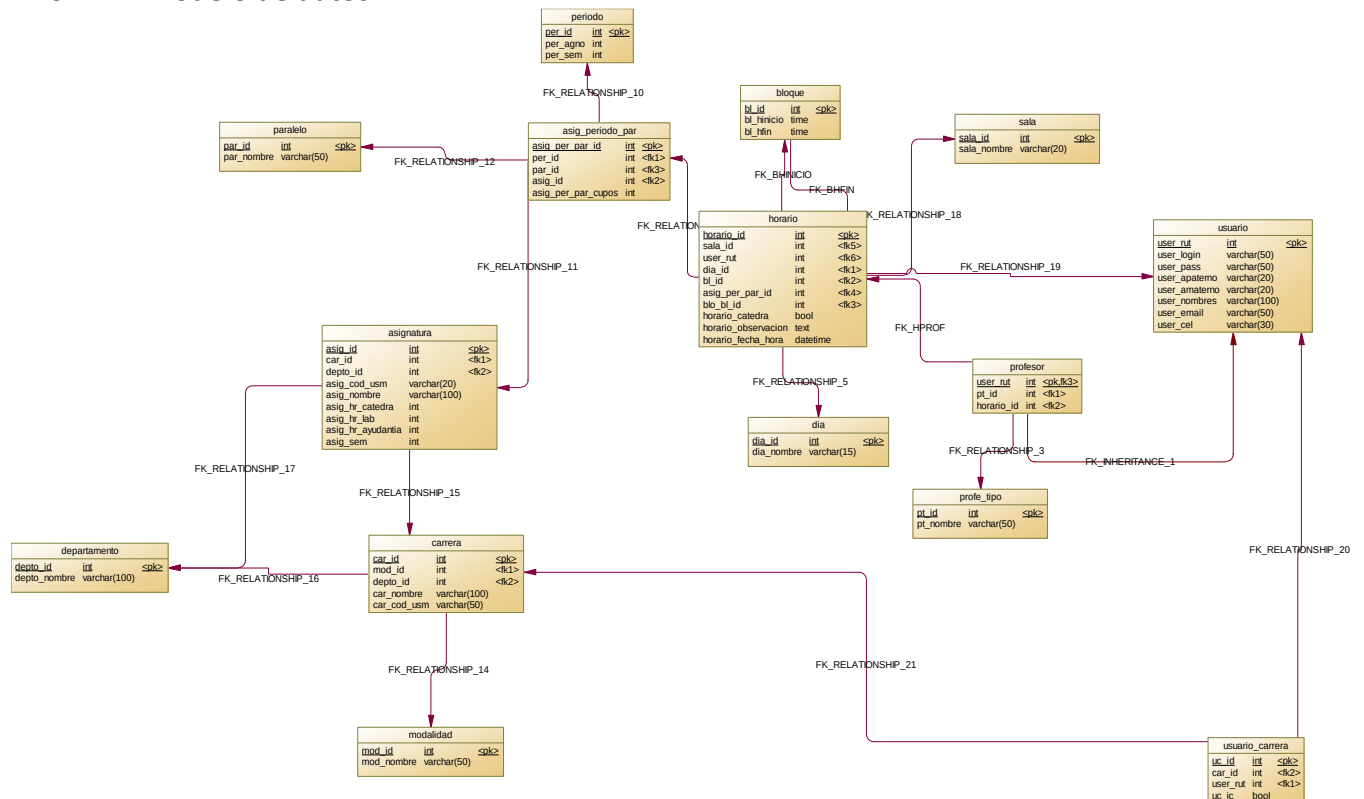


Figura 19: Modelo de datos.

Fuente:Elaboracion propia.

En la figura número 17 se puede ver el modelo de datos propuesto para el sistema, se puede ver un modelo de datos relacional en el que se pueden identificar 15 tablas, las cuales cada tabla contiene sus respectivos datos que almacenaran en la base de datos. A continuación, se detallará cada tabla comenzando de izquierda a derecha como aparece en la figura.

3.4.1.1 Tabla departamento

Tabla 16: Tabla departamento.

Fuente:Elaboracion propia.

Tabla departamento			
Columna	Tipo de dato	Nulo	Comentario
depto_id	int	NO	
depto_nombre	varchar(100)	NO	
Llave primaria	Columna	Comentario	
PK_DEPARTAMENTO	depto_id		
Llave foránea	Columna	Referencia	

La primera tabla identificada es la tabla departamento la cual será encargada de almacenar el id de cada departamento y el nombre de cada departamento, haciendo referencia a los departamentos académicos dentro de la universidad por ejemplo el DEI

Una tabla con sus datos se vería de la siguiente manera:

Tabla 17: Tabla departamento ejemplo.

Fuente:Elaboracion propia.

departamentos	
depto_id	depto_nombre
5	Departamento de electronica e informática

Con este ejemplo podemos ver como se veria la tabla teniendo un funcionamiento normal

3.4.1.2 Tabla paralelo

Tabla 18: Tabla paralelo.
Fuente:Elaboracion propia.

Tabla paralelo			
Columna	Tipo de dato	Nulo	Comentario
par_id	int	NO	
par_nombre	varchar(50)	NO	
Llave primaria	Columna	Comentario	
PK_PARALELO	par_id		
Llave foránea	Columna	Referencia	

La tabla paralelo corresponde a la tabla encargada de mantener los datos relacionado a los paralelos que existen en cada asignatura, por lo que necesitan un identificador y el respectivo nombre del paralelo. A continuación, una tabla de ejemplo.

Tabla 19: Tabla paralelo ejemplo.
Fuente:Elaboracion propia.

paralelo	
par_id	par_nombre
30	705

En la tabla 17 podemos ver los datos correctos de un paralelo llamado 705.

3.4.1.3 Tabla asignatura

Tabla 20: Tabla asignatura.

Fuente:Elaboracion propia.

Tabla asignatura			
Columna	Tipo de dato	Nulo	Comentario
asig_id	int	NO	
car_id	int	NO	
depto_id	int	NO	
asig_cod_usm	varchar(20)	NO	
asig_nombre	varchar(100)	NO	
asig_hr_catedra	int	SI	
asig_hr_lab	int	SI	
asig_hr_ayudantia	int	SI	
asig_sem	int	NO	
Llave primaria	Columna	Comentario	
PK_ASIGNATURA	asig_id		
Llave foranea	Columna	Referencia	
FK_CARRERA	car_id	Tabla carrera	
FK_DEPARTAMENTO	depto_id	Tabla departamento	

La tabla asignatura es un punto importante en el sistema ya que es la tabla que almacena los datos de las asignaturas las cuales corresponderán a carga para el docente. La tabla contiene su identificador, el identificador de la carrera a la que pertenece, el identificador del departamento por el cual es impartida, el “código usm” de la asignatura que corresponde a un código entregado por la universidad, el nombre de la asignatura, la cantidad de horas de catedra, la cantidad de horas de laboratorio, la cantidad de horas de ayudantía y el semestre en el cual es impartida la asignatura. A continuación, un ejemplo de la tabla.

Tabla 21: Tabla asignatura ejemplo.

Fuente:Elaboracion propia.

asignatura								
asig_id	car_id	depto_id	asig_cod_usm	asig_nombre	asig_hr_catedra	asig_hr_lab	asig_hr_ayudantia	asig_sem
1	3	5	EIN413-B	PROGRAMA CION	2	0	0	1

3.4.1.4 Tabla modalidad

Tabla 22: Tabla modalidad.
Fuente:Elaboracion propia.

Tabla modalidad			
Columna	Tipo de dato	Nulo	Comentario
mod_id	int	NO	
mod_nombre	varchar(50)	NO	
Llave primaria	Columna	Comentario	
PK_MODALIDAD	mod_id		
Llave foranea	Columna	Referencia	

La tabla modalidad es la encargada de almacenar los datos de la modalidad. Esta contiene su respectivo identificador y nombre. A continuación, se verá una tabla de ejemplo.

Tabla 23: Tabla modalidad ejemplo
Fuente:Elaboracion propia.

modalidad	
mod_id	mod_nombre
1	Sin mención

En la figura número 21 podemos ver un ejemplo de modalidad con su respectivo identificador y nombre de la modalidad.

3.4.1.5 Tabla carrera

Tabla 24: Tabla carrera
Fuente:Elaboracion propia.

Tabla carrera			
Columna	Tipo de dato	Nulo	Comentario
car_id	int	NO	
mod_id	int	NO	
depto_id	int	NO	
car_nombre	varchar(100)	NO	
car_cod_usm	varchar(50)	NO	
Llave primaria	Columna	Comentario	
PK_CARRERA	car_id		
Llave foranea	Columna	Referencia	
FK_MODALIDAD	mod_id	Tabla modalidad	
FK_DEPARTAMENTO	depto_id	Tabla departamento	

La tabla carrera corresponde a la tabla encargada de almacenar los datos correspondientes a cada carrera, cada carrera tiene su respectivo identificador, modalidad, identificador del departamento al cual pertenece, nombre y su “código usm”, el cual es el código el cual le asigna la universidad. Además de sus respectivas relaciones con otras tablas mediante las llaves foráneas. A continuación, un ejemplo de cómo se vería una tabla.

Tabla 25: Tabla carrera ejemplo
Fuente:Elaboracion propia.

carrera				
car_id	mod_id	depto_id	car_nombre	car_cod_usm
3	1	5	Ingenieria en informatica	29

3.4.1.6 Tabla asig_periodo_par

Tabla 26: Tabla asig_periodo_par
Fuente:Elaboracion propia

Tabla asig_periodo_par			
Columna	Tipo de dato	Nulo	Comentario
asig_per_par_id	int	NO	
per_id	int	NO	
par_id	int	NO	
asig_id	int	NO	
asig_per_par_cupos	int	SI	
Llave primaria	Columna	Comentario	
PK_ASIG_PER_PAR	asig_per_par_id		
Llave foranea	Columna	Referencia	
FK_PERIODO	per_id	Tabla periodo	
FK_PARALELO	par_id	Tabla paralelo	
FK_ASIGNATURA	asig_id	Tabla asignatura	

La tabla asig_periodo_par corresponde a la tabla encargada de almacenar los datos de la relacion entre la asignatura, el periodo en el cual se imparte y el paralelo. La tabla contiene su respectivo identificador, el identificador del periodo al cual pertenece, el paralelo de la asignatura, el identificador de la asignatura y la cantidad de cupos que existen. Ademas de teners sus respectiva relacion mediante las llaves foraneas con las tablas periodo, paralelo y asignatura. A continuacion una tabla de ejemplo

Tabla 27: Tabla asig_periodo_par ejemplo
Fuente:Elaboracion propia

asig_periodo_par				
asig_per_par_id	per_id	par_id	asig_id	asig_per_par_cupos
1	1	30	1	25

3.4.1.7 Tabla periodo

Tabla 28: Tabla periodo
Fuente:Elaboracion propia

Tabla periodo			
Columna	Tipo de dato	Nulo	Comentario
per_id	int	NO	
per_agno	int	NO	
per_sem	int	NO	
Llave primaria	Columna	Comentario	
PK_PERIODO	per_id		
Llave foranea	Columna	Referencia	

La tabla periodo corresponde a la tabla encargada de almacenar los datos del periodo en el cual se impartirá la asignatura, por lo que este necesita almacenar su respectivo identificador, el año del periodo, y el número del semestre. A continuación, un ejemplo de una correcta tabla

Tabla 29: Tabla periodo ejemplo
Fuente:Elaboracion propia

periodo		
per_id	per_agno	per_sem
1	2023	1

3.4.1.8 Tabla horario

Tabla 30: Tabla horario
Fuente:Elaboracion propia

Tabla horario			
Columna	Tipo de dato	Nulo	Comentario
horario_id	int	NO	
sala_id	int	NO	
user_rut	int	NO	
dia_id	int	NO	
bl_id	int	NO	Inicio Bloque
asig_per_par_id	int	NO	
blo_bl_id	int	NO	Fin Bloque
horario_catedra	bool	NO	
horario_observacion	text	NO	
horario_fecha_hora	datetime	NO	
Llave primaria	Columna	Comentario	
PK_HORARIO	horario_id		
Llave foranea	Columna	Referencia	
FK_SALA	sala_id	Tabla sala	
FK_USER	user_rut	Tabla usuario	
FK_DIA	dia_id	Tabla dia	
FK_BLOQUE_INICIO	bl_id	Tabla bloque	
FK_ASIG_PER_PAR	asig_per_par_id	Tabla asig_periodo_par	
FK_BLOQUE_FIN	blo_bl_id	Tabla bloque	

La tabla horaria es una tabla importante dentro del sistema ya que relaciona varias tablas, esta tabla esta encargada de almacenar los datos correspondidos al horario en el que se impartirá la asignatura, por lo que pueden existir más de un horario para la misma asignatura y paralelo ya que la asignatura se puede impartir varias veces a la semana. La tabla almacenara su respectivo identificador, el identificador de la sala, el usuario que la imparte, el identificador del día, el bloque en el que comienza la asignatura, el identificador del paralelo del periodo, el bloque en el que termina la asignatura, un bool para identificar si el horario es de tipo catedra, un texto para observaciones y la fecha en la que se ingresó. A continuación, una tabla correctamente ingresada.

Tabla 31: Tabla horario ejemplo

Fuente:Elaboracion propia

horario									
horario_id	sala_id	user_rut	día_id	bl_id	asig_per_par_id	blo_bl_id	horario_catedra	horario_observacion	horario_fecha_hora
4	4		2	4	1	5	true	sin observacion	28/11

3.4.1.9 Tabla bloque

Tabla 32: Tabla bloque

Fuente:Elaboracion propia

Tabla bloque			
Columna	Tipo de dato	Nulo	Comentario
bl_id	int	NO	
bl_hinicio	time	NO	
bl_hfin	time	NO	
Llave primaria	Columna	Comentario	
PK_BLOQUE	bl_id		
Llave foranea	Columna	Referencia	

La tabla bloque corresponde a los bloques de tiempo que tiene la universidad para impartir las asignaturas, cada bloque tiene su respectivo identificador, horas de inicio y hora de fin. Con lo anterior al ser asignado a alguna asignatura, significará que la asignatura se impartirá entre la hora de inicio y la hora de fin. A continuación, una tabla correctamente ingresada.

Tabla 33: Tabla bloque ejemplo

Fuente:Elaboracion propia

bloque		
bl_id	bl_hinicio	bl_hfin
4	10:10	10:45
5	10:55	11:30

3.4.1.10 Tabla dia

Tabla 34: Tabla dia
Fuente:Elaboracion propia

Tabla dia			
Columna	Tipo de dato	Nulo	Comentario
dia_id	int	NO	
dia_nombre	varchar(15)	NO	
Llave primaria	Columna	Comentario	
PK_DIA	dia_id		
Llave foranea	Columna	Referencia	

La tabla día corresponde a la tabla que almacenara los días de la semana que luego serán utilizados al momento de ingresar un horario. La tabla contiene el identificador del día y el nombre del día. A continuación, un ejemplo.

Tabla 35: Tabla dia ejemplo
Fuente:Elaboracion propia

dia	
dia_id	dia_nombre
2	MARTES

3.4.1.11 Tabla profe_tipo

Tabla 36: Tabla profe_tipo
Fuente:Elaboracion propia

Tabla profe_tipo			
Columna	Tipo de dato	Nulo	Comentario
pt_id	int	NO	
pt_nombre	varchar(50)	NO	
Llave primaria	Columna	Comentario	
PK_PROFESOR_TIPO	pt_id		
Llave foranea	Columna	Referencia	

La tabla profe_tipo corresponde a la tabla que estará a cargo de almacenar los distintos tipos de docentes que pueden impartir asignaturas dentro de la universidad. La tabla almacenará el respectivo identificador del tipo de docente y el nombre del tipo de docente. A continuación, un ejemplo.

Tabla 37: Tabla profe_tipo ejemplo
Fuente:Elaboracion propia

profe_tipo	
pt_id	pt_nombre
1	Planta

3.4.1.12 Tabla sala

Tabla 38: Tabla sala
Fuente:Elaboracion propia

Tabla sala			
Columna	Tipo de dato	Nulo	Comentario
sala_id	int	NO	
sala_nombre	varchar(20)	NO	
Llave primaria	Columna	Comentario	
PK_SALA	sala_id		
Llave foranea	Columna	Referencia	

La tabla sala esta encargada de almacenar los datos de las salas en las cuales se pueden impartir asignaturas dentro de la universidad, por lo que debe almacenar el respectivo id de la sala y su nombre. A continuación, un ejemplo de la tabla.

Tabla 39: Tabla sala ejemplo
Fuente:Elaboracion propia

sala	
sala_id	sala_nombre
4	Lab.informatica 2

3.4.1.13 Tabla profesor

Tabla 40: Tabla profesor
Fuente:Elaboracion propia

Tabla profesor			
Columna	Tipo de dato	Nulo	Comentario
user_rut	int	NO	
pt_id	int	NO	
horario_id	int	NO	
Llave primaria	Columna	Comentario	
PK_USUARIO	user_rut		
Llave foranea	Columna	Referencia	
FK_USUARIO	user_rut	Tabla usuario	
FK_PROFE_TIPO	pt_id	Tabla profe_tipo	
FK_HORARIO	horario_id	Tabla horario	

La tabla profesor corresponde a la tabla encargada de almacenar los datos de los docentes, los cuales deben estar ingresados en el sistema. La tabla almacenara el rut del usuario, el identificador del tipo de docente que le corresponde, y el horario que le corresponde. A continuación, un ejemplo de la tabla

Tabla 41: Tabla profesor ejemplo
Fuente:Elaboracion propia

profesor		
user_rut	pt_id	horario_id
XX.XXX.XXX-1	1	4

3.4.1.14 Tabla usuario

Tabla 42: Tabla usuario
Fuente:Elaboracion propia

Tabla usuario			
Columna	Tipo de dato	Nulo	Comentario
user_rut	int	NO	
user_login	varchar(50)	NO	
user_pass	varchar(50)	NO	
user_apaterno	varchar(20)	NO	
user_amaterno	varchar(20)	NO	
user_nombres	varchar(100)	NO	
user_email	varchar(50)	NO	
user_cel	varchar(30)	NO	
Llave primaria	Columna	Comentario	
PK_USUARIO	user_rut		
Llave foranea	Columna	Referencia	

La tabla usuario corresponde a la tabla que almacenara los usuarios del sistema, estos usuarios podrán ser tanto docentes como ayudantes que se necesiten. La tabla almacenara los datos de cada usuario, cada usuario tendrá su respectivo rut, el nombre con el que iniciaran sesión, la contraseña con la que iniciar sesión, el apellido paterno, apellido materno, ambos nombres, su respectivo email institucional y número de celular de contacto. A continuación, una tabla con usuarios correctamente ingresados.

Tabla 43: Tabla usuario ejemplo
Fuente:Elaboracion propia

usuario							
user_rut	user_login	user_pass	user_apaterno	user_amaterno	user_nombres	user_email	user_cel
XX.XXX.XXX-1	cristian.lara	XXXXX	Lara	Valenzuel	Cristian	correo@usr	9XXXXXXXX
XX.XXX.XXX-2	javier.maldonado	XXXXX	Maldonado	Carmona	Javier	correo@usr	9XXXXXXXX

3.4.1.15 Tabla usuario_carrera

Tabla 44: Tabla usuario_carrera
Fuente:Elaboracion propia

Tabla usuario_carrera			
Columna	Tipo de dato	Nulo	Comentario
uc_id	int	NO	
car_id	int	NO	
user_rut	int	NO	
uc_jc	bool	NO	jefe de carrera
Llave primaria	Columna	Comentario	
PK_USUARIO_CAR	uc_id		
Llave foranea	Columna	Referencia	
FK_CARRERA	car_id	Tabla carrera	
FK_USUARIO	user_rut	Tabla usuario	

La tabla usuario_carrera corresponde a la tabla que almacena la relación que tiene el usuario con la carrera, para identificar si es que es jefe de carrera de esta o no. La tabla almacena el identificador del usuario_carrera, el identificador de la carrera, el rut del usuario y un bool para identificar si es el jefe de carrera o no. Relacionándose con las tablas carrera y usuario para obtener los datos. A continuación, un ejemplo de la tabla.

Tabla 45: Tabla usuario_carrera ejemplo
Fuente:Elaboracion propia

usuario_carrera			
uc_id	car_id	user_rut	uc_jc
2	3	XX.XXX.XXX-2	true
1	3	XX.XXX.XXX-1	false

3.5 INTERFAZ DEL SISTEMA

Esta sección de la solución corresponde a la propuesta de interfaz con la que el sistema podría funcionar correctamente. Esta propuesta, es una aplicación web que nos permite visualizar el proceso de una manera mas amigable, comprensible. Esta solución es más amigable para nuevos usuarios en el proceso y usuarios que ya conozcan el proceso.

A continuación, se mostrarán imágenes de mockups de propuesta para la solución. Las imágenes de la solución se mostrarán en 6 puntos, cada uno con su respectivo contexto.

3.5.1 Inicio de sesión

En este punto se mostrará la propuesta del inicio de sesión.

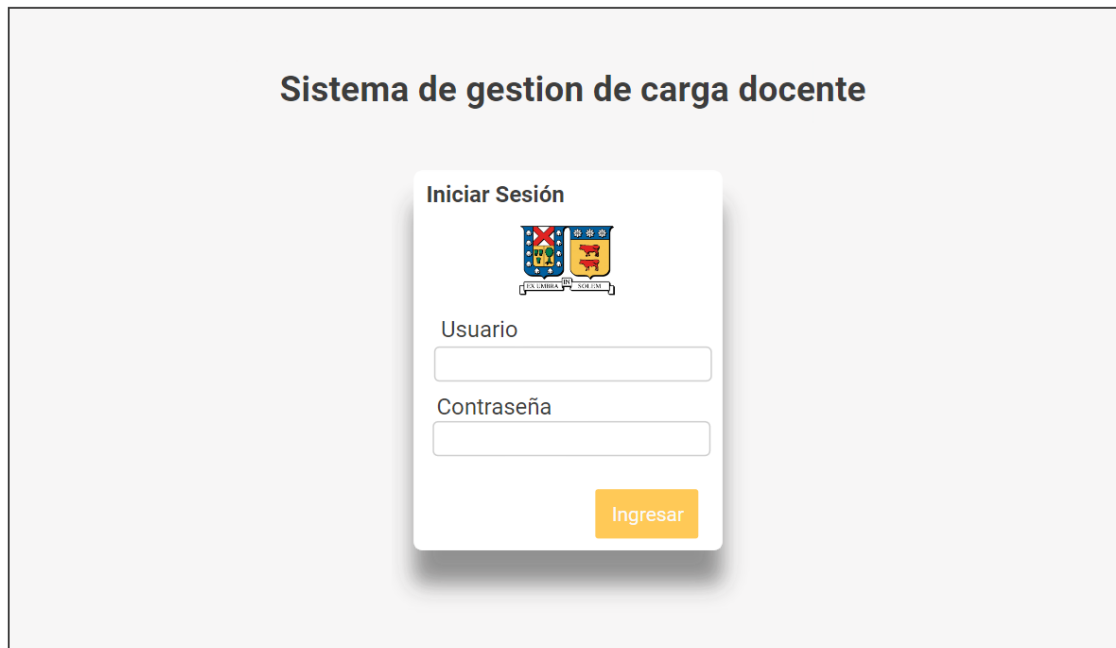


Figura 20:Inicio sesión mockup

Fuente:Elaboracion propia

En la figura número 21 podemos ver el inicio de sesión con los campos necesarios y el botón para iniciar la sesión al sistema.

3.5.2 Menú



Figura 21: Menú mockup
Fuente:Elaboracion propia

En la figura número 22 se puede visualizar el menú propuesto, este menú contiene 4 botones, 3 botones en azul para las principales acciones y el botón de sistema.

3.5.3 Docente

En este punto se mostrará la propuesta de la opción de docentes.

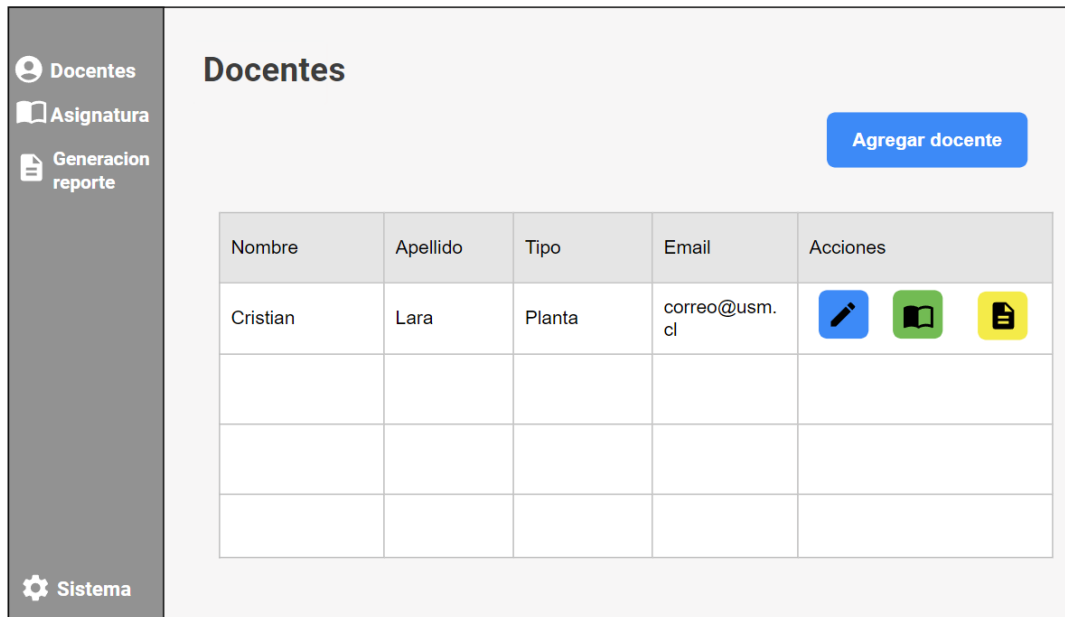


Figura 22: Docentes mockup
Fuente:Elaboracion propia

En la figura número 23 podemos ver la vista de docentes en la cual se encuentra una tabla en la que se visualizarán los docentes que estén en el sistema junto a su información. Cada

docente tiene una columna llamada “acciones”, esta columna contiene 3 botones: la edición del docente, la carga del docente y la generación del informe del docente. Además, existe un botón que permite agregar nuevos docentes.

3.5.3.1 Agregar docente

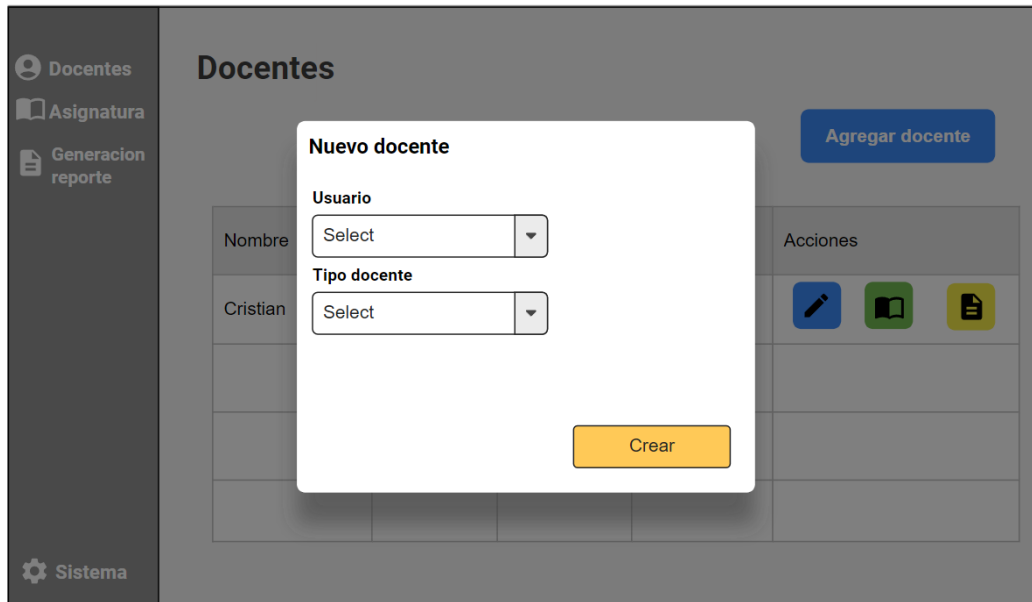


Figura 23: Docentes nuevo docente mockup

Fuente:Elaboracion propia

En la figura número 24 podemos ver el modal que se abre al momento de presionar el botón de agregar docente. El modal nos pide seleccionar un usuario del sistema y seleccionar el tipo de docente que le corresponda.

3.5.3.2 Edición del docente

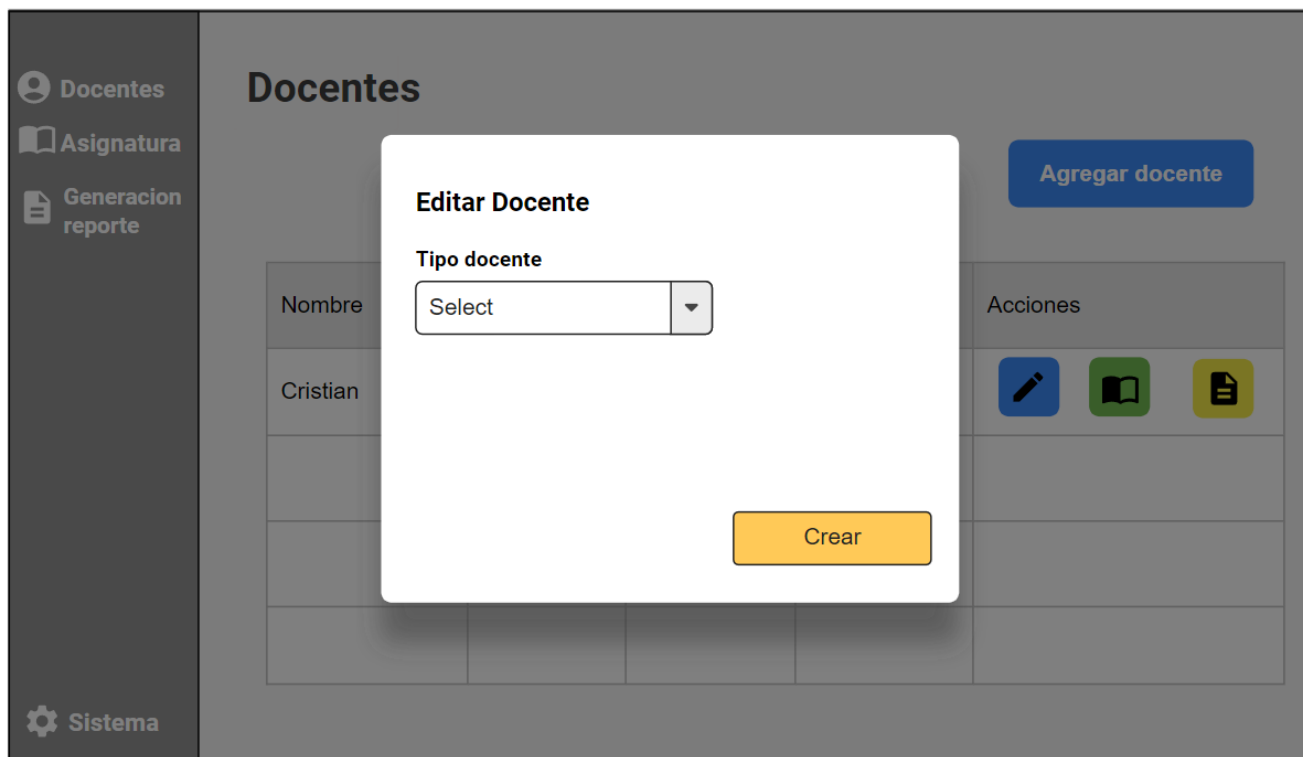


Figura 24: Docentes editar mockup

Fuente:Elaboracion propia

En la figura número 25 podemos ver el modal que se abre al momento de presionar el botón editar en la fila del docente. El modal nos muestra la información que le podemos modificar. A este modal se le puede agregar más información a modificar, pero por términos de alcance de la memoria solo se editará el tipo de docente.

3.5.3.3 Carga del docente

Figura 25: Docentes carga mockup

Fuente:Elaboracion propia

Tal como se ve en la figura número 26 se puede ver el modal desplegado al momento de presionar el botón de carga en el docente. En el modal se puede ver una tabla con la carga que se le a asignado al docente en cuestión. Además de existir un botón para volver y un botón para añadir nueva carga para el docente.

Figura 26: Docentes añadir carga mockup

Fuente:Elaboracion propia

La figura número 27 nos muestra el modal que aparece al momento de presionar añadir en la figura número 26. Este modal nos pedirá los datos necesarios para asignarle una nueva carga al docente.

3.5.3.4 Generación informe del docente

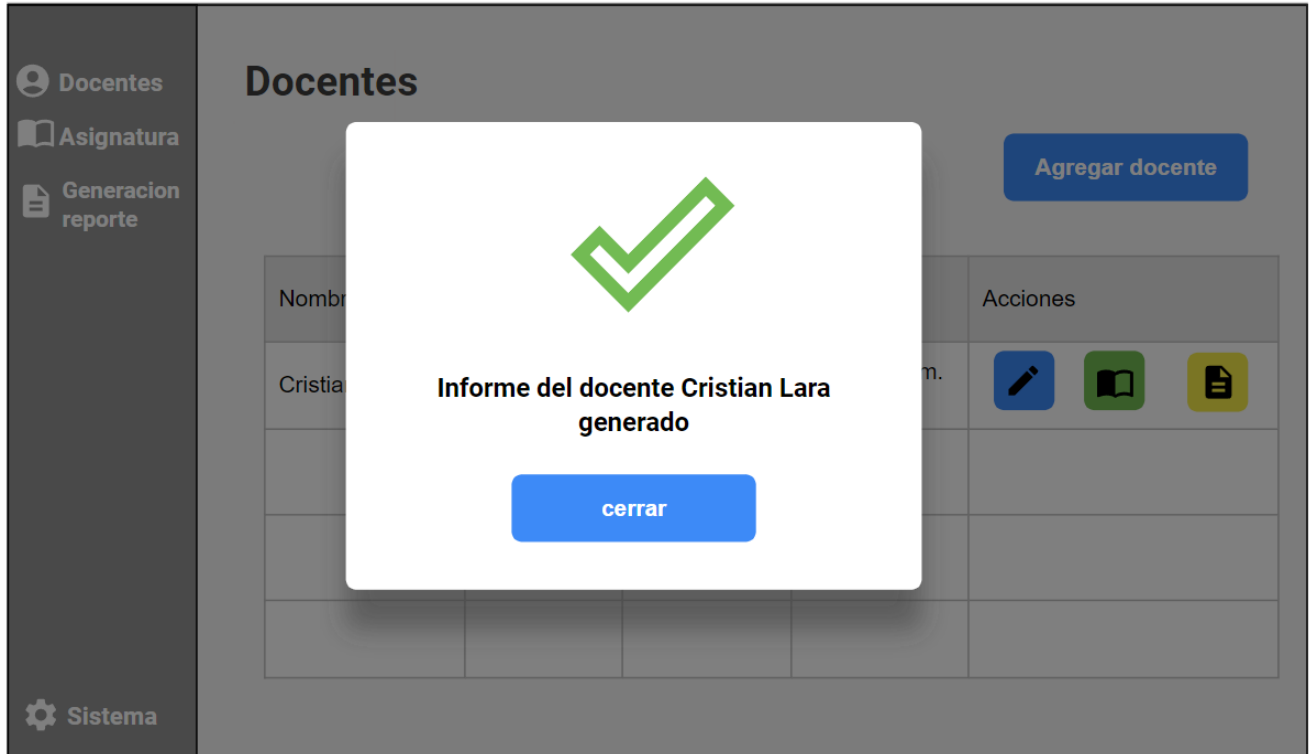


Figura 27: Docentes informe generado mockup
Fuente:Elaboracion propia

En la figura número 28 se puede ver el modal que aparece al momento de presionar el ultimo botón, el botón de generar el informe del docente. Este modal no nos pide nada ya que al momento de presionar el botón el sistema genera el informe del docente y solo nos alerta que este ya se a generado y el navegador lo descargara automáticamente.

3.5.4 Asignaturas

A continuación, se mostrará la propuesta para la opción de asignaturas.

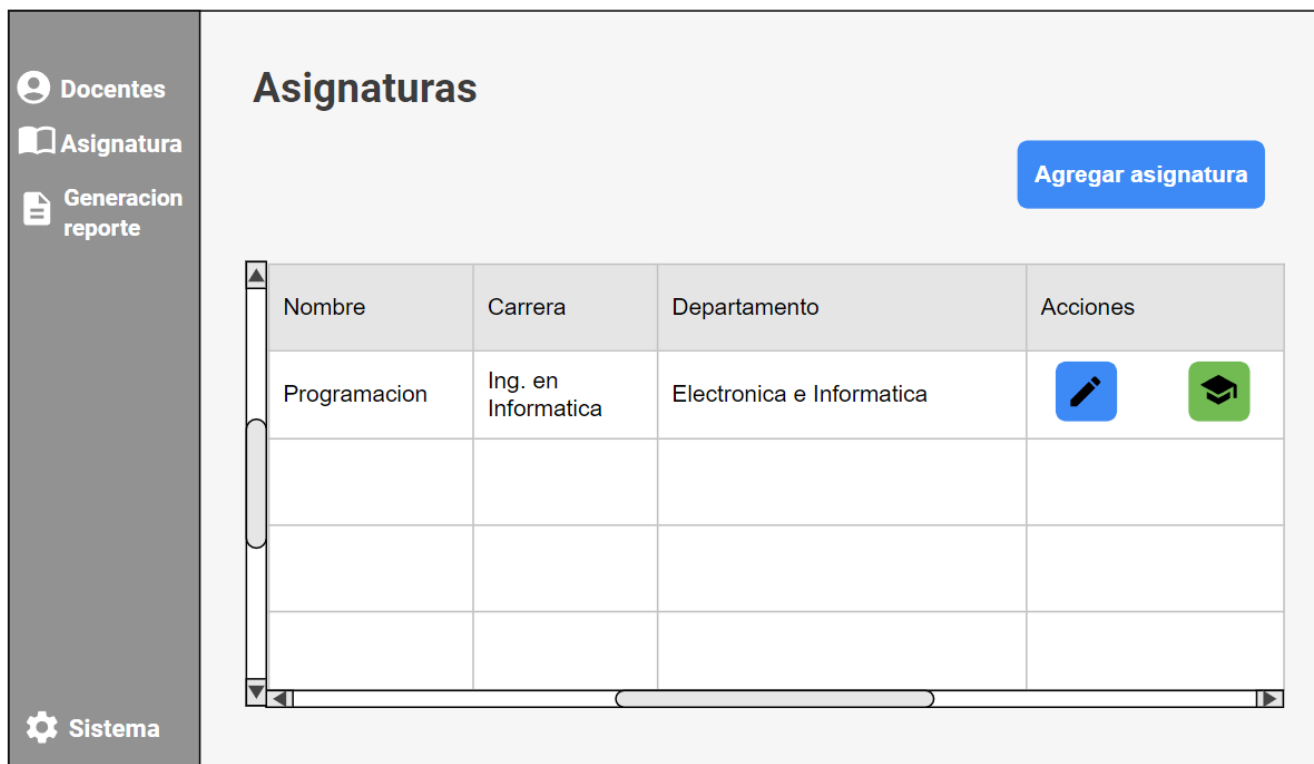


Figura 28: Asignaturas mockup
Fuente:Elaboracion propia

En la figura número 29 podemos ver la vista encargada de mostrar las asignaturas que existen en el sistema. En la vista podemos ver una tabla con las asignaturas registradas en el sistema, esta tabla contiene una columna llamada “acciones”, la cual contiene 2 botones: el botón de editar la asignatura y el botón de los paralelos. Además, en la vista existe un botón para agregar una asignatura al sistema.

3.5.4.1 Agregar asignatura

The mockup shows a sidebar with 'Docentes', 'Asignatura', 'Generacion reporte', and 'Sistema'. The main area is titled 'Asignaturas' and contains a table with columns 'Nombre', 'Programación', and 'Acciones'. A modal titled 'Añadir asignatura' is open, featuring a form with the following fields: 'Nombre' (text input), 'Carrera' (dropdown), 'Codigo usm' (text input), 'semestre' (dropdown), 'Horas catedra' (text input), 'Horas laboratorio' (text input), and 'Horas ayudantia' (text input). An 'Añadir' button is at the bottom right of the modal. A blue 'Agregar asignatura' button is located in the top right corner of the main area.

Figura 29: Asignaturas añadir mockup

Fuente:Elaboracion propia

En la figura número 30 podemos ver el modal que aparece al momento de presionar el botón de agregar asignatura. En el modal nos aparece un formulario el cual nos pide los datos necesarios para el ingreso de una nueva asignatura.

3.5.4.2 Editar asignatura

The mockup is identical to the previous one, but the modal is titled 'Editar asignatura' and features an 'Editar' button at the bottom right. The 'Agregar asignatura' button remains in the top right corner of the main area.

Figura 30: Asignaturas editar mockup

Fuente:Elaboracion propia

En la figura número 31 podemos ver el modal que se despliega al momento de presionar el botón de edición en la asignatura. Este modal tiene los mismos campos que el de agregar ya que en este se mostrara en los campos los datos que la asignatura tenga, para así editarlos en estos mismos campos.

3.5.4.3 Paralelos



Figura 31: Asignaturas paralelos mockup

Fuente:Elaboracion propia

En la figura número 32 se puede ver el modal que se despliega al momento de presionar el botón de los paralelos. En este modal se puede ver una tabla la cual contendrá los paralelos que existan para la asignatura. Además de existir un botón para añadir un nuevo paralelo.

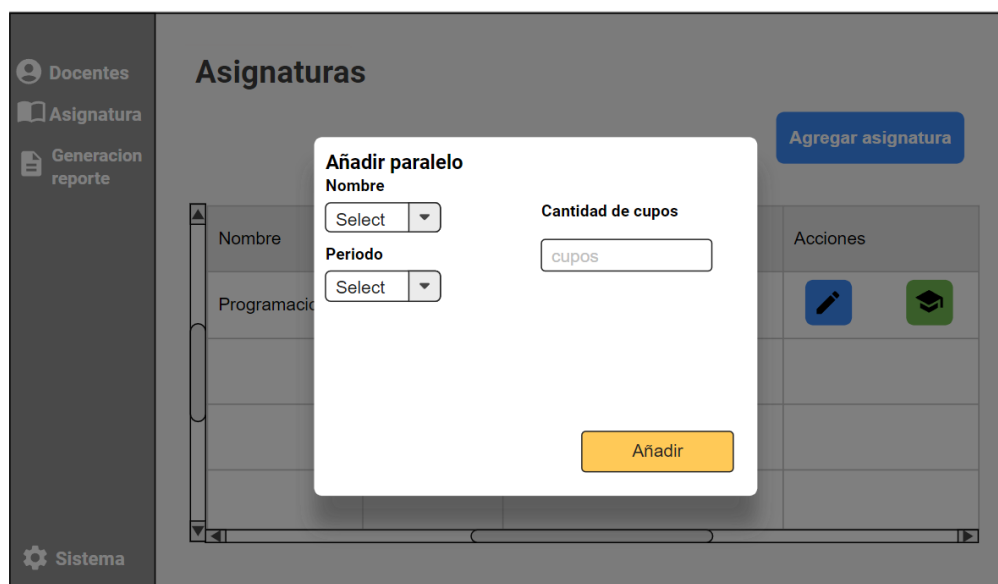


Figura 32: Asignaturas paralelo añadir mockup

Fuente:Elaboracion propia

Tal como se ve en la figura número 33 podemos ver le modal que se despliega al momento de presionar el botón de añadir en la figura número 32. En el modal se puede ver un formulario el cual no pide los datos necesarios para agregar un nuevo paralelo para la asignatura.

3.5.5 Generación reporte

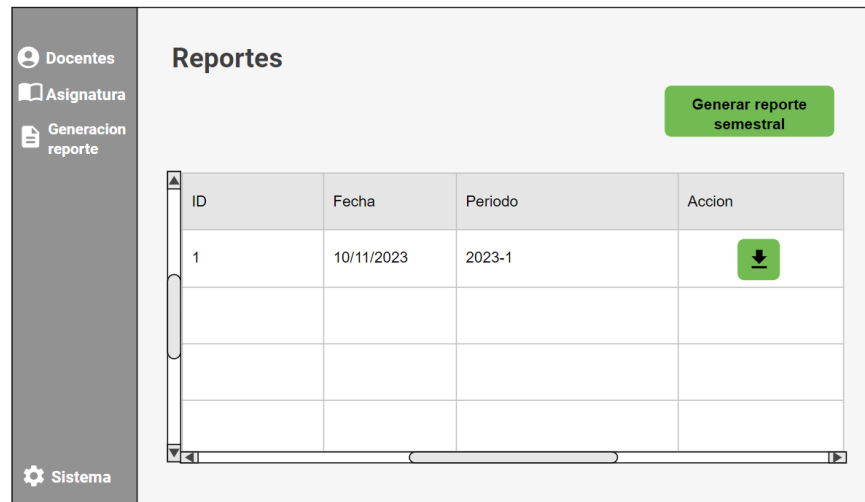


Figura 33: Reportes mockup
Fuente:Elaboracion propia

En la figura número 34 podemos ver la vista para la generación de los reportes semestrales. En esta vista podemos ver una tabla que mostrara los archivos previamente generados, en la misma tabla se encuentra un botón en la columna “accion”, el cual es el de la descarga del reporte generado. Además, en la vista está el botón de generar el reporte. Cabe recalcar que el almacenamiento de estos reportes no está contemplado en el modelo de datos, pero es un punto que puede ser agregado al momento de implementar la solución para así tener un mejor orden de los archivos de los reportes.

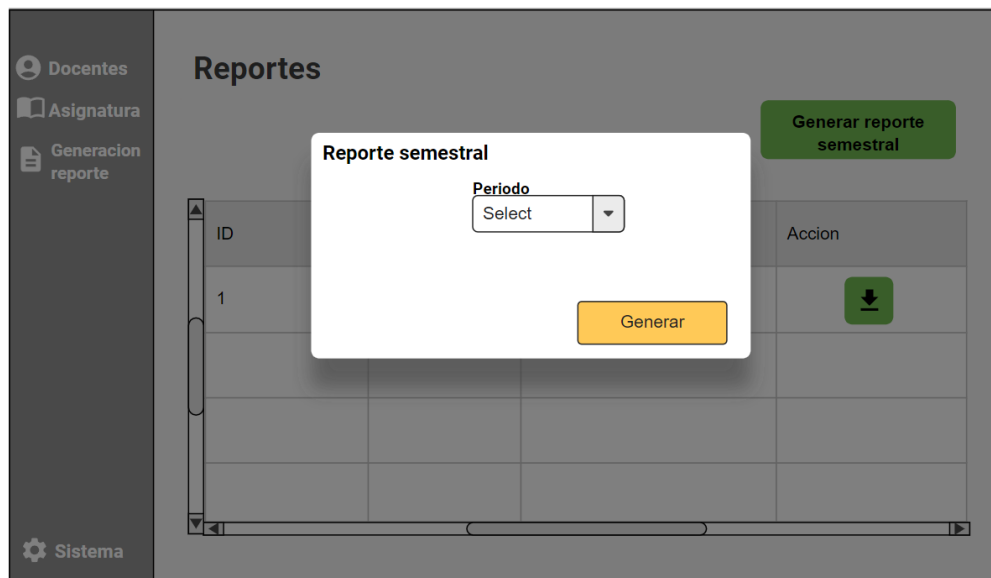


Figura 34: Reportes generar modal mockup
Fuente:Elaboracion propia

En la figura número 35 se puede ver un modal, el cual se despliega al momento de presionar el botón de generar reporte semestral. En este modal se solicita el periodo para así generar el informe con respecto a ese periodo.

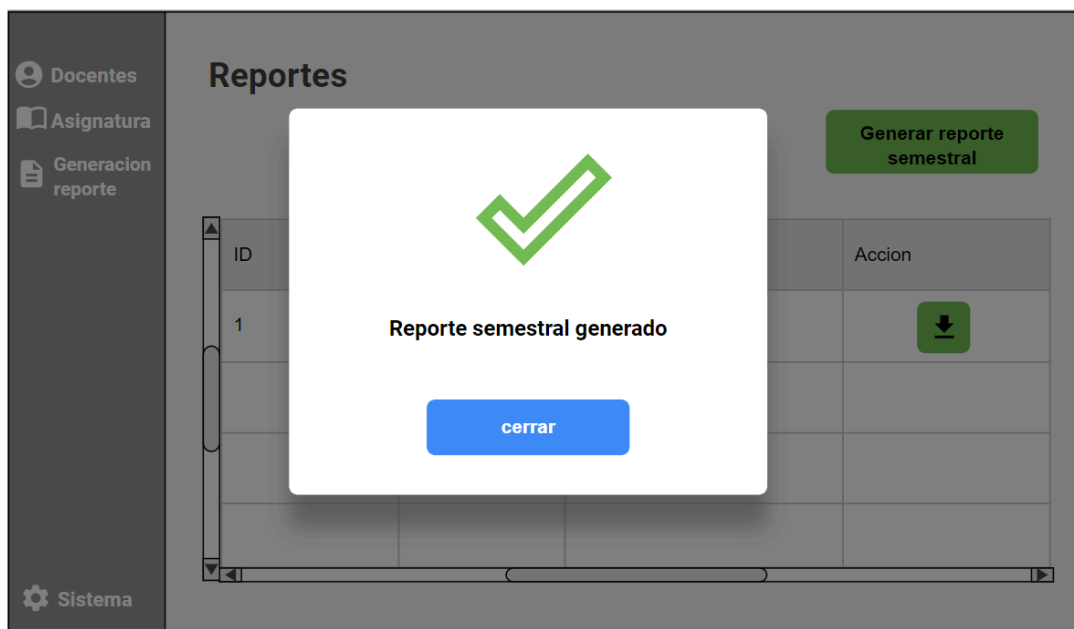


Figura 35: Reportes generado mockup

Fuente:Elaboracion propia

Una vez presionado el botón de generar en el modal de la Figura 36 se despliega este modal, el cual nos dice que el reporte a sido generado. Solo queda presionar el botón de cerrar y presionar el botón de descargar en el reporte que solicite el usuario.

3.5.6 Sistema



Figura 36: Sistema mockup

Fuente:Elaboracion propia

En la última opción tal como se puede ver en la figura número 37, la vista no contiene alguna tabla ya que, hay 2 nuevas opciones. La opción de ver los usuarios y el de iniciar un nuevo periodo.

3.5.6.1 Usuarios

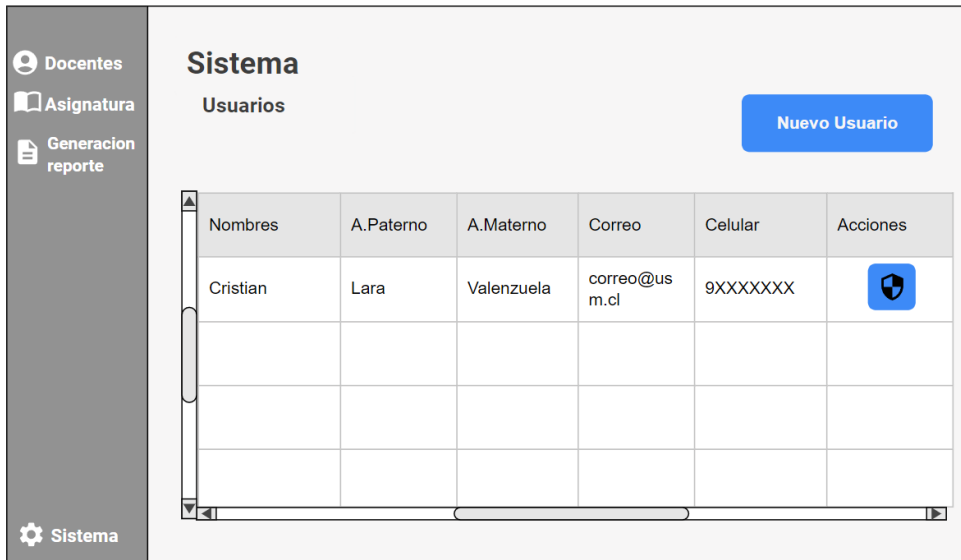


Figura 37: Sistema usuarios mockup

Fuente:Elaboracion propia

En la figura número 38 podemos ver la vista de los usuarios al momento de presionar el botón usuario en la vista anterior. Esta vista contiene una tabla con todos los usuarios ingresados en el sistema, la tabla contiene los datos de cada usuario y una columna “acciones”, esta columna contiene un botón el cual es para darle los permisos de jefe de carrera al usuario seleccionado. Además, en la vista hay un botón para registrar nuevos usuarios en el sistema.

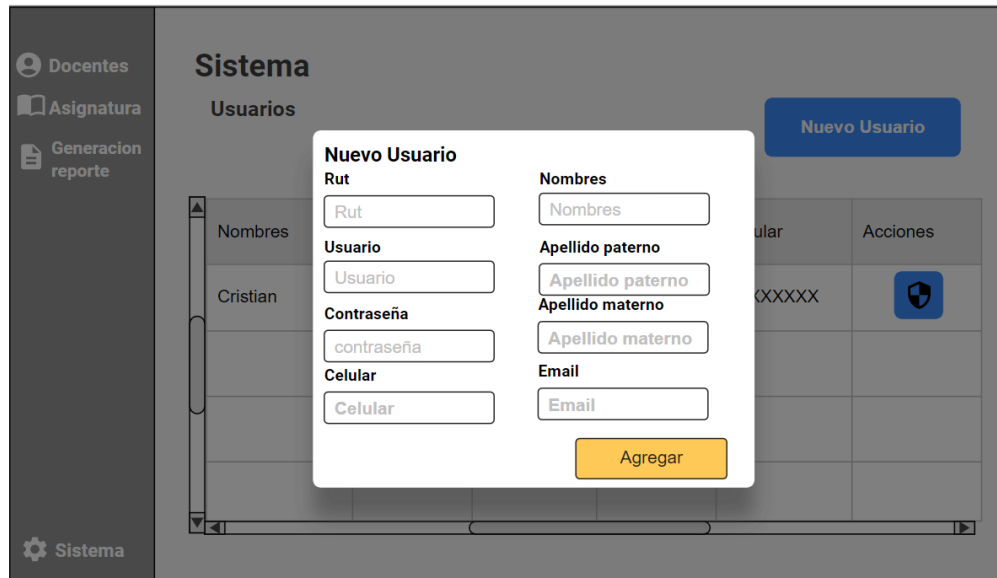


Figura 38: Sistema usuarios nuevo mockup

Fuente:Elaboracion propia

En la figura número 39 podemos ver un modal que es desplegado al momento de presionar el botón de nuevo usuario. En este modal se puede ver un formulario, el cual nos pide los datos necesarios para registrar un nuevo usuario.

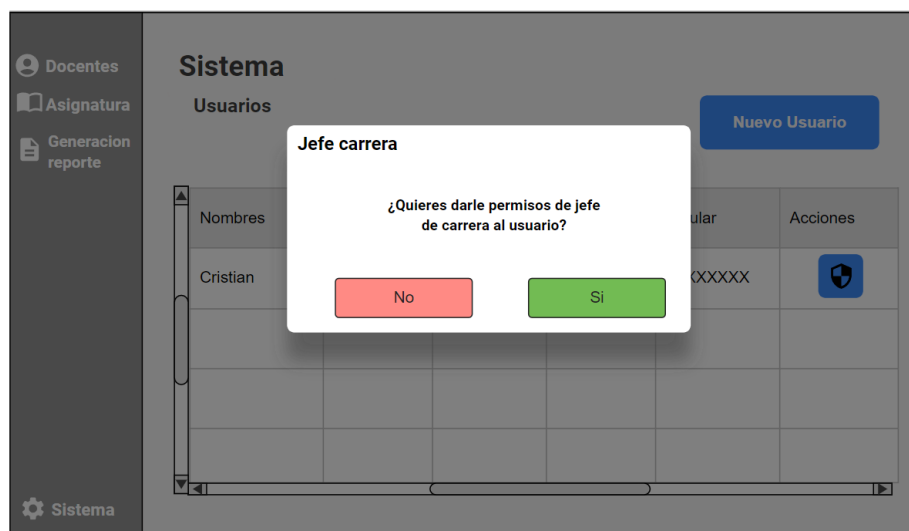


Figura 39: Sistema usuarios permisos mockup

Fuente:Elaboracion propia

En la figura número 40 podemos ver un modal que es desplegado al momento de presionar el botón de dar permiso de jefe de carrera en la vista de usuarios. Este modal nos pregunta si queremos darle estos permisos al usuario seleccionado dando a elegir entre un “sí” y un “no”.

Cabe recalcar que la opción de nuevo usuario y la de dar permisos solo tendrán acceso los administradores del sistema o los que tengan los permisos.

3.5.6.2 Periodo

En este punto volvemos a la vista de sistema y presionamos el botón de Periodo.

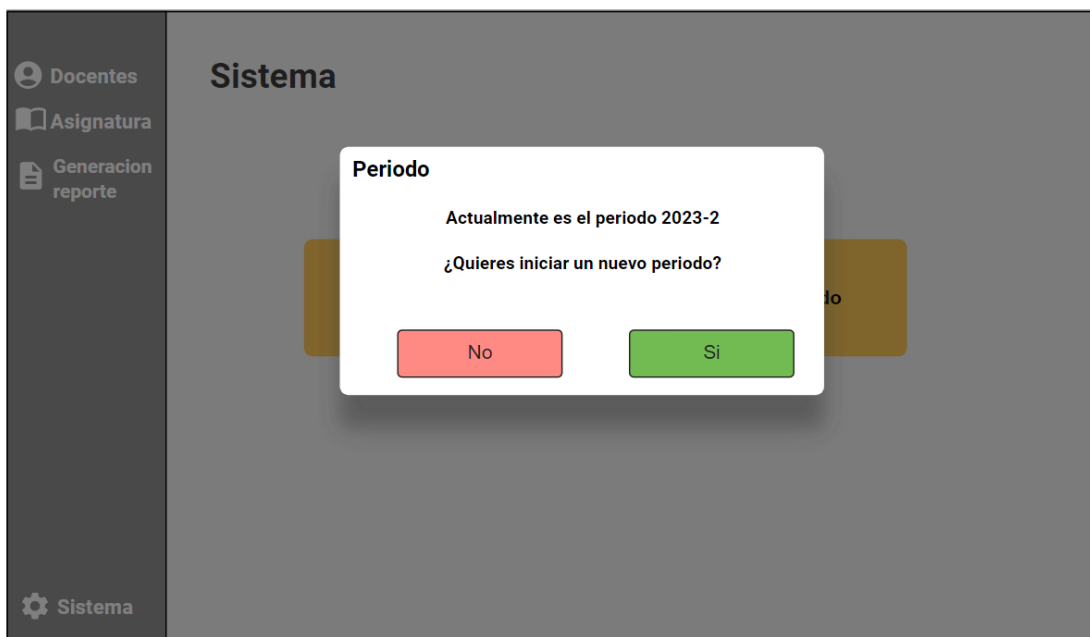


Figura 40: Periodo alerta nuevo

Fuente:Elaboracion propia

Tal cual como se ve en la figura número 41 se puede ver un modal el cual nos dice el periodo en el que estamos ahora y si queremos iniciar un nuevo periodo. Dándonos como respuesta un “sí” y “no”. Al presionar el botón “no” este nos devuelve a la vista anterior.

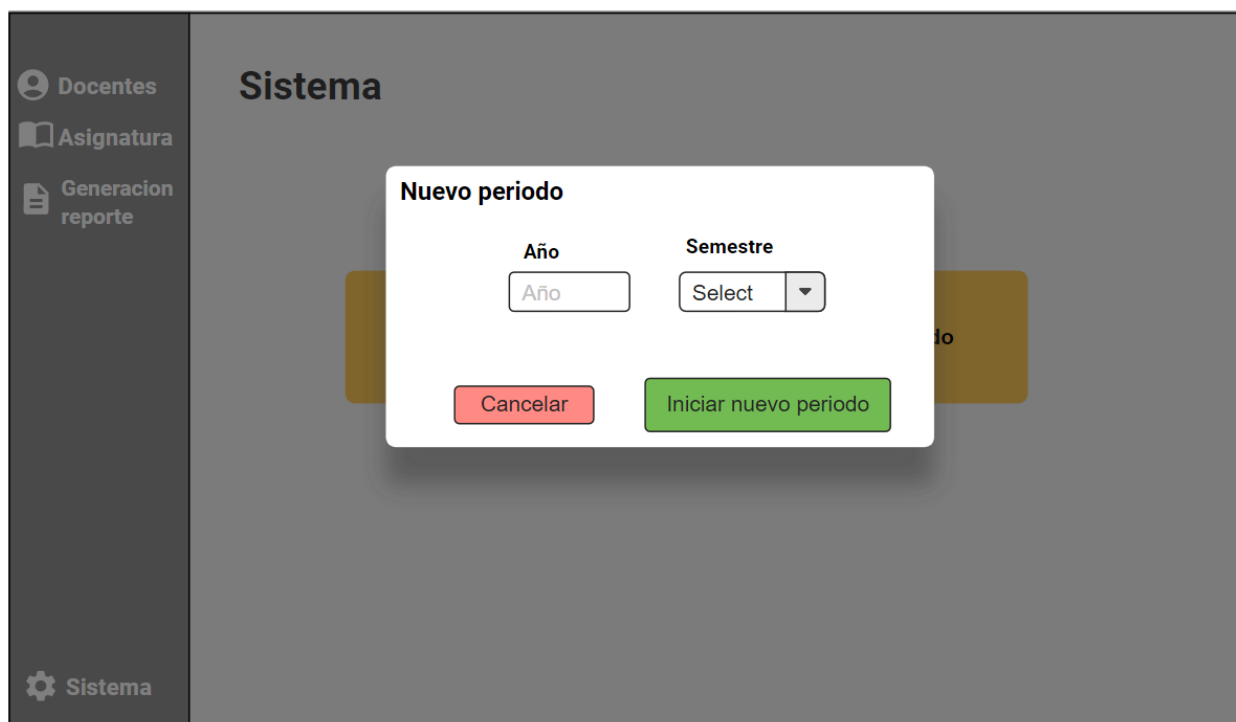


Figura 41: Sistema periodo iniciar

Fuente:Elaboracion propia

Al momento de presionar el botón “si” nos despliega un modal el cual nos pregunta el año del nuevo periodo y el semestre correspondiente.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

En este punto de la memoria se demostrará la validez de la solución propuesta mediante el análisis de distintos puntos de esta solución. Detallando los distintos tipos de validaciones utilizadas para validar la solución. Para medir esta la efectividad se utilizarán métricas en las validaciones que sea relevante

A continuación, se detallarán 3 validaciones.

4.1 Validaciones.

Las pruebas de validación seleccionadas para la validación de esta solución fueron seleccionadas, según lo desarrollado durante esta memoria.

4.1.1 Validación de interfaz de usuario.

La validación de la interfaz de usuario corresponde a un análisis que dará como respuesta si logra el objetivo o no, que tomará en cuenta la información en pantalla que mostrará el sistema al momento de ser utilizado, además de la forma, colores y cercanía con el usuario.

A continuación, tomaremos 2 figuras anteriormente mostradas en esta memoria. Analizaremos la figura 22 y la figura 29

- Figura 22

En esta figura se puede ver una interfaz de usuario simple, mostrando los botones los cuales acceden a las funcionalidades de manera llamativa y simple. Además de centrar el enfoque de la vista en las funcionalidades.

- Figura 29

En esta figura podemos ver el interior de la funcionalidad de asignaturas, por lo que vemos otra forma de visualizar el sistema, ya que, se puede ver una barra lateral la cual ayuda al usuario a visualizar el resto de las funciones y mantener un buen flujo dentro del sistema. Dentro de asignaturas podemos visualizar una tabla la cual contiene los datos de cada asignatura que este ingresada en el sistema. Para cada asignatura existen botones individuales, los cuales tienen un icono representativo de su función y un color llamativo para así tener una amigable experiencia.

Con el anterior análisis de las figuras podemos concluir que las interfaces propuestas para el desarrollo del sistema tienen la capacidad para ser amigables, intuitivas y llamativas para los usuarios del sistema.

4.1.2 Pruebas de modelo de datos.

En esta prueba se analizará el correcto funcionamiento del modelo de datos con una petición hacia una tabla en específico del modelo de datos.

La tabla seleccionada es la tabla de asignatura, la cual tiene gran relevancia dentro del sistema. A continuación, la petición ingresada para la prueba de validación.

```
INSERT INTO
ASIGNATURA (ASIG_ID,CAR_ID,DEPTO_ID,ASIG_COD_USM,ASIG_NOMBRE,ASIG_HR_CA
TEDRA,ASIG_HR_LAB,ASIG_HR_AYUDANTIA,ASIG_SEM) VALUES (
'2',
'3',
'5',
'EIN413-B',
'PROGRAMACION',
'2',
'0',
'0',
'2'
); SELECT * FROM ASIGNATURA;
```

Esta petición tiene como objetivo ingresar una nueva asignatura al sistema con sus respectivos datos.

En la siguiente figura podemos ver como la petición fue ejecutada correctamente y en la próxima figura se puede ver la tabla con la asignatura ingresada en el sistema.

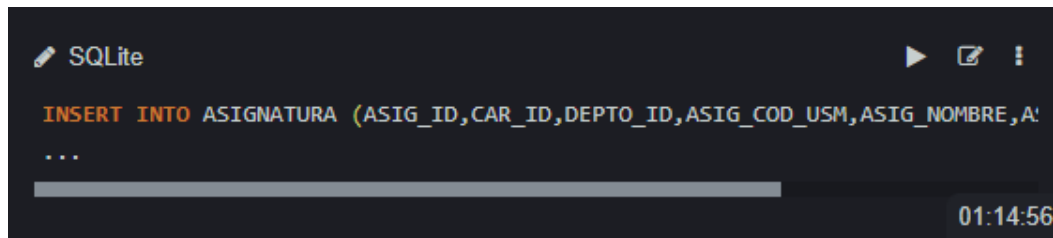


Figura 42: Imagen petición tabla asignatura

Fuente:Elaboracion propia

```

1 INSERT INTO ASIGNATURA (ASIG_ID,CAR_ID,DEPTO_ID,ASIG_COD_USM,ASIG_NOMBRE,ASIG_HR_CATEDRA,ASIG_HR_LAB,ASIG_HR_AYUDANTIA,ASIG_SEM) VALUES (
2 '2',
3 '3',
4 '5',
5 'EIN413-B',
6 'PROGRAMACION',
7 '2',
8 '0',
9 '0',
10 '2'
11 ); SELECT * FROM ASIGNATURA;

```

#	ASIG_ID	CAR_ID	DEPTO_ID	ASIG_COD_USM	ASIG_NOMBRE	ASIG_HR_CATE...	ASIG_HR_LAB	ASIG_HR_AYUDANTIA	ASIG_SEM
1	2	3	5	EIN413-B	PROGRAMACION	2	0	0	2
2	2	3	5	EIN413-B	PROGRAMACION	2	0	0	2

Figura 43: Imagen petición tabla asignatura 2
Fuente:Elaboracion propia

Tras realizar la prueba de validación del modelo de datos podemos sacar como conclusión que la tabla tendrá un correcto funcionamiento por lo que la solución es funcional.

4.1.3 Validación de requerimientos.

La validación de requerimientos tomara en cuenta las funcionalidades descritas en el sistema y mostradas en las interfaces propuestas.

Para validar que la propuesta de solución del sistema contiene los requerimientos para un correcto funcionamiento del proceso, analizaremos las interfaces propuestas anteriormente.

Luego de revisar las interfaces propuestas podemos ver que en la propuesta existen 4 funcionalidades, las cuales corresponden a los principales requerimientos del sistema, en primer lugar, la funcionalidad del docente, la cual nos permite agregar y editar docentes al sistema y agregarle la respectiva carga al docente. En segundo lugar, se puede ver la funcionalidad de las asignaturas, en la cual podemos visualizar las asignaturas ingresadas, añadir nuevas y editar ya existentes. En tercer lugar, podemos ver la funcionalidad de los reportes semestrales, los cuales podemos ver y generar respectivamente. Y en ultimo lugar los ajustes del sistema, los cuales corresponden a la gestión de los usuarios del sistema y del periodo académico en el sistema.

A raíz de este breve análisis de las funcionalidades podemos decir que las funcionalidades del sistema tienen contemplado todos los requerimientos para una correcta ejecución del proceso en cuestión.

CAPÍTULO 5: CONCLUSIONES

En síntesis, la propuesta de instaurar un sistema que digitalice y gestione la información de carga académica mediante sistema informático se revela como una mejora fundamental para la Universidad. Al cumplir con los objetivos establecidos inicialmente, se ha logrado diseñar una plataforma eficiente y amigable, optimizando el proceso de gestión de la carga de los docentes y al momento de la generación de reportes semestrales. Este enfoque no solo facilita la tarea de los encargados del proceso, sino que también sienta las bases para futuras expansiones y conexiones entre sedes, mejorando la eficacia y rapidez en procesos administrativos y pagos.

Desde el inicio de esta memoria, hemos delineado nuestros objetivos, y al llegar a este punto, podemos afirmar con confianza que estos fueron cumplidos a lo largo del proceso. El objetivo primordial de diseñar y analizar una plataforma ha sido alcanzado exitosamente mediante la solución propuesta en esta memoria. La plataforma diseñada no solo cumple con la función de ser eficiente y amigable, sino que también optimiza el tiempo de aprendizaje para los usuarios, especialmente aquellos que se enfrentan a un proceso tedioso cuando se realiza de manera manual, rellenando una única tabla en un archivo. Esto resulta particularmente beneficioso para usuarios inexpertos o nuevos en este proceso, mejorando significativamente su experiencia.

Cabe destacar que el diseño presentado no solo es útil al final de cada semestre para la generación de reportes, sino que también puede ser empleado para obtener información sobre la carga que cada docente tiene en la universidad. Este aspecto adquiere relevancia al gestionar los pagos hacia cada docente, ya que la solicitud de esta información es común en tales procesos.

Aunque el alcance de esta memoria no incluyó el desarrollo completo del sistema informático propuesto, se ha llevado a cabo el diseño y análisis integral. Esto abarcó desde la identificación de procesos, casos de uso, arquitectura propuesta para el desarrollo, modelo de datos, hasta el diseño de la interfaz del usuario. Es importante señalar que los límites de esta memoria se centraron en las necesidades específicas de la sede Concepción de la UTFSM. Sin embargo, se reconoce que, si otras sedes de la UTFSM presentan diferentes necesidades, el sistema puede expandirse para abordar proyectos más extensos. Considerando que la UTFSM es una universidad con múltiples sedes, existe la posibilidad de una conexión futura mediante este sistema, optimizando así los procesos de gestión y reduciendo la dependencia de correos electrónicos y revisiones manuales de archivos. Con un posible crecimiento del sistema, los pagos a los docentes podrían agilizarse considerablemente. En este sentido, la propuesta no solo responde a las necesidades actuales, sino que también se proyecta como una solución escalable y adaptable para futuros desarrollos en el contexto de la Universidad.

Referencias bibliográficas

Digital Ocean. (29 de Noviembre de 2023). Obtenido de

<https://docs.digitalocean.com/products/databases/mysql/>

Django Software Foundation. (s.f.). *Django: The Web framework for perfectionists with deadlines.*

Obtenido de <https://www.djangoproject.com/>

Mora, S. L. (2012). *Universidad de Alicante*. Obtenido de

<https://desarrolloweb.dlsi.ua.es/cursos/2011/html5-css3-es/conceptos-basicos-css>

Pantoja, E. B. (Diciembre de 2004). *Scielo*. Obtenido de

http://www.scielo.org.bo/scielo.php?pid=S1683-07892004000100005&script=sci_arttext