

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA ELÉCTRICA Y  
ELECTRÓNICA**

**DESARROLLO DE UN PROTOTIPO DE SISTEMA WEB  
INFORMATIVO PARA CÁLCULO DE MATRÍCULA Y  
ARANCELES**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN INGENIERÍA EN ELECTRÓNICA Y REDES DE INFORMACIÓN**

**JEFFERSON GUILLERMO CARGUA MEDINA**

jeffersoncarga@gmail.com

**DIRECTOR: ING. RAÚL DAVID MEJÍA NAVARRETE, M.Sc.**

david.mejia@epn.edu.ec

**Quito, marzo 2023**

## **AVAL**

Certifico que el presente trabajo fue desarrollado por Jefferson Guillermo Cargua Medina, bajo mi supervisión.



---

**RAÚL DAVID MEJÍA NAVARRETE**  
**DIRECTOR DEL TRABAJO DE TITULACIÓN**

## DECLARACIÓN DE AUTORÍA

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

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.



---

Jefferson Guillermo Cargua Medina

# DEDICATORIA

Dedico este trabajo a:

Mis padres Elvia y Guillermo, quienes, estuvieron día tras día luchando junto a mí, y que, con su esfuerzo y paciencia, me impulsaron a cumplir con una meta importante en mi vida., gracias por darme ánimos de perseguir mis sueños con valentía y perseverancia.

Mis hermanos Juan, Bryan, Cristian y Yadira por estar junto a mí en los momentos más difíciles, por darme palabras de aliento cuando lo necesitaba, por toda la paciencia y el apoyo que me brindaron.

A mi abuelita Luz María, quien fue como una segunda madre, por todo el amor y cariño, por inculcar en mí la nobleza y la humildad, por sus enseñanzas, por todas sus oraciones, por ser la mujer más alegre del mundo siempre te llevaré en mi corazón.

A María, quien me apoyo mucho cuando lo necesitaba, por extender su mano en los momentos difíciles, de verdad mil gracias por todo.

A mi hermoso Chester, quien fue el que me acompañaba en mis desveladas y que con su presencia calmaba mi alma, y que cuando movía su colita, me intentaba decir que todo va a salir bien.

## **AGRADECIMIENTO**

A mi madre Elvia por apoyarme y motivarme a perseguir mis sueños, por enseñarme a enfrentar mis miedos con ímpetu y esfuerzo y que, a pesar de la distancia, siempre ha estado para mí, dándome palabras de aliento y sobretodo su amor.

A mi padre Guillermo por estar presente en todo momento, por ayudarme a crecer y formar en mí una persona leal y solidaria.

A mis amigos de toda la vida Alejandra Flores, Anabel Haro, Jonathan Buñay y Franklin Peña por estar siempre conmigo, apoyándome y dándome su cariño para seguir adelante y cumplir mis sueños.

A mis amigos de la Poli; Miguel Gutiérrez, Alejandro Guerrero, Karol Chileno, Lorena Quimbiulco, Wendy Rengifo, Andrea Burgos y Marialejandra Contreras con quienes pase grandes momentos en la universidad.

Gracias por todo lo compartido, por los días de estudio, por las salidas a comer y los trabajos realizados, ustedes son las mejores personas que pude conocer durante la carrera.

En especial a mi Director de Trabajo de Titulación, el Ingeniero David Mejía, M.Sc., por su apoyo, tiempo, orientación y guía y sobre todo paciencia al dirigir este trabajo.

# ÍNDICE DE CONTENIDO

AVAL .....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN .....	VIII
ABSTRACT .....	IX
1 INTRODUCCIÓN .....	1
1.1 OBJETIVOS .....	1
1.2 ALCANCE .....	2
1.3 MARCO TEÓRICO.....	3
1.3.1 METODOLOGÍA KANBAN .....	4
1.3.2 HISTORIAS DE USUARIO EN METODOLOGÍAS ÁGILES .....	6
1.3.3 PATRÓN MODELO VISTA CONTROLADOR.....	7
1.3.4 ASP.NET .....	9
1.3.5 ENTITY FRAMEWORK .....	10
1.3.6 HERRAMIENTAS DE DESARROLLO WEB .....	11
1.3.7 HERRAMIENTAS DE DESARROLLO PARA EL BACK-END .....	14
1.4 RELACIÓN CON TRABAJOS SIMILARES DEL ÁREA .....	14
2 METODOLOGÍA.....	16
2.1 PLANIFICACIÓN INICIAL .....	16
2.1.1 REQUERIMIENTOS FUNCIONALES DEL SISTEMA.....	16
2.1.2 REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA.....	17
2.1.3 PARÁMETROS BASE DEL SISTEMA.....	18
2.1.4 HISTORIAS DE USUARIO .....	19

2.1.5	BACKLOG DE KANBAN.....	20
2.2	DISEÑO .....	22
2.2.1	ARQUITECTURA DEL SISTEMA.....	22
2.2.2	DIAGRAMAS.....	23
2.2.3	SKETCHES .....	29
2.3	IMPLEMENTACIÓN.....	30
2.3.1	CODIFICACIÓN DE LA BASE DE DATOS.....	30
2.3.2	PROYECTO DE VISUAL STUDIO.....	31
2.3.3	CONEXIÓN DE LA BASE DE DATOS.....	33
2.3.4	CODIFICACIÓN DE LAS CLASES .....	37
2.3.5	CODIFICACIÓN DE LOS CONTROLADORES Y VISTAS .....	40
2.3.6	INICIO DE SESIÓN DEL ADMINISTRADOR .....	42
2.3.7	GESTIÓN DE LA INFORMACIÓN DE LOS PARÁMETROS BASE .....	46
2.3.8	GESTIÓN DE ADMINISTRADORES .....	51
2.3.9	CÁLCULO DE MATRÍCULA Y ARANCELES .....	58
3	RESULTADOS Y DISCUSIÓN .....	65
3.1	PRUEBAS DE FUNCIONAMIENTO.....	65
3.1.1	AUTENTICACIÓN DE USUARIO.....	65
3.1.2	RECUPERACIÓN DE CREDENCIALES POR CORREO.....	66
3.1.3	CREACIÓN DE UN NUEVO ADMINISTRADOR.....	68
3.1.4	ELIMINACIÓN DE UN ADMINISTRADOR .....	71
3.1.5	EDICIÓN DE LOS PARÁMETROS BASE .....	72
3.1.6	CÁLCULO DEL VALOR A PAGAR .....	74
3.2	PRUEBAS DE VALIDACIÓN .....	77
3.3	CORRECCIÓN DE ERRORES .....	80
4	CONCLUSIONES Y RECOMENDACIONES.....	85
4.1	CONCLUSIONES.....	85
4.2	RECOMENDACIONES .....	87

5	REFERENCIAS BIBLIOGRÁFICAS .....	88
	ANEXOS .....	90



# RESUMEN

En este Trabajo de Titulación se presenta el diseño e implementación de un prototipo de sistema web informativo para el cálculo de matrícula y aranceles.

El prototipo permite gestionar la información de los parámetros base para el cálculo de matrícula y aranceles, gestionar usuarios para el mantenimiento de la información de los parámetros base y está en capacidad de calcular y realizar consultas de los valores a cancelar relacionados con la matrícula y aranceles.

Este documento se encuentra organizado en cuatro capítulos. El Capítulo 1 presenta un resumen referente a la teoría relacionada con la metodología Kanban, el *framework* ASP.NET, la tecnología .NET, el patrón Modelo – Vista – Controlador, y se hace un repaso sobre las herramientas de desarrollo web utilizadas. El Capítulo 2 presenta la planificación inicial del prototipo mediante el análisis de requerimientos del sistema, la determinación de los parámetros base del sistema, el levantamiento de historias de usuario y la implementación del tablero Kanban. Luego, se presenta el diseño del prototipo por medio de los diagramas de casos de uso, actividades, secuencia y clases. Así mismo se presenta la estructura de la base de datos relacional y los *sketches* generados. Posterior a esto se presenta un resumen de la implementación del prototipo mediante la codificación de la base de datos y las capas construidas con base en el patrón MVC. El Capítulo 3 presenta las pruebas de funcionamiento, los resultados obtenidos de las pruebas de validación y las correcciones realizadas al prototipo. El Capítulo 4 presenta las conclusiones y recomendaciones generadas durante la realización del presente Trabajo de Titulación. Finalmente, como anexos se incluyen: las historias de usuario, diagramas de actividades, secuencias y los *sketches*.

**PALABRAS CLAVE:** Kanban, ASP.NET, MVC, matriculación, aranceles.

## **ABSTRACT**

In this Final-degree project, the design and implementation of a prototype of an informative web system for the calculation of registration and fees is presented.

The prototype allows managing the information of the base parameters for the calculation of registration and fees, managing users for the maintenance of the information of the base parameters and is capable of calculating and making queries of the values to be canceled related to registration and fees.

This document is organized into four chapters. Chapter 1 presents a summary regarding the theory related to the Kanban methodology, the ASP.NET framework, .NET technology, the Model – View – Controller pattern, and a review of the web development tools used. Chapter 2 presents the initial planning of the prototype through the analysis of system requirements, the determination of the base parameters of the system, the collection of user stories and the implementation of the Kanban board. Then, the prototype design is presented through use case, activity, sequence, and class diagrams. Likewise, the structure of the relational database and the generated sketches are presented. After this, a summary of the implementation of the prototype is presented through the coding of the database and the layers built based on the MVC pattern. Chapter 3 presents the performance tests, the results obtained from the validation tests and the corrections made to the prototype. Chapter 4 presents the conclusions and recommendations generated during the realization of this Degree Project. Finally, as annexes are included: user stories, activity diagrams, sequences and sketches.

**KEYWORDS:** Kanban, ASP.NET, MVC, registration, fees.

# 1 INTRODUCCIÓN

Los estudiantes de la Escuela Politécnica Nacional que han perdido la gratuidad no disponen de mecanismos que les permitan conocer los valores que deben cancelar por la matriculación de acuerdo con la cantidad de horas o créditos que deban tomar en primera, segunda o tercera matrícula, ni según el grupo socioeconómico y de la condición de gratuidad, considerando las disposiciones del «REGLAMENTO DE COBRO DE MATRÍCULAS Y ARANCELES DE LAS CARRERAS DE GRADO, TECNOLOGÍA SUPERIOR Y NIVELACIÓN DE LA ESCUELA POLITÉCNICA NACIONAL» [1].

La información sobre los valores que un estudiante debe cancelar por concepto de matrícula y aranceles se conoce una vez que concluye con su inscripción en el sistema SAEw (Sistema de Administración Estudiantil web) de la Escuela Politécnica Nacional [2].

El no conocer esta información a priori puede provocar que el estudiante o incluso un aspirante no pueda determinar si contará con los recursos para poder matricularse. Algunos intentos se han realizado por parte de los estudiantes para subsanar estos problemas liberando información incompleta de los valores a pagar, lo cual ha generado inconvenientes de facto en la matriculación.

El prototipo propuesto está enfocado en facilitar la entrega de información a los estudiantes de la Escuela Politécnica Nacional o a posibles aspirantes, brindando información sobre el valor que deberán cancelar por concepto de matrícula y aranceles, de acuerdo con algunos parámetros. Estos parámetros incluyen la cantidad de asignaturas que optarán cursar en primera, segunda o tercera matrícula; así como su grupo socioeconómico y la condición de gratuidad en la que se encuentren [1], [3].

El prototipo se construirá mediante un sistema web basado en ASP.NET MVC que permita realizar consultas sobre los valores que los estudiantes deberán cancelar por su matriculación de acuerdo con los datos que proporcionen.

## 1.1 OBJETIVOS

El objetivo general de este Proyecto Integrador es:

- Desarrollar un prototipo de sistema web informativo para el cálculo de matrícula y aranceles.

Los objetivos específicos de este Proyecto Integrador son:

- Estudiar el funcionamiento de las herramientas necesarias para el desarrollo del trabajo de titulación.
- Diseñar los componentes que conforman el sistema web.
- Implementar el sistema web en función del diseño de los componentes.
- Analizar los resultados de las pruebas de funcionamiento del sistema web.

## 1.2 ALCANCE

El Presente Trabajo de Titulación tendrá como finalidad desarrollar un sistema web prototipo que permitirá a los estudiantes de la EPN o posibles aspirantes llevar a cabo consultas sobre los valores que deberán cancelar por su matriculación. El prototipo será desarrollado mediante el *framework* ASP.NET MVC y usando herramientas como: Scaffolding, JQuery, Entity Framework, Bootstrap, Razor, entre otras.

El prototipo empezará mostrando una vista que permitirá escoger el rol que tendrá el usuario. Entre los roles se tendrán los siguientes:

Administrador:

- Podrá visualizar o editar información de los parámetros base para el cálculo de matrícula y aranceles, tanto para carreras de ingeniería como de tecnología.

Estudiante:

- Podrá consultar el valor que deberá cancelar por la matriculación.
- Podrá ingresar información de acuerdo con los requerimientos para el cálculo del valor a cancelar por la matriculación.

El administrador podrá visualizar la información de los parámetros base de acuerdo a la normativa vigente. Además, se tendrá la opción que le permitirá modificar los parámetros base para adecuarse a nuevas disposiciones.

El estudiante podrá ingresar la información de acuerdo con los requerimientos para realizar el cálculo. También, se tendrá la opción que le permitirá calcular el valor a cancelar, con el cual se detallará la información del valor que se deberá cancelar de acuerdo a los parámetros ingresados.

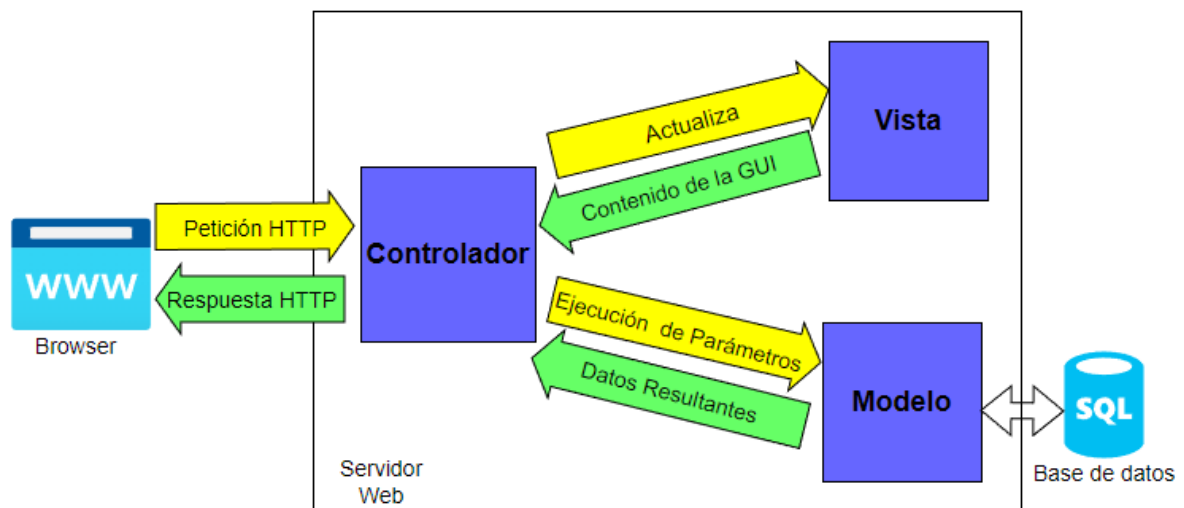
El sistema web permitirá mostrar y actualizar la información de los parámetros base cuando se ingrese como administrador. De manera que el sistema permitirá al administrador consultar o modificar los valores de los parámetros base. Para lo cual, el sistema web

deberá estar conectado a una base de datos que le permitirá obtener o cambiar los valores de los parámetros base almacenados.

El sistema web permitirá a los usuarios que escojan el rol de estudiante calcular los posibles valores que deberán cancelar por su matriculación. Para ello, el sistema web permitirá a los estudiantes o aspirantes ingresar su información de acuerdo con los requerimientos para realizar el cálculo. El sistema web podrá calcular los valores en función de los datos que se proporcionen y de los parámetros base de acuerdo a la normativa. Una vez que el sistema web realice el cálculo, mostrará la información de los valores que deberán cancelar. Para ello, el sistema deberá estar conectado a una base de datos que le permitirá obtener los valores de los parámetros base almacenados para los cálculos.

Para el desarrollo del sistema web propuesto se utilizará como base la metodología ágil Kanban [4]. Finalmente, para confirmar el funcionamiento del sistema web se realizarán pruebas con quince estudiantes de la Escuela Politécnica Nacional.

El esquema del prototipo se presenta en la Figura 1.1.



**Figura 1.1.** Esquema de la arquitectura MVC del prototipo de sistema web

### 1.3 MARCO TEÓRICO

En esta sección se indican los conceptos fundamentales y principios utilizados para el desarrollo del sistema web, que abarca la Metodología Kanban, Historias de usuario, Patrón Modelo-Vista-Controlador, ASP.NET y Entity Framework.

Además, se presenta un breve resumen de las herramientas web utilizadas para el desarrollo del prototipo.

### 1.3.1 METODOLOGÍA KANBAN

Kanban es una metodología de producción desarrollada en Japón, y su traducción en español significa tarjeta visual. Esta metodología fue ideada por Toyota y nace del método *just-in-time* (JIT)<sup>1</sup>, que emplea la utilización de las tarjetas y permite identificar el flujo de trabajo en la producción [4].

Kanban es un sistema visual que permite gestionar proyectos mediante la realización de tareas. Estas tareas se llevan a cabo mediante el flujo de trabajo que se representa a través de un tablero Kanban.

El desarrollo de un proyecto aplicando la metodología Kanban se basa en una serie de principios básicos para obtener el máximo rendimiento del flujo de trabajo. Estos principios básicos son [5]:

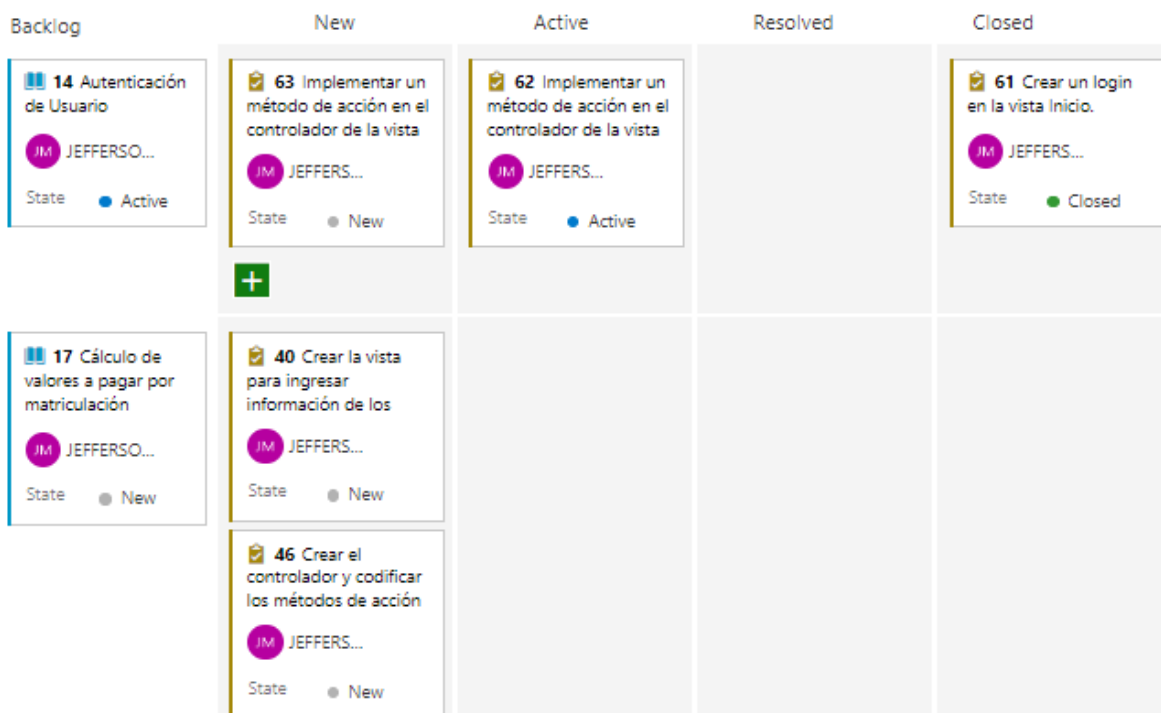
- **Visualizar lo que se hace:** el desarrollador visualiza el flujo de trabajo con todas sus tareas y elementos en el tablero Kanban, con el que se mantendrá al corriente de todo el progreso del desarrollo del proyecto.
- **Limitar la cantidad de trabajo en proceso:** el desarrollador establece objetivos alcanzables para lograr avances importantes dentro del desarrollo del proyecto. Se establece un equilibrio del flujo del trabajo para evitar problemas que impidan el avance del proyecto.
- **Realizar un seguimiento del tiempo:** el desarrollador sigue el tiempo de forma continua del flujo de trabajo y evalúa con precisión el desarrollo del proyecto.
- **Leer de forma fácil los indicadores visuales:** permite conocer lo que está sucediendo con el desarrollo del proyecto de un solo vistazo al tablero Kanban, utilizando prioridades, etiquetas, fechas límites en las tareas, entre otros.
- **Identificar los cuellos de botella y eliminar lo que resulta descartable:** el desarrollador aprovecha al máximo los plazos y ciclos de ejecución del flujo de trabajo para evaluar el rendimiento, detectar problemas y ajustar el flujo de trabajo en consecuencia.

El tablero Kanban es una herramienta para asociar y visualizar el flujo de trabajo del proyecto. El tablero consta de filas y columnas, donde cada columna representa una fase del proceso y las filas representan las diferentes tareas a llevar a cabo para el desarrollo

---

<sup>1</sup> JIT (*Just in time*): es un método de organización de la producción que basa su idea en eliminar los elementos que no generen valor en la producción y a su vez buscar elementos que optimicen la producción.

del proyecto [6]. En la Figura 1.2 se muestra un ejemplo de Tablero Kanban desarrollado en la herramienta Azure DevOps<sup>2</sup>.



**Figura 1.2.** Tablero Kanban en Azure DevOps

Las columnas del tablero Kanban se dividen en varias secciones que permiten mostrar el progreso alcanzado en el desarrollo del proyecto [7], estas son:

- **Bitácora (Backlog):** esta sección consta de los requerimientos de los usuarios para el desarrollo del proyecto.
- **Nuevo:** esta sección consta de tareas definidas por el desarrollador para cumplir con los requerimientos de los usuarios. Estas tareas se encuentran a la espera de que el desarrollador requiera iniciar con alguna tarea que permita el progreso adecuado del proyecto.
- **Activo:** esta sección contiene las tareas que se están desarrollando actualmente. Además, se tendrán todas las actividades que el desarrollador requiera realizar dentro de un plazo establecido y cumplir con el progreso del proyecto.
- **Resuelto:** esta sección consta de las tareas que han sido resueltas por el desarrollador, pero están pendientes de revisión para determinar si se han completado o si necesitan cambios para culminar la misma.

---

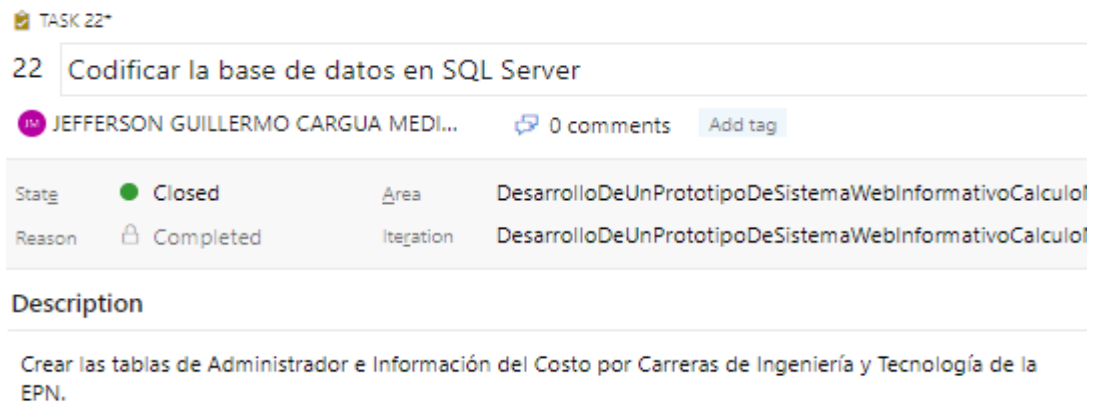
<sup>2</sup> Azure DevOps: es un conjunto de herramientas y servicios que ayudan a administrar los proyectos de desarrollo de software en todas sus fases.

- **Cerrado:** esta sección consta de las tareas que se han completado por el desarrollador. Además, permite que el desarrollador visualice el progreso del proyecto y centre su atención en las demás tareas que se encuentran en las distintas fases del tablero Kanban.

Una tarjeta Kanban es una representación visual de una sola pieza de trabajo, que se coloca dentro del tablero Kanban. La posición de la tarjeta en el tablero representa el estado o fase en la que se encuentra una tarea para el cumplimiento de los requerimientos de los usuarios. Las tarjetas Kanban se usan para [8]:

- Distribuir la información de una manera uniforme y consistente.
- Construir confianza con el desarrollo del proyecto.
- Explicar el proceso del desarrollo del proyecto.

Un ejemplo de tarjeta Kanban desarrollada en Azure DevOps se muestra en la Figura 1.3.



**Figura 1.3.** Tarjeta Kanban en Azure DevOps

### 1.3.2 HISTORIAS DE USUARIO EN METODOLOGÍAS ÁGILES

Una historia de usuario es una explicación breve e informal sobre los requerimientos que debe cumplir el software, y que es escrita desde la perspectiva de un cliente.

Las historias de usuario se caracterizan por ser escritas en pocas líneas y usar un lenguaje sencillo que describe el resultado deseado. En Kanban las historias de usuario se introducen en el *backlog* que se ejecuta siguiendo el flujo de trabajo del tablero Kanban [9].

Para la definir una historia de usuario en Kanban es necesario que la descripción sea corta y simple, contada desde la perspectiva del cliente que desea una nueva funcionalidad del proyecto. Generalmente se sigue una plantilla simple:

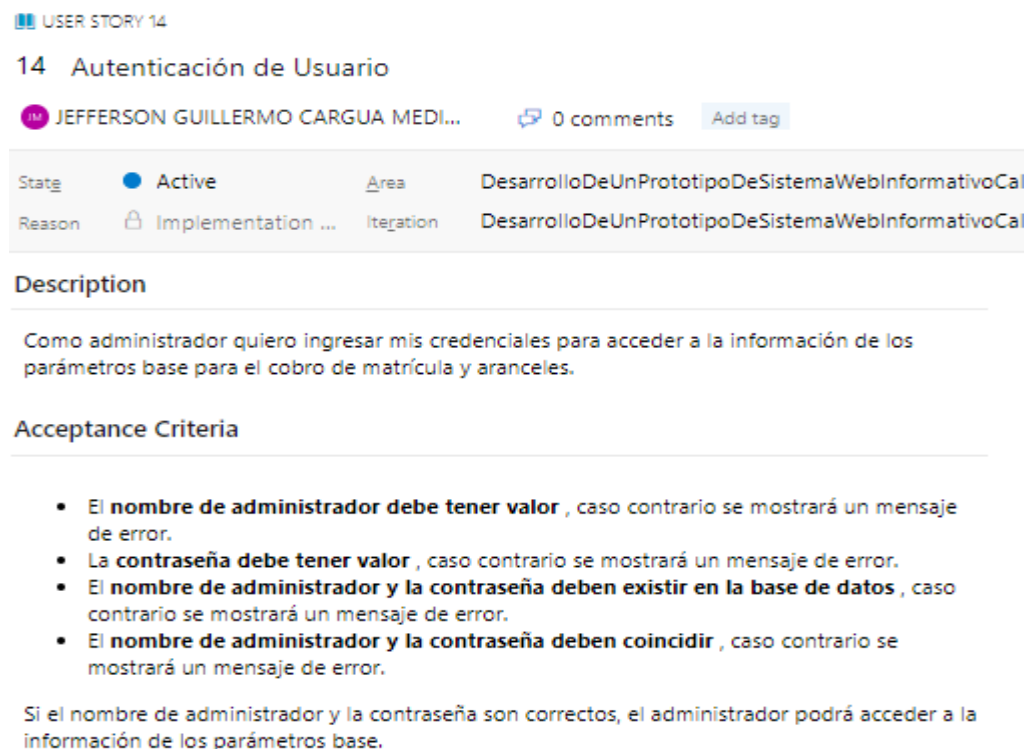
- **Como** <Usuario>
- **Quiero** <Acción>
- **Para que** <Beneficio/Valor>



La estructura de la historia de usuario consta de:

- **Identificador:** valor único que identifica a la historia de usuario.
- **Título:** título que se asigna a la historia de usuario para identificarla.
- **Desarrollador:** nombre de la persona encargada en desarrollar el requerimiento especificado en la historia de usuario.
- **Descripción:** explicación del requerimiento de usuario que se redacta en la historia de usuario.
- **Criterios de aceptación:** condiciones que se ejecutan al completar la implementación de la historia de usuario.

En la Figura 1.4 se presenta un ejemplo de historia de usuario desarrollada en Azure DevOps.



USER STORY 14

### 14 Autenticación de Usuario

JEFFERSON GUILLERMO CARGUA MEDI... 0 comments Add tag

Status	Active	Area	DesarrolloDeUnPrototipoDeSistemaWebInformativoCal
Reason	Implementation ...	Iteration	DesarrolloDeUnPrototipoDeSistemaWebInformativoCal

#### Description

Como administrador quiero ingresar mis credenciales para acceder a la información de los parámetros base para el cobro de matrícula y aranceles.

#### Acceptance Criteria

- El **nombre de administrador debe tener valor** , caso contrario se mostrará un mensaje de error.
- La **contraseña debe tener valor** , caso contrario se mostrará un mensaje de error.
- El **nombre de administrador y la contraseña deben existir en la base de datos** , caso contrario se mostrará un mensaje de error.
- El **nombre de administrador y la contraseña deben coincidir** , caso contrario se mostrará un mensaje de error.

Si el nombre de administrador y la contraseña son correctos, el administrador podrá acceder a la información de los parámetros base.

**Figura 1.4.** Historia de usuario en Azure DevOps

### 1.3.3 PATRÓN MODELO VISTA CONTROLADOR

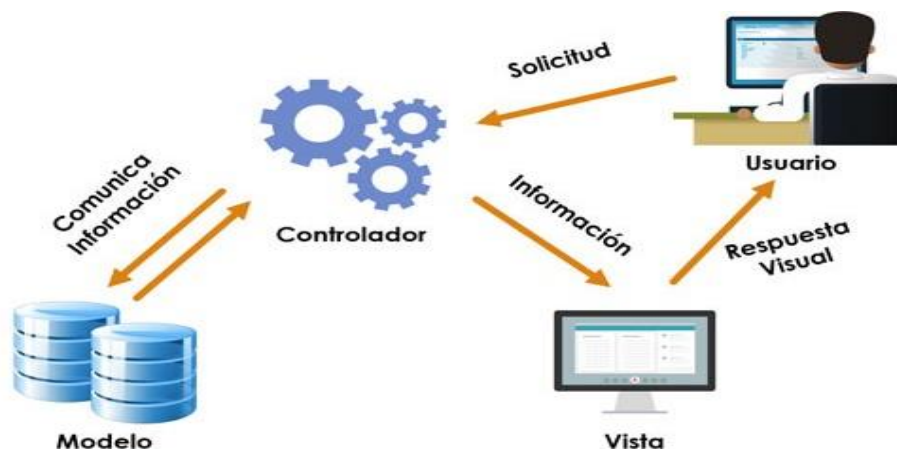
El patrón Modelo Vista Controlador es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio. Este modelo se fundamenta en la separación de tres componentes diferentes: modelo, vista y controlador. Donde, cada componente se define como una representación de lo que va a realizar el software. Por un lado, define los componentes para la representación de la información, y por otro lado define los componentes para la interacción con el usuario [16].

El propósito del patrón MVC se basa en la idea reducir el esfuerzo de programación mediante la reutilización de código y separación de conceptos, para facilitar el desarrollo de aplicaciones, su escalamiento y su mantenimiento.

El patrón MVC es muy utilizado a la hora de implementar aplicaciones web, junto con los principales lenguajes de programación como lo son: Python, C#, Java, entre otros. Esto permite que el patrón MVC se utilice adecuadamente con programación orientada a objetos.

Los componentes del patrón MVC tienen las siguientes responsabilidades [17], las cuales se ilustran en la Figura 1.5.

- **Modelo:** se encarga de gestionar los datos que maneja la aplicación. Su responsabilidad es permitir el acceso y actualización de la información que se encuentra almacenada en una base de datos como SQL<sup>3</sup> Server.
- **Vista:** es la interfaz de usuario, que compone la información que se envía al cliente y los mecanismos de interacción con este. Por lo general, en las vistas se encuentran los códigos escritos en el lenguaje de marcado HTML que permiten mostrar la salida en el navegador web.
- **Controlador:** actúa como intermediario entre el modelo y la vista, gestionando el flujo de información entre ellos y transformándolos para adaptar la información a las necesidades de cada uno. Su responsabilidad es la de responder a acciones que se solicitan desde la aplicación, sea visualizar un elemento, realizar operaciones, buscar información, etc.



**Figura 1.5.** Patrón Modelo – Vista - Controlador (MVC)

---

<sup>3</sup> SQL (*Structured Query Language*): es un lenguaje de programación diseñado para actualizar, obtener, y calcular información en bases de datos relacionales.

### 1.3.4 ASP.NET

ASP.NET es un *framework* o marco de trabajo para programar la web, es gratuito y permite crear sitios dinámicos, aplicaciones web y servicios web modernos mediante el uso de herramientas de desarrollo web como: CSS, Bootstrap, HTML, entre otros. ASP.NET ofrece bibliotecas que permiten generar modelos web comunes como el patrón Modelo-Vista-Controlador (MVC).

ASP.NET permite mezclar el lenguaje de programación C# con HTML para crear páginas limpias y fáciles denominadas páginas de Razor<sup>4</sup>. Razor ofrece una sintaxis para crear sitios web dinámicos, donde se evalúa la expresión escrita en código C# y el contenido resultante representa la salida HTML que se envía al usuario [10].

Para crear sitios y aplicaciones web basados en ASP.NET es necesario el *framework* .NET y un IDE<sup>5</sup> para realizar la codificación.

#### 1.3.4.1 ASP.NET MVC

ASP.NET MVC es un marco de desarrollo web que combina las características del patrón Modelo-Vista-Controlador para construir aplicaciones web. Este *framework* permite compilar de forma eficaz sitios web dinámicos basados en modelos en capas, ya que se separan las capas y se logra un control total para desarrollar de aplicaciones web sofisticadas [15].

Un aspecto importante de ASP.NET MVC es el enfoque que tiene en el concepto de enrutamiento. Esto permite que las aplicaciones acepten peticiones HTTP<sup>6</sup> que van dirigidas a direcciones URL<sup>7</sup> mapeadas en el código fuente. Por tanto, ASP.NET enruta las peticiones a la dirección URL en función al controlador y la vista solicitadas por el usuario. ASP.NET MVC incorpora las funcionalidades de Razor que permite emplear código C# para la programación de las vistas. Donde, es necesario utilizar el símbolo "@" para indicar que se emplea un lenguaje de programación en particular. Razor evalúa estas expresiones

---

<sup>4</sup> Razor: es una sintaxis de programación de ASP.NET utilizada para crear páginas web dinámicas con el lenguaje de programación C#.

<sup>5</sup> IDE (*Integrated Development Environment*): es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

<sup>6</sup> HTTP (*Hypertext Transfer Protocol*): es un protocolo que permite comunicarse con un servidor web para solicitar recursos a través de documentos HTML.

<sup>7</sup> URL (*Uniform Resource Locator*): es una dirección específica que se asigna a cada recurso disponible en Internet para ser localizados a través de un navegador web.

escritas en un lenguaje de programación cuyo resultado transforma en lenguaje de marcado HTML.

### 1.3.5 ENTITY FRAMEWORK

Entity Framework es un conjunto de tecnologías de ADO.NET<sup>8</sup> que respaldan el desarrollo de aplicaciones de software orientadas a datos. Entity Framework permite trabajar con datos en forma de objetos y propiedades específicos del dominio, donde el desarrollador no se preocupe por los datos que se almacenan en la base de datos subyacente. Además, permite que los desarrolladores realicen una mejor abstracción de los datos [13].

La característica principal de Entity Framework es que permite a los desarrolladores que emplean .NET trabajar con una base de datos mediante el uso de objetos creados mediante lenguajes de programación de .NET. Por lo tanto, Entity Framework permite optimizar las líneas de código para acceder a los datos.

Las características de Entity Framework son [14]:

- **Multiplataforma:** Entity Framework puede ejecutarse en Windows, Linux y Mac.
- **Modelado:** Entity Framework crea un Entity Data Model (EDM)<sup>9</sup> basado en *Plain Old CLR Object* (POCO)<sup>10</sup> con propiedades get y set de los diferentes datos para consultar y guardar datos en la base de datos subyacente.
- **Consultable:** Entity Framework permite utilizar consultas mediante *Language Integrated Query* (LINQ)<sup>11</sup> para acceder o actualizar datos de la base de datos. Además, permite ejecutar consultas SQL, sin que estas sean procesadas directamente en la base de datos.

---

<sup>8</sup> ADO.NET: es una arquitectura basada en un conjunto de clases orientadas a objetos que permite a los desarrolladores acceder y manipular de forma eficiente los datos que se conecta a una base de datos.

<sup>9</sup> EDM (*Entity Data Model*): un conjunto de conceptos que describen la estructura de datos, independientemente de su forma almacenada.

<sup>10</sup> POCO (*Plain Old CLR Object*): es un objeto que no hereda de otra clase ni implementa una interfaz.

<sup>11</sup> LINQ (*Language-Integrated Query*): es un conjunto de herramientas que permite llevar a cabo consultas, inserciones, actualizaciones y/o eliminaciones de colecciones de datos de forma rápida y sencilla.

- **Seguimiento a los cambios:** Entity Framework da un seguimiento a los cambios producidos en las instancias de sus entidades que deben enviarse a la base de datos.
- **Simple:** Entity Framework ejecuta los comandos INSERT, UPDATE y DELETE en la base de datos en función de los cambios que se produjeron en las entidades al llamar al método.
- **Es Transaccional:** Entity Framework realiza gestión automática de transacciones mientras consulta o guarda datos.
- **Caché:** Entity Framework permite almacenar datos en cache, por lo que si se realiza consultas repetidas se devolverá los datos que se encuentren almacenados en el cache el lugar de golpear la base de datos.

### 1.3.6 HERRAMIENTAS DE DESARROLLO WEB

Las herramientas de desarrollo web permiten que los desarrolladores trabajen con una gran variedad de tecnologías para la creación de aplicaciones web y se aseguren que se cumplan los objetivos propuestos en el proyecto de forma enfocada. En este sentido las tecnologías para el desarrollo web son recursos que permiten la implementación una aplicación web, cubriendo el diseño de la aplicación que interactúa con los usuarios y las peticiones que se realizan desde la misma.

El desarrollo web puede ser concebido considerando: el *front-end* y el *back-end*. Esta separación ayuda que la aplicación mantenga una comunicación entre ambas partes que permite enviar solicitudes y recibir respuestas.

En la arquitectura MVC las vistas se usan para desarrollar las interfaces de usuario de la aplicación web, es decir; es el *front-end* del sistema, mientras que los controladores se usan para que la parte lógica de la aplicación web funcione adecuadamente; es decir; es el *back-end* del sistema.

**Herramientas de Desarrollo para el Front-End:** Estas herramientas sirven para agilizar y optimizar el trabajo realizado con los lenguajes de programación y de diseño de la interfaz de usuario del sistema. El objetivo de las herramientas de *front-end* es que el diseño permita al usuario interactuar con el sistema. A continuación, se presentan algunas herramientas de desarrollo *front-end* utilizadas en el desarrollo del prototipo.

### 1.3.6.1 HTML5

El lenguaje de marcado de hipertexto versión cinco o HTML5 es la versión más reciente para estructurar el contenido de una aplicación web. Esta versión es compatible con el *framework* ASP.NET MVC y también puede trabajar con lenguajes de programación como C# y desarrollo web como JavaScript.

Un aspecto importante de HTML5 es que permite crear aplicaciones web con contenido multimedia, de forma que elimina la necesidad solicitar recursos a terceros como lo es en el caso del obsoleto Flash Player<sup>12</sup>.

HTML5 ofrece nuevas funcionalidades que mejoran la experiencia de navegación. Los navegadores como: Chrome, Firefox, Explorer, Microsoft Edge, Safari, incluso dispositivos Android o iPhone entienden rápidamente como está estructurada la página web y cuáles son los elementos que la componen, reduciendo así la implementación de *plugins*<sup>13</sup> y aumentando la velocidad de carga [18]. Estos exploradores web interpretan el documento HTML generado y muestran una página web en el que se visualiza el contenido de la interfaz de usuario.

### 1.3.6.2 Javascript

JavaScript es un lenguaje de programación robusto que se puede aplicar a un documento HTML para proporcionar mejor interactividad, operabilidad y dinamismo a las aplicaciones web. JavaScript funciona con HTML y CSS para crear aplicaciones web dinámicas. Además, es compatible con la mayoría de los navegadores web modernos como Google Chrome, Firefox, Microsoft Edge, y también funcionan con navegadores móviles como los disponibles en los sistemas operativos Android y iPhone.

Una de las ventajas de JavaScript es que no necesita de un compilador, ya que los distintos navegadores web interpretan el código directamente.

JavaScript permite realizar operaciones dentro del documento HTML como [19]:

- Declarar variables.
- Almacenar y recuperar valores.
- Definir y buscar funciones.
- Cargar y usar de módulos externos.

---

<sup>12</sup> Flash Player: es una tecnología que se usa para generar contenido multimedia como: animaciones, botones, etc. La cual requiere de la plataforma Adobe Flash Player.

<sup>13</sup> *Plugins*: son complementos que permiten ampliar funciones para mejorar el desempeño de las aplicaciones web y programas de escritorio.

- Escribir controladores de eventos que respondan a ciertos eventos, como por ejemplo al hacer clic.

### 1.3.6.3 Jquery

JQuery es una librería de JavaScript que incluyen varias funciones que facilitan el manejo de documentos HTML, eventos y animaciones; los cuales se utilizan mucho para el desarrollo web. Esta librería es de código abierto, lo que simplifica la programación en JavaScript y permite agregar interactividad y dinamismo a una página web [20].

### 1.3.6.4 CSS3

CSS3 es un lenguaje de diseño que permite definir y crear el aspecto visual de un documento escrito en lenguaje HTML de forma avanzada.

La función principal de CSS es separar el contenido del documento HTML de su forma de presentación. Mediante esta separación se busca mejorar la accesibilidad del documento, así como controlar las características de diseño. Además, permite compartir la misma hoja de estilos con varios documentos HTML, evitando la repetición de código en la estructura del documento.

Entre las características de CSS3 están [21]:

- Permite agregar efectos de animación en los elementos del documento HTML, sin emplear JavaScript, Plugins o Flash Player.
- Es compatible con navegadores como: Google Chrome, Firefox, Internet Explorer, Microsoft Edge, entre otros.
- Mejora el tiempo de respuesta de la página web.
- Permite maximizar la interacción del usuario con los dispositivos móviles, ya que cuenta con diseños adaptativos.

### 1.3.6.5 Bootstrap

Bootstrap es un *framework* utilizado para diseñar aplicaciones web. Esta herramienta incluye plantillas necesarias para el diseño, botones, formularios, menús, *widgets*<sup>14</sup>, imágenes, etiquetas y todo tipo de características [22].

---

<sup>14</sup> *Widget*: es una pequeña aplicación o programa para facilitar el acceso a las funciones más usadas de un dispositivo.

El principal objetivo de Bootstrap es generar aplicaciones web de tipo *responsive*, lo cual permite que puedan funcionar en equipos de escritorio, portátiles, *smartphones*, *Smart TV* y *tables* de forma simple y automática.

### **1.3.7 HERRAMIENTAS DE DESARROLLO PARA EL BACK-END**

Estas herramientas sirven para diseñar la lógica del sistema y dar las soluciones a todas las acciones solicitadas desde el *front-end*, para que sean ejecutadas de forma correcta. Por lo que el desarrollador debe asegurarse de que todo funcione de acuerdo a las interacciones que hace el usuario.

Las funciones principales que desempeña un desarrollador en el *back-end* del sistema son [23]:

- Gestionar el desarrollo de las funciones que simplifican el proceso de desarrollo.
- Acciones de lógica.
- Uso de librerías del servidor.
- Conexiones con la base de datos.

Para el desarrollo de la lógica de una aplicación existen varios lenguajes de programación que permiten el desarrollo web de una aplicación.

## **1.4 RELACIÓN CON TRABAJOS SIMILARES DEL ÁREA**

En este apartado se presenta una comparativa de trabajos similares desarrollados en la carrera de Ingeniería en Electrónica y Redes de Información con este proyecto integrador. En [24] se implementó una aplicación web responsiva para el procesamiento de información de las observaciones meteorológicas. El objetivo principal del proyecto integrador fue el de proporcionar información de las observaciones meteorológicas al público, con la cual se pueda analizar y conocer la variabilidad climática en la que se encuentra el mundo.

La diferencia con este proyecto integrador es que, este proyecto integrador se enfoca en proporcionar información sobre los valores que se deberán cancelar por la matriculación de un estudiante, mientras que el proyecto integrador en comparación se enfoca en proporcionar información sobre las mediciones meteorológicas para determinar las consecuencias del cambio climático.

En [25] se implementó un prototipo de sistema web que realiza la emisión de comprobantes electrónicos de las sucursales de la ferretería “MADERS HERRAJES”. En el proyecto se planteó una alternativa al modelo tradicional de facturación haciendo uso de los medios



digitales para la emisión y almacenamiento de comprobantes electrónicos, los cuales a su vez serían enviados a sus clientes por las transacciones realizadas.

La diferencia con este proyecto integrador está en la utilización de la metodología de desarrollo del prototipo, en este proyecto integrador se utilizará la metodología Kanban, mientras que el proyecto integrador en comparación utiliza la metodología *Rapid Application Development* (RAD)<sup>15</sup>.

---

<sup>15</sup> RAD (*Rapid Application Development*): es una metodología de desarrollo de aplicaciones que se basa en la creación de prototipos de forma veloz por medio de iteraciones frecuentes y comentarios de los clientes.

## **2 METODOLOGÍA**

En este capítulo se describe el proceso utilizado para diseñar e implementar el prototipo. En particular se empleará la investigación aplicada para desarrollar el prototipo de sistema web informativo para el cálculo de matrícula y aranceles de la EPN.

Para el levantamiento de requerimientos se realizará el análisis del «Reglamento para el cobro de matrícula y aranceles de la EPN», con lo cual se realizará la determinación de historias de usuario, luego se definirán los requerimientos funcionales y no funcionales. Así mismo, se determinarán los parámetros base para el cálculo del valor a cancelar por concepto de matrícula y aranceles.

Empleando la metodología Kanban, se generará el *backlog* y las tareas que se requieran para el desarrollo del prototipo. Así mismo, se elaborará un tablero Kanban para dar seguimiento al progreso del prototipo.

Para el diseño del prototipo se elaborarán: diagrama de la arquitectura del sistema, diagrama de casos de uso, diagrama de actividades, diagrama de secuencia y diagrama de clases. Así mismo, se elaborará el esquema relacional de la base de datos y *sketches* para las vistas del sistema web.

Como parte de la implementación del prototipo se realizará la instalación de las herramientas y librerías necesarias para el desarrollo web. Así mismo, se realizará la codificación de la base de datos y la codificación de las capas MVC del prototipo.

Al finalizar con la implementación del prototipo, se realizarán pruebas de validación del funcionamiento del prototipo a quince estudiantes de la EPN. También, se realizará una encuesta para evaluar el cumplimiento de las funcionalidades del prototipo. De ser necesario se realizarán las debidas correcciones basándose en los resultados recibidos de las pruebas.

### **2.1 PLANIFICACIÓN INICIAL**

Como parte de la planificación inicial se determinaron los requerimientos funcionales y no funcionales del sistema y los parámetros base del sistema para el cálculo de matrícula y aranceles. También, se levantaron las historias de usuario y se elaboró el Backlog de Kanban.

#### **2.1.1 REQUERIMIENTOS FUNCIONALES DEL SISTEMA**

Para determinar los requerimientos funcionales del presente prototipo se realizó el análisis del «Reglamento para el cálculo de matrícula y aranceles de la Escuela Politécnica

Nacional» y se levantaron historias de usuario. Con base en lo anterior se obtuvieron los siguientes requerimientos:

El prototipo contará con tres tipos de usuario: administrador principal, administrador secundario, usuario.

El prototipo permitirá al usuario administrador principal:

- Crear y eliminar administradores secundarios.

El prototipo permitirá a los usuarios administrador principal y secundario:

- Acceder a la información de los parámetros base de las carreras de nivel de grado como de nivel técnico-tecnológico. Por simplicidad a lo largo del documento se referirá a las carreras de grado como carreras de Ingeniería y a las carreras del nivel técnico-tecnológico como carreras de Tecnología
- Recuperar las credenciales de acceso.
- Consultar la información de los parámetros base de las carreras tanto del nivel de grado como del nivel técnico-tecnológico.
- Modificar la información de los parámetros base de las carreras tanto del nivel de grado como del nivel técnico-tecnológico.

El sistema permitirá al usuario (estudiante de la EPN o aspirante):

- Consultar los valores a cancelar por concepto de matrícula y aranceles.
- Ingresar información para realizar los cálculos.
- Calcular el valor total a cancelar.
- Visualizar el detalle de los cálculos y el valor a cancelar.

Además, el sistema permitirá:

- Enviar un correo al administrador principal acerca de la modificación de los parámetros base indicando el administrador que realizó la modificación y que información fue modificada.
- Enviar un correo electrónico al nuevo administrador con las credenciales de acceso.
- Enviar un correo al administrador con la recuperación de sus credenciales.
- Enviar un correo electrónico al administrador eliminado con un mensaje de agradecimiento.

### **2.1.2 REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA**

Los requerimientos no funcionales del sistema se listan a continuación:

- El prototipo usará SQL Server para almacenar la información en una base de datos.

- El prototipo utilizará Entity Framework para gestionar la información de la base de datos.
- El prototipo estará vinculado a la cuenta de correo electrónico del administrador principal para la emisión y recepción de correos electrónicos.

### 2.1.3 PARÁMETROS BASE DEL SISTEMA

Para determinar los parámetros base se analizó el «REGLAMENTO DE COBRO DE MATRÍCULAS Y ARANCELES DE LAS CARRERAS DE GRADO, TECNOLOGÍA SUPERIOR Y NIVELACIÓN DE LA ESCUELA POLITÉCNICA NACIONAL» y se obtuvieron los siguientes parámetros:

**Costo óptimo (CO):** valor correspondiente al costo anual por carrera por estudiante de la EPN. Considerando que la EPN adoptará el valor del costo óptimo del año inmediato anterior al de la matrícula.

**Costo óptimo por periodo académico (COP):** valor correspondiente al 50% del costo óptimo anual.

**Valor mínimo a pagar por periodo académico (Valor mínimo):** valor correspondiente al 10% del costo óptimo que se cobrará por concepto de aranceles y matrícula en cada periodo. Equivalente a la suma de los valores de matrícula y arancel mínimo.

**Valor matrícula mínimo (VM min.):** valor correspondiente al 10% del total del valor del arancel.

**Valor arancel mínimo (VA min.):** este valor será determinado en función del valor del costo óptimo y mínimo a pagar por periodo académico.

**Valor máximo a pagar por periodo académico (Valor máximo):** valor correspondiente al 50% del costo óptimo que se cobrará por concepto de aranceles y matrícula en cada periodo. Equivalente a la suma de los valores de matrícula y arancel máximo.

**Valor matrícula máximo (VM máx.):** Valor correspondiente al 10% del total del valor de arancel.

**Valor Arancel Máximo (VA máx.):** este valor será determinado en función del valor del costo óptimo y máximo a pagar por periodo académico.

**Horas por periodo académico (HPA):** este valor se calcula al considerar que un estudiante podrá inscribirse en 15 créditos, de acuerdo con la malla curricular de la carrera y que cada crédito corresponde a 48 horas de actividad de aprendizaje del estudiante; por tanto, el valor de HPA es de 720 horas.

**Horas promedio por periodo académico (HPPA):** valor que corresponde a las actividades de aprendizaje en contacto con el docente y practico-experimental (AC+AP). Este valor dependerá del porcentaje establecido para el cálculo de HPPA.

**Créditos por periodo académico (CPA):** valor correspondiente a la cantidad de créditos que un estudiante de la EPN podrá inscribirse por periodo académico de acuerdo a lo establecido en la malla curricular.

**Créditos por pérdida temporal (CPT):** valor correspondiente al 60% de la cantidad de créditos de las asignaturas permitidos en la malla curricular.

**Cálculo del costo por hora por periodo académico (CHC):** valor que corresponde al costo hora por asignatura para cada periodo académico.

**Asignación del costo por situación socioeconómica (CHCS):** valor que se ajustará con la información socioeconómica entregada por el estudiante de la EPN y del CHC.

**Valor matrícula extraordinaria (VMEX):** valor que se cobrará si se realiza la matriculación en el plazo establecido en matrículas extraordinarias. El recargo será del 25% de valor de matrícula que la EPN cobra al estudiante. Además, se considera la situación de gratuidad del estudiante de la EPN para realizar dicho cobro.

**Valor matrícula especial (VMES):** valor que se cobrará si se realiza la matriculación por concepto de matrículas especiales. El recargo será del 25% de valor de matrícula que la EPN cobra al estudiante. Además, se considera la situación de gratuidad del estudiante de la EPN para realizar dicho cobro.

**Recargo por Segunda Matrícula (R2):** valor que se cobrará por matriculación de asignaturas en segunda matrícula. Considerando la situación de gratuidad del estudiante y el costo por hora del quintil socioeconómico.

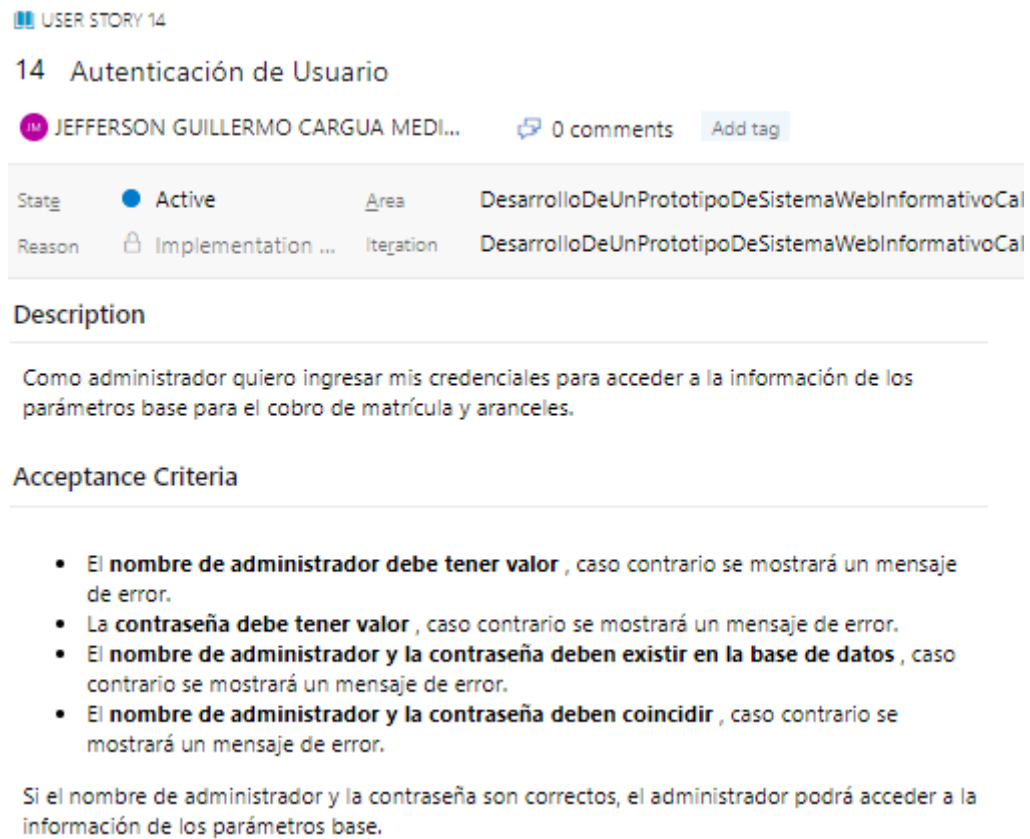
**Recargo por Tercera Matrícula (R3):** valor que se cobrará por matriculación de asignaturas en tercera matrícula. Considerando la situación de gratuidad del estudiante y el costo por hora del quintil socioeconómico.

#### **2.1.4 HISTORIAS DE USUARIO**

Las historias de usuario se desarrollaron utilizando la plataforma de Microsoft Azure DevOps. Esta herramienta permitió recolectar y redactar los requerimientos de usuario mediante el uso del formato indicado en la sección 1.3.

En la Figura 2.1 se muestra un ejemplo de una historia de usuario creada utilizando la herramienta Azure DevOps. Esta historia establece la tarea para realizar la autenticación de usuario que permite acceder a la información de los parámetros base definidos en el costo de las carreras de Tecnología e Ingeniería de la EPN.

Además, se requiere cumplir con los criterios de aceptación para permitir el acceso al sistema web cuando el nombre del administrador y la contraseña coincidan con la información almacenada en la base de datos.



USER STORY 14

### 14 Autenticación de Usuario

JEFFERSON GUILLERMO CARGUA MEDI... 0 comments Add tag

State	Active	Area	DesarrolloDeUnPrototipoDeSistemaWebInformativoCal
Reason	Implementation ...	Iteration	DesarrolloDeUnPrototipoDeSistemaWebInformativoCal

#### Description

Como administrador quiero ingresar mis credenciales para acceder a la información de los parámetros base para el cobro de matrícula y aranceles.

#### Acceptance Criteria

- El **nombre de administrador debe tener valor** , caso contrario se mostrará un mensaje de error.
- La **contraseña debe tener valor** , caso contrario se mostrará un mensaje de error.
- El **nombre de administrador y la contraseña deben existir en la base de datos** , caso contrario se mostrará un mensaje de error.
- El **nombre de administrador y la contraseña deben coincidir** , caso contrario se mostrará un mensaje de error.

Si el nombre de administrador y la contraseña son correctos, el administrador podrá acceder a la información de los parámetros base.

**Figura 2.1.** Historia de usuario - Acceso a la información de los parámetros base

Las historias de usuario restantes se incluyen en el Anexo A.

### 2.1.5 BACKLOG DE KANBAN

En el *backlog* de Kanban están todas las tareas a realizar para cumplir con los requerimientos descritos en las historias de usuario.

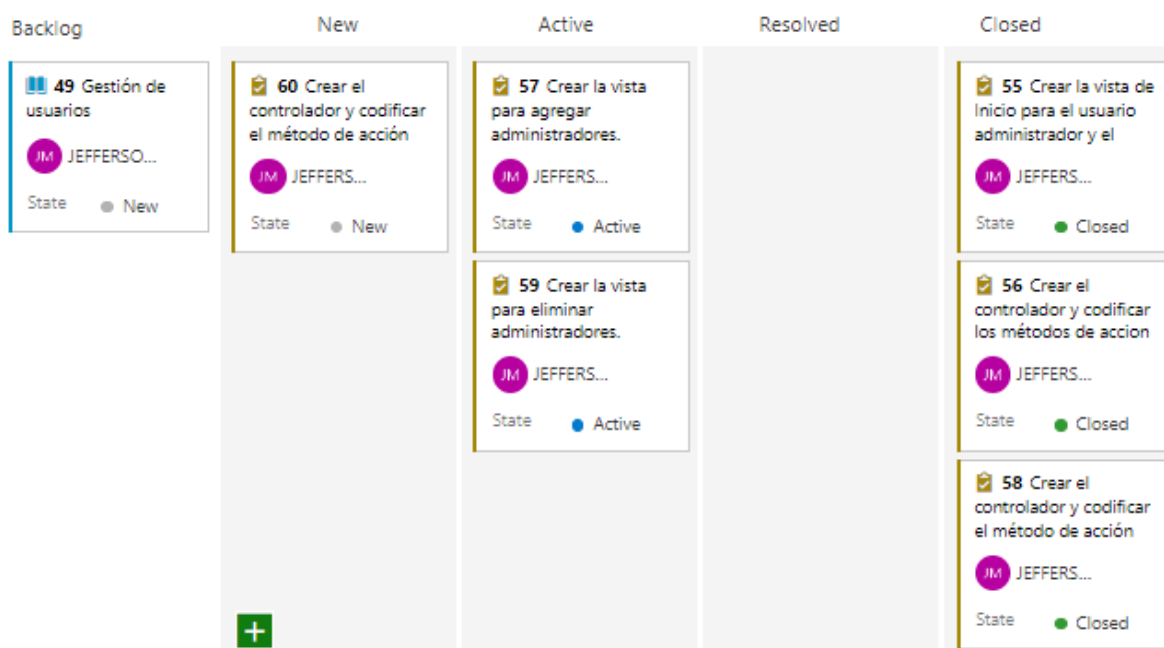
Las tareas se desarrollan de forma ordenada y cumpliendo con los requisitos descritos en cada una de las historias de usuario. Cada tarea resuelta se mueve a través del tablero Kanban para evaluar los avances alcanzados del prototipo.

En la Tabla 2.1 se presenta el *backlog* generado para este Trabajo de Titulación.

**Tabla 2.1. Backlog definido**

Requerimientos	Tareas
Generar la base de datos que permita almacenar información de los administradores y de los parámetros base.	Codificar la base de datos en Microsoft SQL Server.
	Introducir datos iniciales en la base de datos.
Generar el proyecto que permita desarrollar las funcionalidades del prototipo.	Crear un proyecto ASP.NET MVC en Visual Studio.
	Instalar las herramientas de desarrollo web y librerías.
	Conectar la base de datos al proyecto empleando Entity Framework.
	Codificar las clases para la interacción del sistema.
Generar la funcionalidad que permita la gestión de usuarios.	Crear la vista de Inicio.
	Crear el controlador y codificar los métodos de acción para la vista Inicio.
	Crear la vista para agregar administradores.
	Crear el controlador y codificar los métodos de acción para agregar administradores.
	Crear la vista para eliminar administradores.
	Crear el controlador y codificar los métodos de acción para eliminar administradores.
Generar el <i>Login</i> que permita la autenticación de usuario.	Crear un formulario de <i>login</i> en la vista Inicio
	Implementar un método de acción en el controlador de la vista Inicio para autenticación de usuarios.
	Implementar un método de acción en el controlador de la vista Inicio para recuperación de clave por correo.
Generar la funcionalidad que permita gestionar la información de los parámetros base almacenada en la base de datos.	Crear la vista para visualizar la información de los parámetros base.
	Crear el controlador y codificar los métodos de acción para visualizar la información de los parámetros base.
	Crear la vista para editar la información de los parámetros base.
	Crear el controlador y codificar los métodos de acción para editar la información de los parámetros base.
Generar la funcionalidad que permita consultar y calcular el valor a pagar por matriculación en la EPN.	Crear las vistas para ingresar información de los estudiantes o aspirantes de la EPN y presentar el cálculo del valor a pagar.
	Crear el controlador y codificar los métodos de acción para ingresar información de los estudiantes o aspirantes de la EPN y presentar el cálculo del valor a pagar.

El tablero Kanban contiene el *backlog* de Kanban, con lo que se determinó el flujo de trabajo con el que se desarrollará el prototipo. A continuación, se muestra un ejemplo del flujo de trabajo que se realizó para llevar a cabo las tareas que se encontraban en el tablero Kanban, como se muestra en la Figura 2.2.



**Figura 2.2.** Tablero Kanban del prototipo en Azure DevOps

## 2.2 DISEÑO

Como parte del diseño del prototipo se planteó la arquitectura del sistema, el diagrama de casos de uso, diagrama de actividades, diagrama de secuencia, el esquema relacional de la base de datos y los *sketches* de las vistas del sistema.

### 2.2.1 ARQUITECTURA DEL SISTEMA

La arquitectura del sistema se desarrolló de acuerdo a las capas definidas en el patrón MVC y de los componentes con los que interactúan. Por un lado, los usuarios (administradores, estudiantes o aspirantes) y por otro lado el servidor web considerando la: arquitectura MVC y la base de datos.

En la Figura 2.3 se presenta la arquitectura del sistema. Se aprecia que los usuarios realizan peticiones HTTP mediante la vista del prototipo, las cuales son el resultado de acciones realizadas por el usuario sobre la interfaz, como presionar un botón; el servidor al recibir la petición debe entregarla a un controlador específico, el cual maneja dicha acción, y si es necesario accede al modelo en busca de datos de acuerdo con la acción



solicitada en la vista del patrón MVC. Finalmente, el controlador actualiza el contenido de la vista para que el usuario obtenga una respuesta HTTP a la acción que realizó.

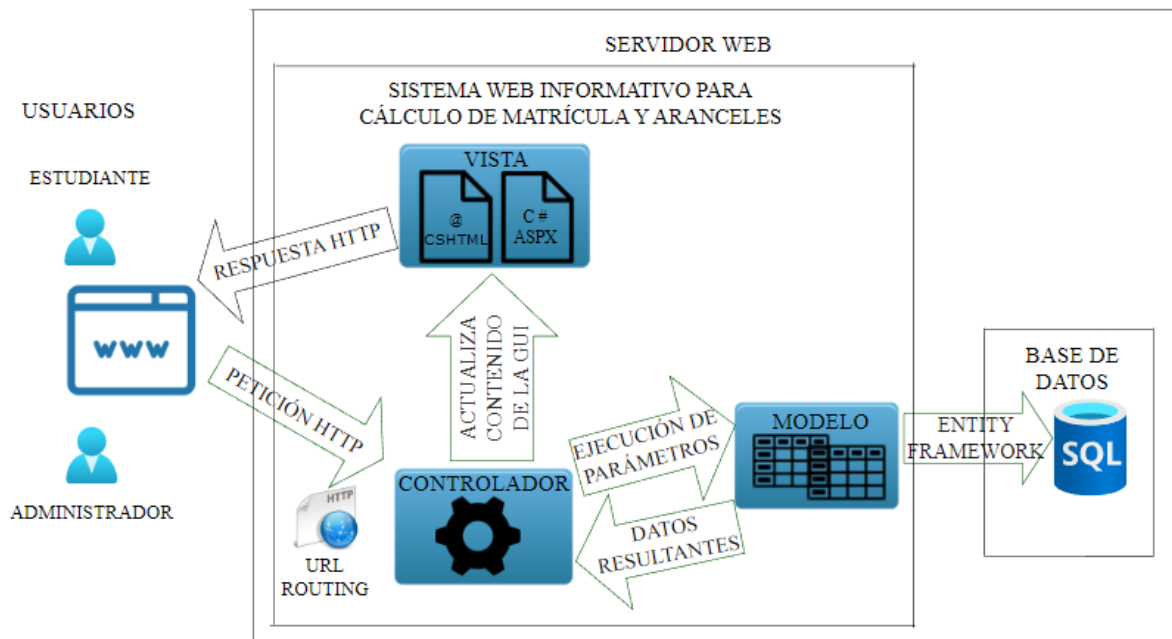
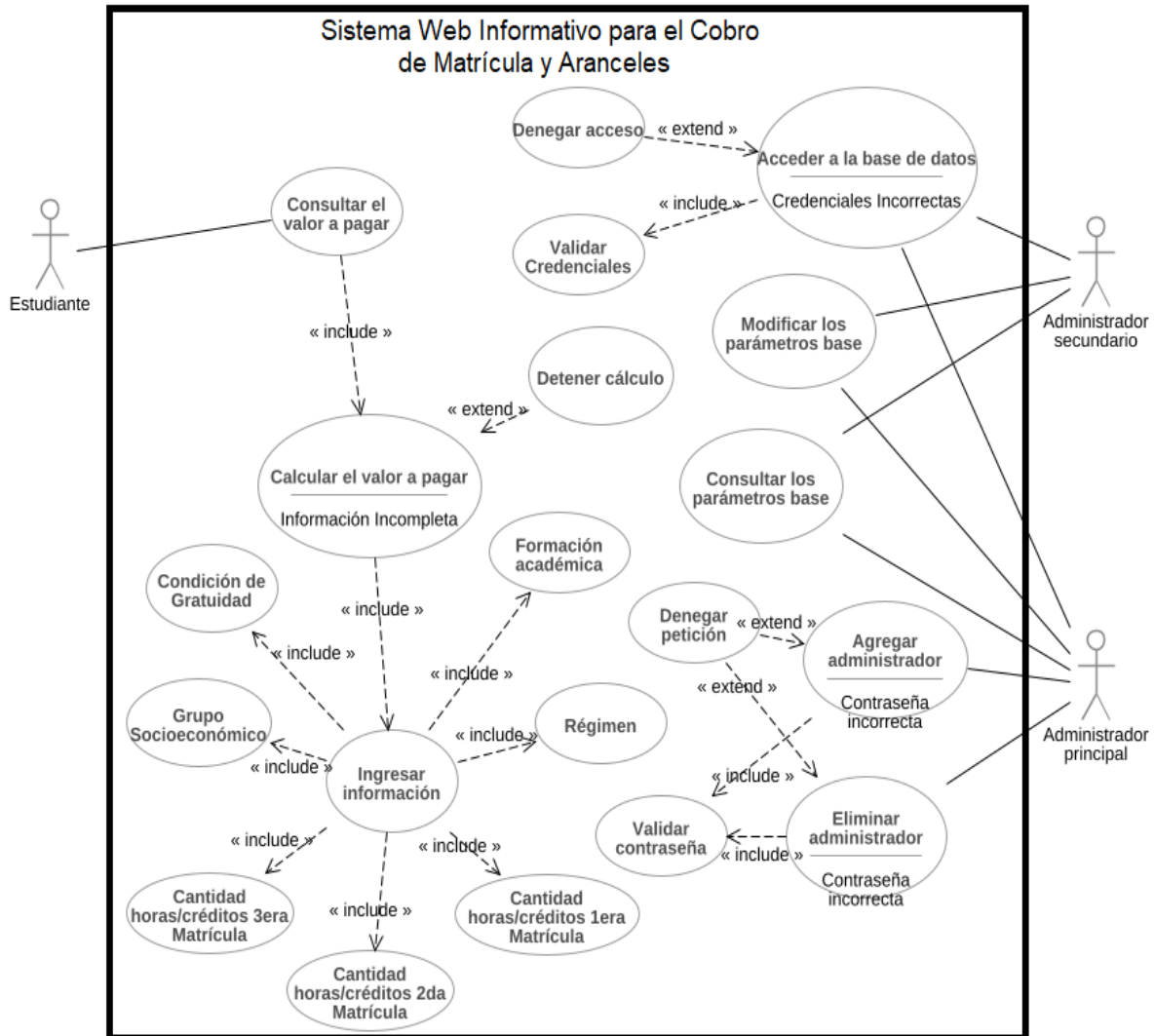


Figura 2.3. Arquitectura del sistema

## 2.2.2 DIAGRAMAS

El diagrama de casos de uso se desarrolló a partir de los requerimientos de usuarios. En la Figura 2.4 se muestra el diagrama de casos de uso del prototipo, en el que se puede observar a los actores y las acciones que realizan en el sistema web.

Así mismo, se muestran las relaciones que se tiene entre los casos de uso. Por ejemplo, para ejecutar el caso de uso *Calcular el valor a pagar* se tiene una relación *include* con el caso de uso *Ingresar información*, este caso de uso se debe ejecutar primero para poder cumplir con la solicitud del estudiante de ver el valor a pagar por la matriculación. También, se tiene una relación *extend* entre los casos de uso *Calcular valor a pagar* y *Detener cálculo*, esta relación se ejecuta cuando el estudiante ha ingresado información incompleta para realizar el cálculo, por tanto, el sistema detiene el cálculo del valor a pagar hasta que se ingrese la información correctamente.

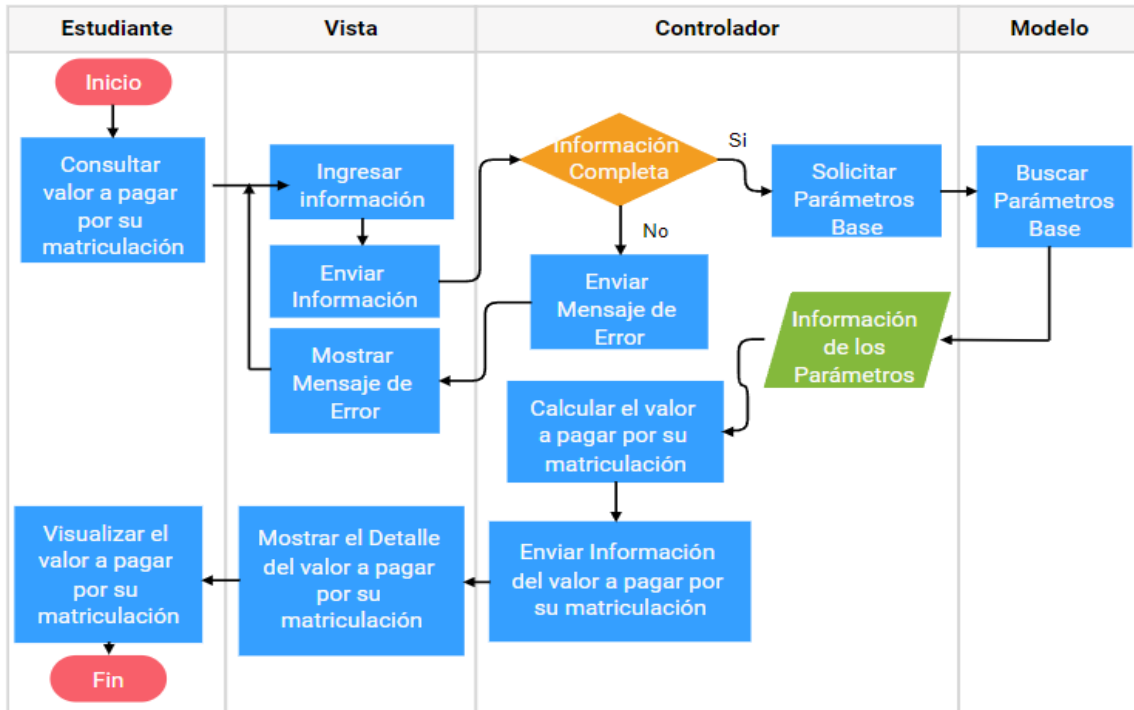


**Figura 2.4.** Diagrama de casos de uso

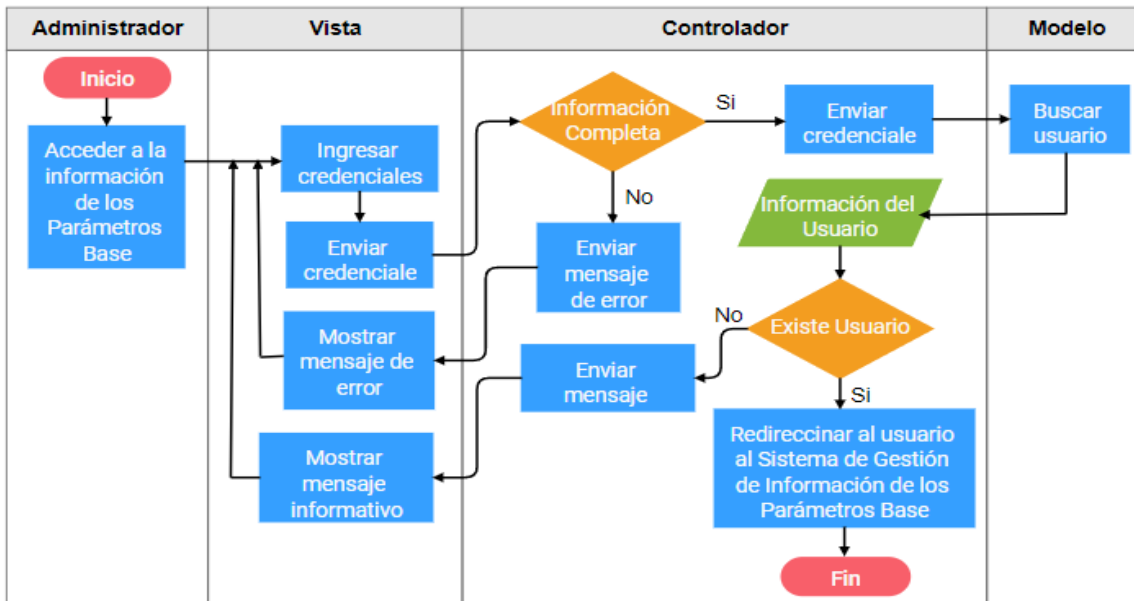
El diagrama de actividades se diseñó de acuerdo a las acciones que los actores del prototipo realizarán desde el ingreso hasta la salida correspondiente. Además, se detalla el flujo de trabajo, donde se modelan las actividades y condiciones que se codificarán.

En la Figura 2.5 se muestra el diagrama de actividades que permite al usuario estudiante consultar el valor a pagar por concepto de matrícula y aranceles en la EPN.

En la Figura 2.6, se muestra como ejemplo el diagrama de actividades que permite al usuario administrador acceder a la información de los parámetros base para el cálculo de matrícula y aranceles de las carreras de Ingeniería o Tecnología de la EPN.



**Figura 2.5.** Diagrama de actividades - Consulta del valor a pagar

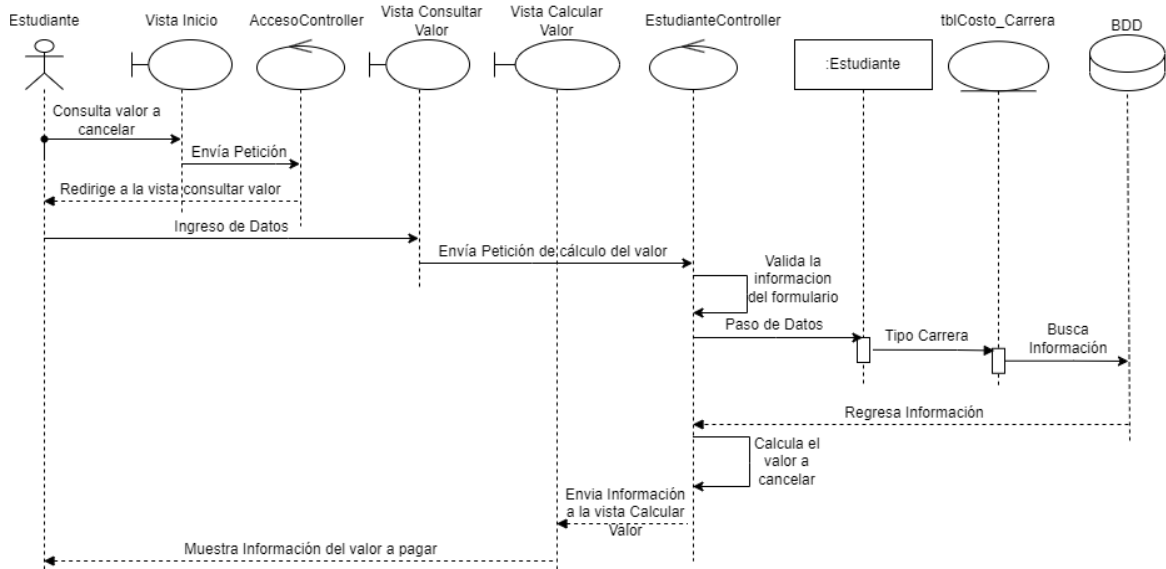


**Figura 2.6.** Diagrama de actividades - Acceso a la información de los parámetros base

Los diagramas de actividades restantes se incluyen en el Anexo B.

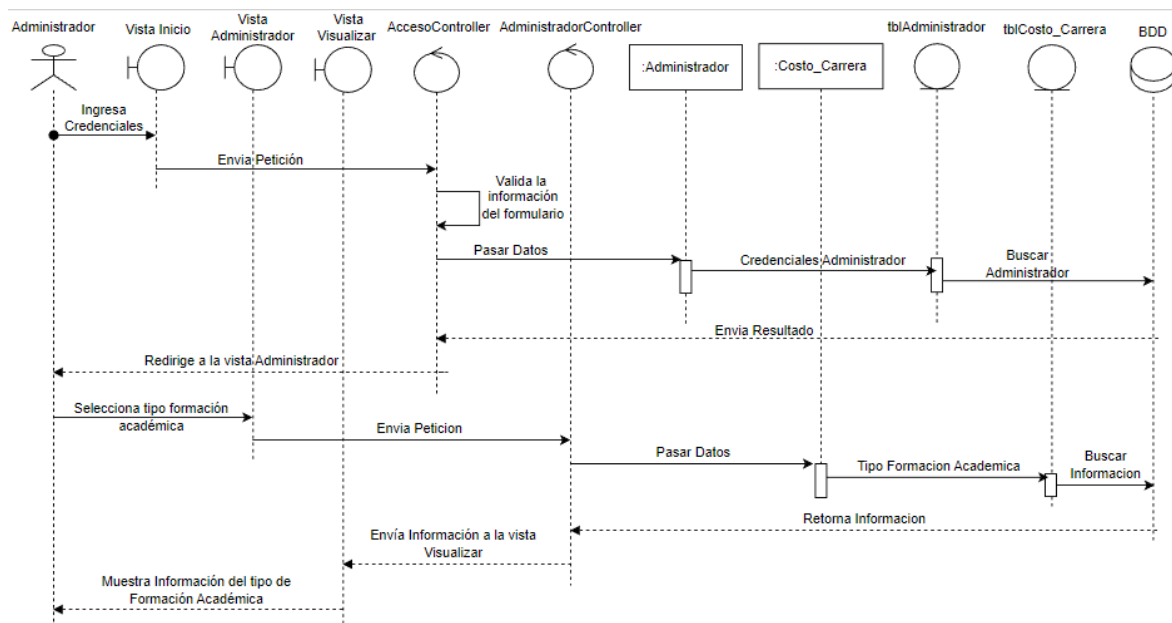
El diagrama de secuencia se diseñó a partir de la secuencia de interacciones a través del tiempo entre los actores y el prototipo. Estos diagramas describen el intercambio de mensajes y el orden de su ejecución para cumplir con los requerimientos de los usuarios.

En la Figura 2.7 se muestra el intercambio de mensajes entre el usuario estudiante y el prototipo, para poder consultar el valor a pagar.



**Figura 2.7.** Diagrama de secuencia - Consultar valor a pagar por la matriculación

En la Figura 2.8 se muestra el intercambio de mensajes entre el usuario administrador y el prototipo para permitir el acceso a la información de los parámetros base para el cálculo de matrícula y aranceles de la EPN mediante el ingreso de las credenciales de acceso. Así mismo, se aprecia un intercambio de mensajes para que el usuario administrador visualice la información de los parámetros base.



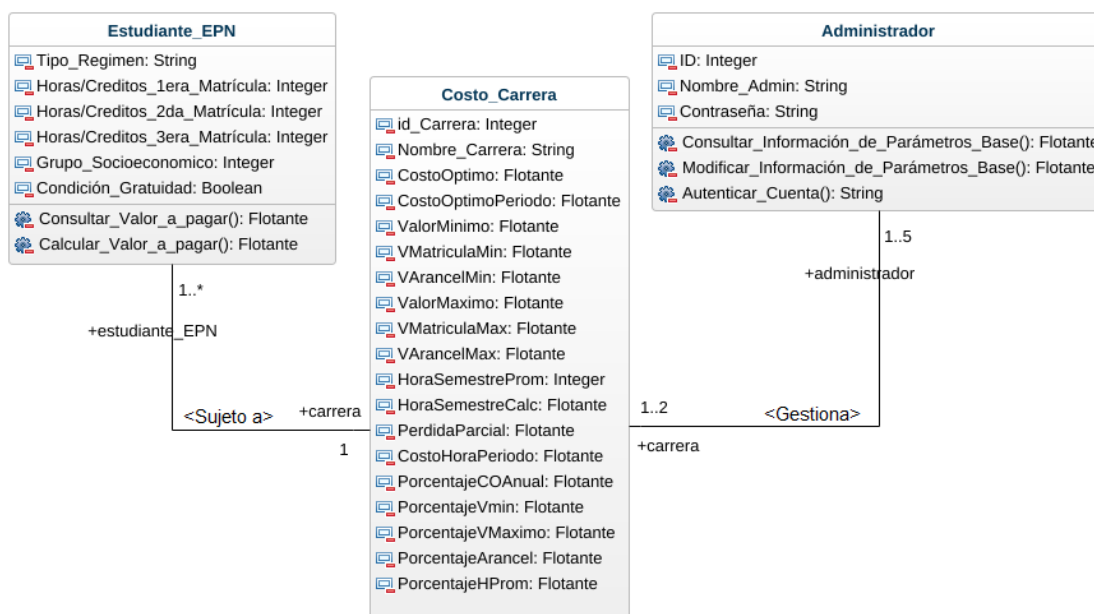
**Figura 2.8.** Diagrama de secuencia – Acceso y visualización de la información de los parámetros base

Los diagramas de secuencia restantes se incluyen en el Anexo C.

El diagrama de clases se diseñó a partir de la información obtenida del diagrama de casos de uso y de secuencia del prototipo para cobro de matrícula y aranceles de la EPN. Para el diseño del diagrama de clases se utilizó la programación orientada a objetos, para definir las clases, atributos y métodos.

En la Figura 2.9 se muestra el diagrama de clases del prototipo. Se disponen de las siguientes clases:

- **Estudiante\_EPN:** permitirá gestionar la información necesaria de los estudiantes/aspirantes para consultar y realizar los cálculos del valor a pagar por la matriculación en el prototipo.
- **Costo\_Carrera:** permitirá gestionar la información de los parámetros base de las carreras de Tecnología e Ingeniería para el cálculo de matrícula y aranceles.
- **Administrador:** permitirá almacenar información de los administradores. Además, sirve para verificar si las credenciales del administrador son válidas para acceder a la información de los parámetros base.



**Figura 2.9.** Diagrama de clases

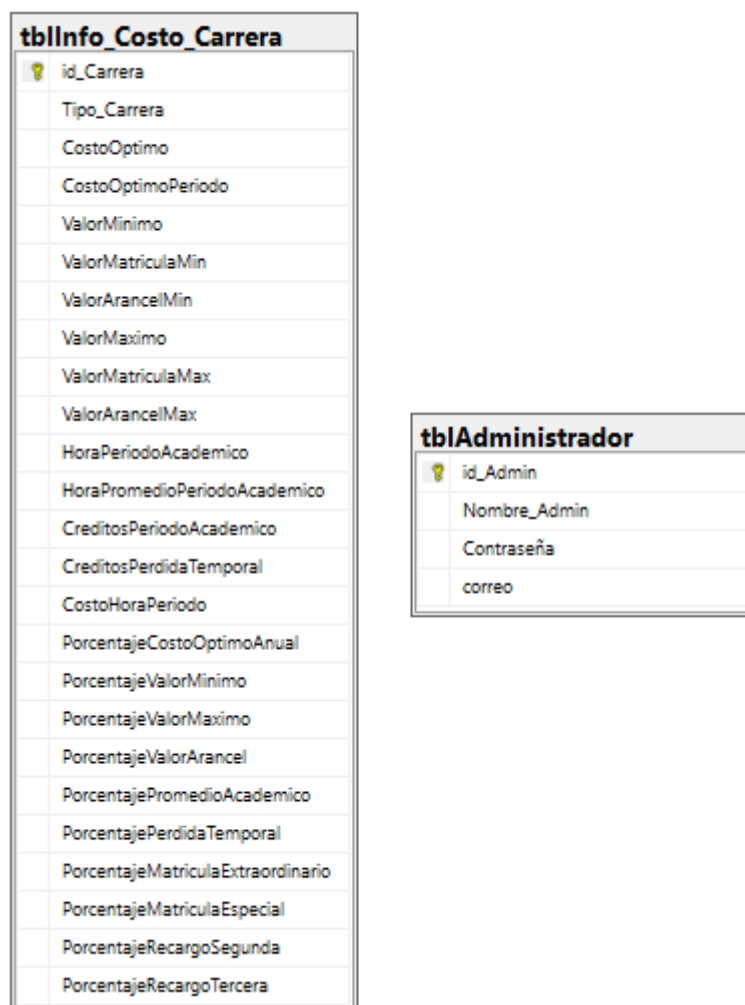
El diagrama relacional de la base de datos está definido en tablas y atributos, que permiten describir los objetos que interactúan en el sistema y los tipos de datos que se almacenan.

En la Figura 2.10 se muestra el diagrama relacional de la base de datos del sistema web para el cobro de matrícula y aranceles de la EPN.

A continuación, se describen las tablas:

**Administrador:** usada para almacenar la información de los usuarios del rol de Administrador, en esta tabla se tiene como llave primaria el `idAdmin` que sirve para identificar al administrador que gestione la información de los parámetros base de la tabla `InfoCostoCarrera`. También, cuenta con otros atributos como `NombreAdmin` y `contraseña` que sirve para autenticarse y acceder a la información de los parámetros base de las carreras de Tecnología e Ingeniería de la tabla `InfoCostoCarrera`.

**InfoCostoCarrera:** usada para almacenar la información de los parámetros base para el cálculo de matrícula y aranceles en la EPN. La tabla tiene como llave primaria el `idCarrera` que sirve para identificar el tipo de formación académica. Los atributos restantes sirven para almacenar los valores de los parámetros base de las carreras de Tecnología e Ingeniería de acuerdo con la normativa. Además, estos atributos sirven para realizar el cálculo de los valores a pagar por parte del usuario estudiante por concepto de matrícula y aranceles.



**Figura 2.10.** Diagrama Relacional de la base de datos

### 2.2.3 SKETCHES

Las interfaces del prototipo permitirán interactuar con este. Las interfaces están diseñadas en función del levantamiento de historias de usuario previamente elaboradas. Para las interfaces se definieron *sketches* para planificar los diseños que tendrán las vistas del prototipo.

En la Figura 2.11 se muestra el *sketch* para escoger el rol de usuario. El usuario administrador deberá ingresar sus credenciales para acceder a la información de los parámetros base, mientras que el usuario estudiante no necesita de credenciales para consultar el valor a pagar.

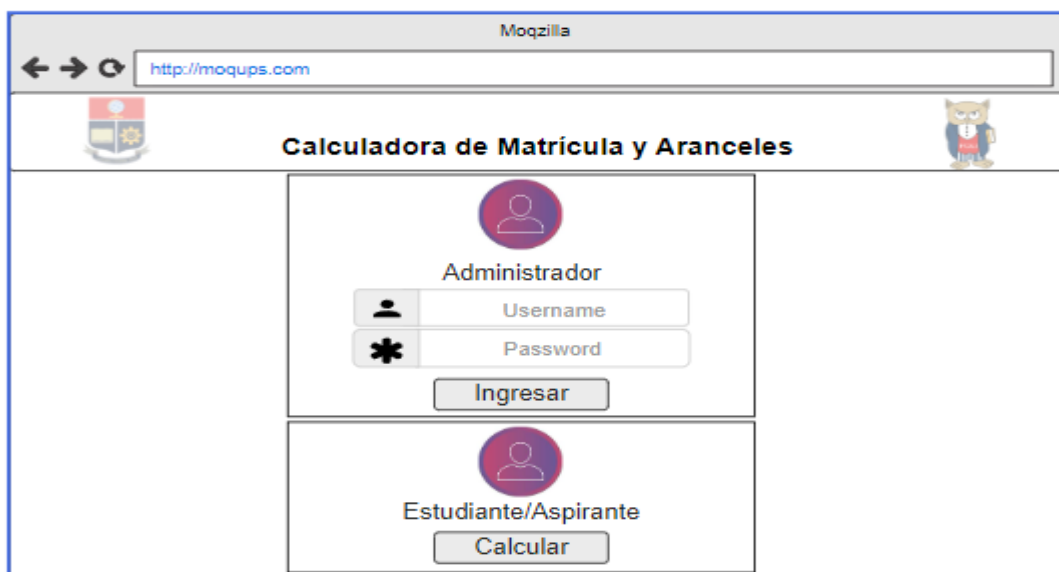


Figura 2.11. Sketch para escoger el rol de usuario

En la Figura 2.12 se muestra el *sketch* para seleccionar el tipo de carrera que el usuario administrador desea visualizar o modificar.



Figura 2.12. Sketch para seleccionar el tipo de carrera

Los *sketches* restantes se incluyen en el Anexo D.

## 2.3 IMPLEMENTACIÓN

Para la implementación del prototipo en esta sección se presenta un resumen de la codificación de la base de datos en Microsoft SQL Server 2019 y la creación del proyecto ASP.NET MVC en Visual Studio 2022. Así mismo, se presenta la instalación de las herramientas y librerías necesarias para el desarrollo del prototipo. Además, se presenta la conexión de la base de datos mediante el uso de Entity Framework.

Finalmente, se presenta la codificación de las capas del patrón Modelo – Vista - Contralador del prototipo.

### 2.3.1 CODIFICACIÓN DE LA BASE DE DATOS

Para la codificación de la base de datos del prototipo se generó un *script* con base en el diagrama relacional presentado en la Figura 2.10.

En el Figura 2.13 se muestran las líneas de código necesarias para la generación y utilización de la base de datos del prototipo. La base de datos para el prototipo se denomina dbSistemaWeb.

```
-----Creación de la Base de Datos-----  
Go  
Create Database dbSistemaWeb  
Go  
Use dbSistemaWeb  
Go
```

**Figura 2.13.** *Script* para la creación de la base de datos

En el Figura 2.14 se muestran las líneas necesarias para la creación de la tabla. En este fragmento se define el atributo `idAdmin` como llave primaria, esta será única y auto incrementable. Así mismo, se definen atributos obligatorios que almacenarán la información de los usuarios de administradores. Esta información permite gestionar el acceso a la información de los parámetros base.

```
-----Creación de la Tabla Administrador-----  
CREATE TABLE tbl_Administrador(  
id_Admin int PRIMARY KEY IDENTITY(1,1) NOT NULL,  
Nombre_Admin nvarchar(50) NOT NULL,  
Contraseña nvarchar(30) NOT NULL,  
correo nvarchar(50) NOT NULL,  
)
```

**Figura 2.14.** *Script* para la creación de la tabla



En el Figura 2.15 se muestran las líneas necesarias para la creación de la tabla `tblInfoCostoCarrera`. En este fragmento del *script* se define el atributo `idCarrera` como llave primaria. Así mismo, se definen atributos obligatorios que almacenarán la información de los parámetros base para el cálculo de matrícula y aranceles de la EPN. La información de esta tabla permitirá realizar el cálculo de los posibles valores que deberán cancelar los estudiantes de la EPN o aspirantes.

```

-----Creación de la Tabla Información Costo Carrera-----
CREATE TABLE tbl_Info_Costo_Carrera(
    id_Carrera int PRIMARY KEY NOT NULL,
    Tipo_Carrera varchar(20) NOT NULL,
    CostoOptimo float NOT NULL,
    CostoOptimoPeriodo float NOT NULL,
    ValorMinimo float NOT NULL,
    ValorMatriculaMin float NOT NULL,
    ValorArancelMin float NOT NULL,
    ValorMaximo float NOT NULL,
    ValorMatriculaMax float NOT NULL,
    ValorArancelMax float NOT NULL,
    HoraPeriodoAcademico int NOT NULL,
    HoraPromedioPeriodoAcademico float NOT NULL,
    CreditosPeriodoAcademico int NOT NULL,
    CreditosPerdidaTemporal float NOT NULL,
    CostoHoraPeriodo float NOT NULL,
    PorcentajeCostoOptimoAnual float NOT NULL,
    PorcentajeValorMinimo float NOT NULL,
    PorcentajeValorMaximo float NOT NULL,
    PorcentajeValorArancel float NOT NULL,
    PorcentajePromedioAcademico float NOT NULL,
    PorcentajePerdidaTemporal float NOT NULL,
    PorcentajeMatriculaExtraordinario float NOT NULL,
    PorcentajeMatriculaEspecial float NOT NULL,
    PorcentajeRecargoSegunda float NOT NULL,
    PorcentajeRecargoTercera float NOT NULL,
)

```

**Figura 2.15.** *Script* para la creación de la tabla

### 2.3.2 PROYECTO DE VISUAL STUDIO

Para el desarrollo del prototipo se instaló la versión de Visual Studio 2022, con las herramientas necesarias para utilizar el *framework* ASP.NET MVC.

En la Figura 2.16 y en la Figura 2.17 se muestran los pasos para la creación del proyecto ASP.NET MVC en Visual Studio. Se observa que primero se debe crear un nuevo proyecto Aplicación web ASP.NET (.NET Framework), y a continuación se elige la arquitectura MVC para el desarrollo del proyecto.

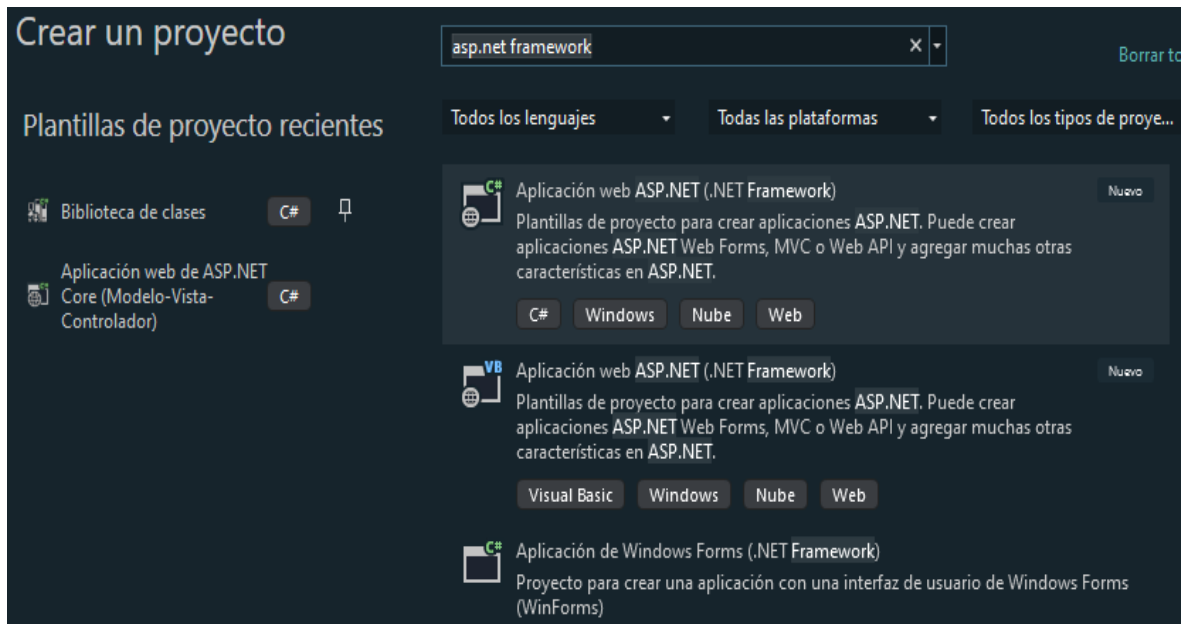


Figura 2.16. Creación del proyecto ASP.NET en Visual Studio



Figura 2.17. Selección de la arquitectura MVC del proyecto en ASP.NET

Como parte del desarrollo del *front-end* del prototipo se instaló en Visual Studio la herramienta denominada Bootstrap.

En la Figura 2.18 y en la Figura 2.19 se presenta el proceso de instalación de Bootstrap. Como primer paso se accede a Administrar paquetes NuGet para la solución y luego en la ventana NuGet – Solución se debe buscar Bootstrap y proceder a instalarlo.

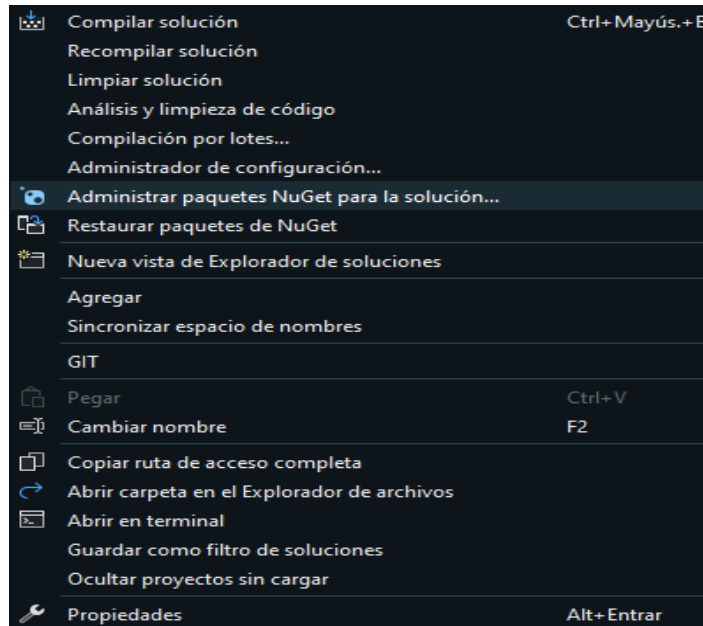


Figura 2.18. Instalación de Bootstrap (parte I de II)

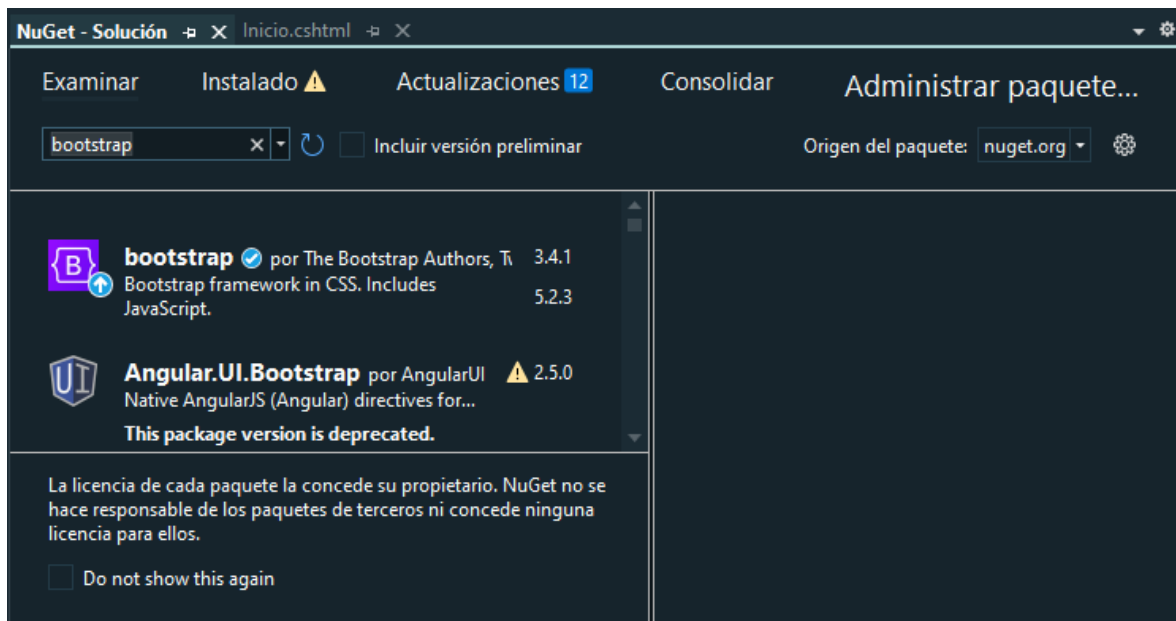


Figura 2.19. Instalación de Bootstrap (parte II de II)

### 2.3.3 CONEXIÓN DE LA BASE DE DATOS

Para la conexión de la base de datos con el proyecto creado en Visual Studio, se emplea la tecnología Entity Framework.

En la Figura 2.20 y en la Figura 2.21 se presenta el proceso de instalación de Entity Framework. Este proceso es muy similar al realizado para instalar Bootstrap, con la diferencia que se busca Entity Framework.

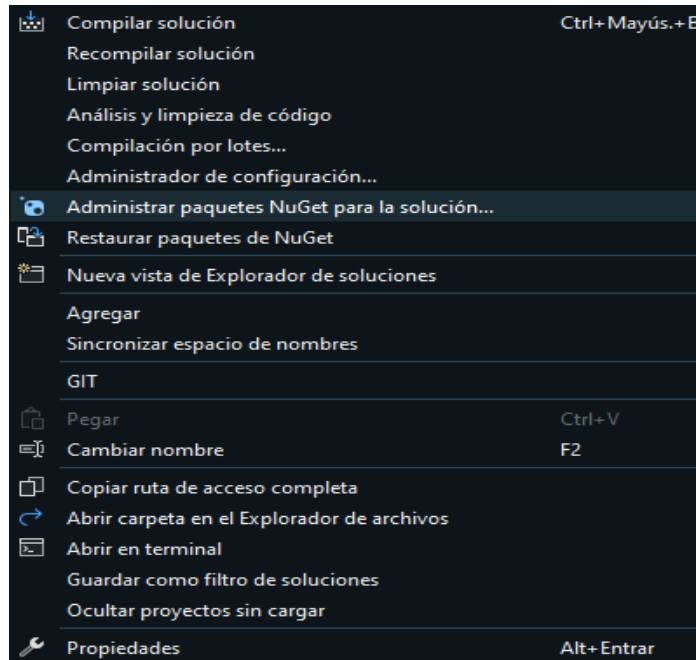


Figura 2.20. Instalación de Entity Framework (parte I de II)

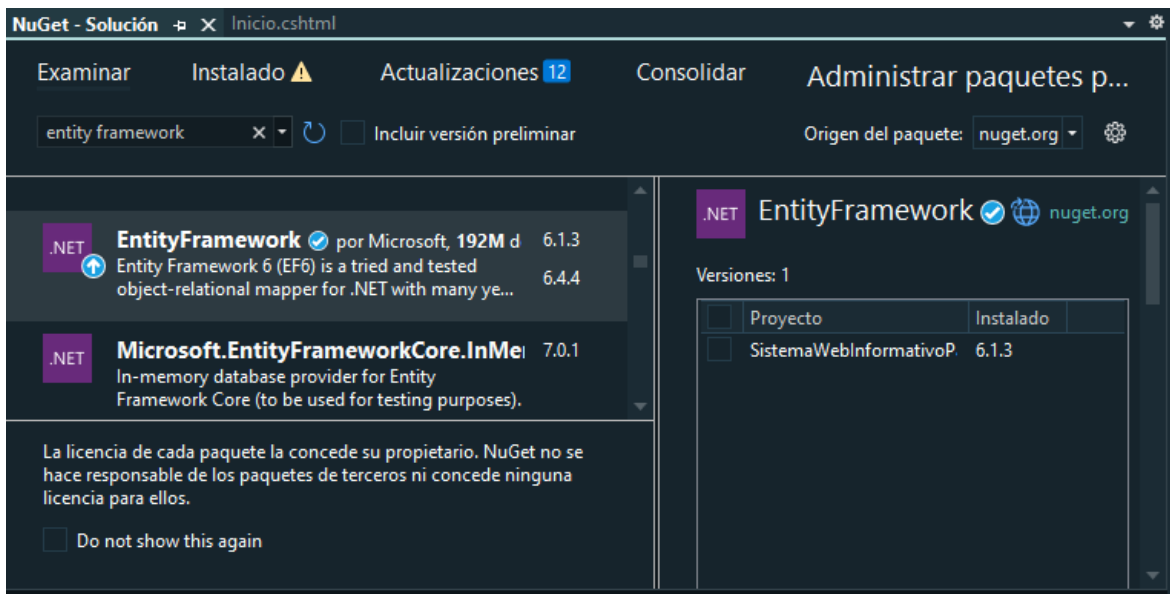
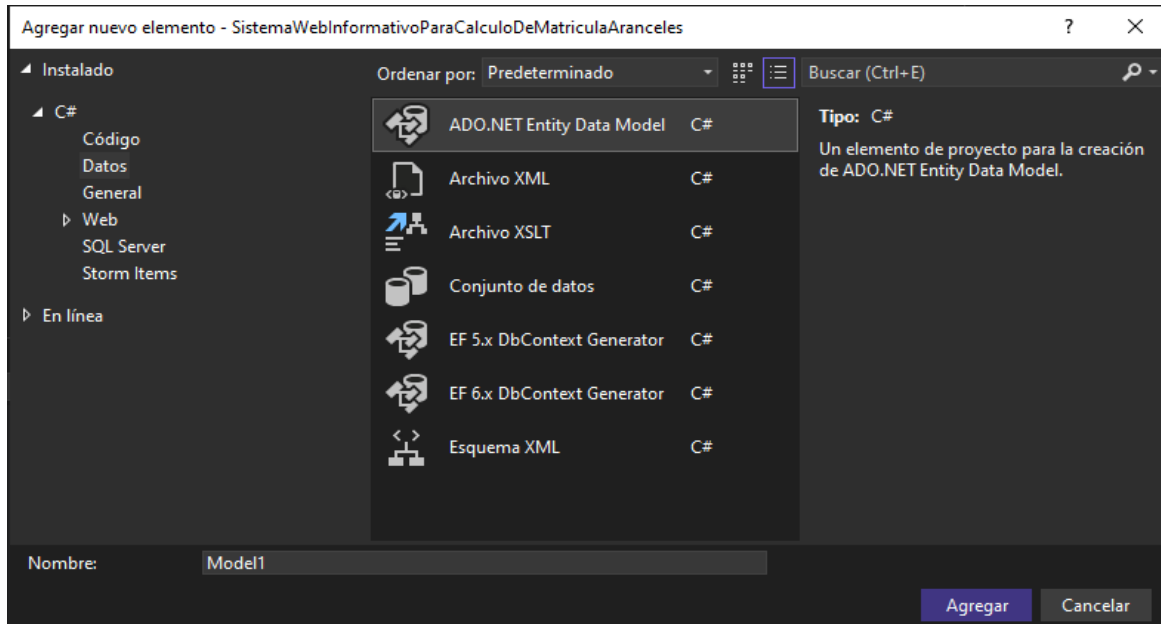


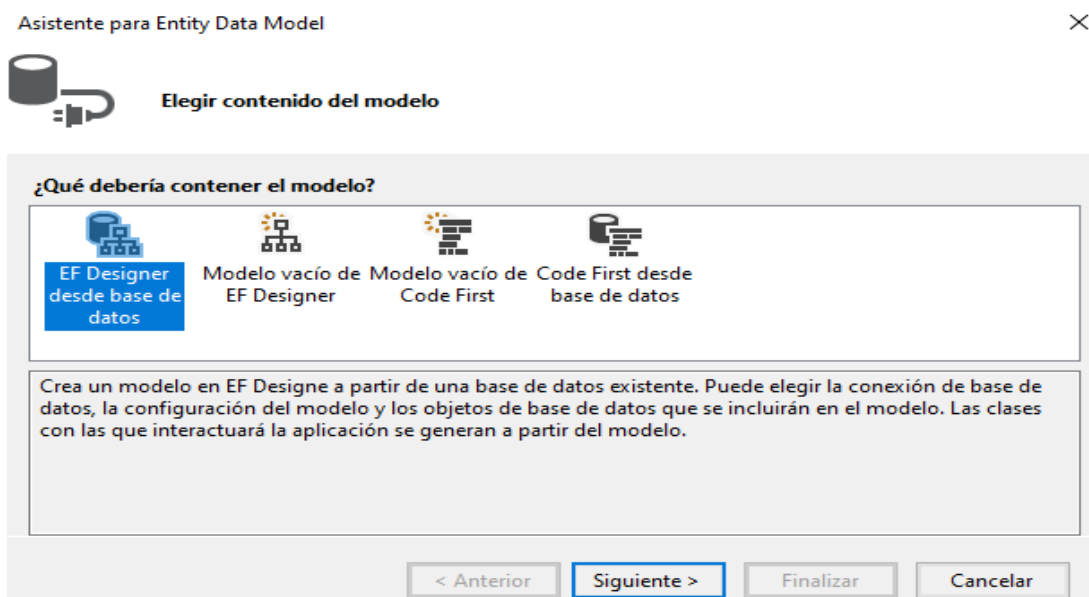
Figura 2.21. Instalación de Entity Framework (parte II de II)

Los paquetes instalados con Entity Framework se agregan a la biblioteca del Entity Data Model. A continuación, se muestra la opción ADO.NET Entity Data Model, que se utiliza para generar la conexión entre el sistema de administración Microsoft SQL Server y el proyecto creado en Visual Studio, como se observa en la Figura 2.22.



**Figura 2.22.** Selección de la opción ADO.NET Entity Data Model

Luego se debe escoger la opción *EF Designer* desde la base de datos para utilizar la base de datos generada y presionar la opción *Siguiente*, como se muestra en la Figura 2.23.



**Figura 2.23.** Enlace de la base de datos con EF Designer desde la base de datos

A continuación, se presenta una ventana en el que se debe ingresar el nombre del servidor de la base de datos y luego seleccionar la base de datos creada previamente, la cual se denominó `dbSistemaWeb`, lo cual se aprecia en la Figura 2.24.

Especifique la información para conectarse al origen de datos seleccionado o haga clic en "Cambiar" para elegir otro origen o proveedor de datos.

Origen de datos:

Nombre del servidor:

Conexión con el servidor

Autenticación:

Nombre de usuario:

Contraseña:

Guardar mi contraseña

Establecer conexión con una base de datos

Seleccionar o escribir el nombre de la base de datos:

**Figura 2.24.** Selección de la base de dato

En la Figura 2.25 se muestra la cadena de conexión generada para enlazar la base de datos con el proyecto.



**Elegir la conexión de datos**

**¿Qué conexión de datos debe usar la aplicación para conectarse a la base de datos?**

Esta cadena de conexión parece contener datos confidenciales (por ejemplo, una contraseña) que son necesarios para conectarse con la base de datos. Almacenar datos confidenciales en la cadena de conexión puede suponer un riesgo para la seguridad. ¿Desea incluir estos datos en la cadena de conexión?

No, excluir datos confidenciales de la cadena de conexión. Los estableceré en el código de mi aplicación.

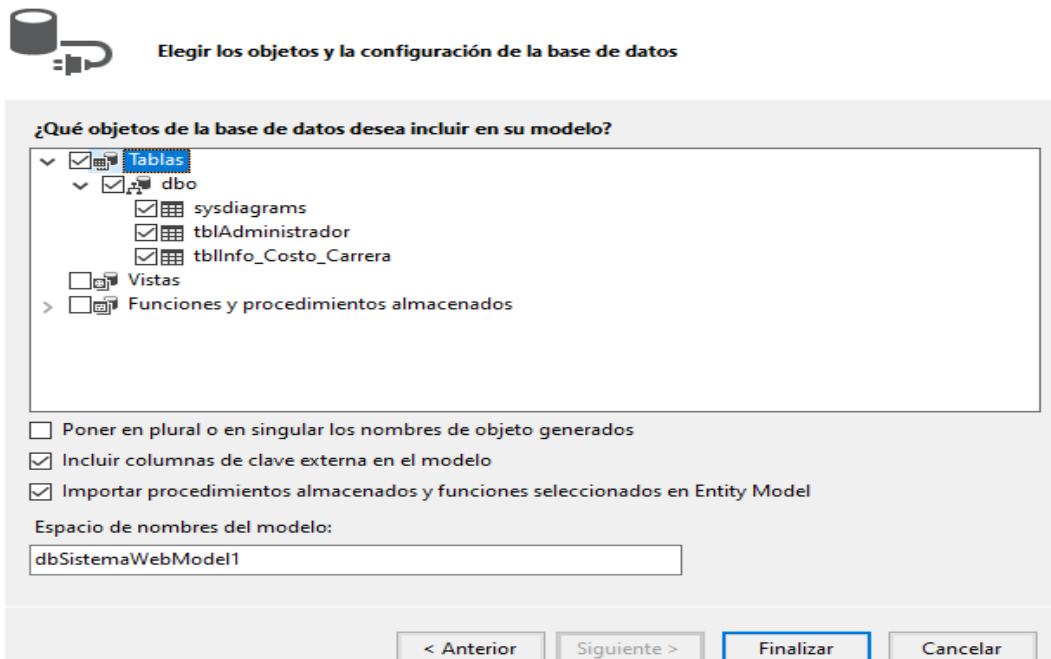
Sí, incluir datos confidenciales en la cadena de conexión.

Cadena de conexión:

Guardar configuración de conexión en Web.Config como:

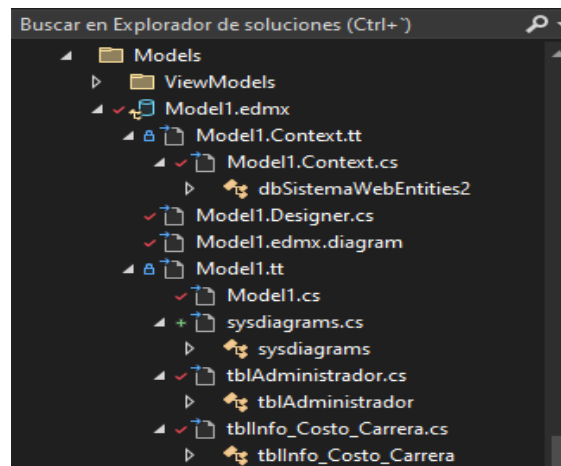
**Figura 2.25.** Cadena de conexión a la base de datos

Para finalizar el proceso se agregan las tablas generadas previamente, como se muestra en la Figura 2.26.



**Figura 2.26.** Creación del enlace con la base de dato

En la Figura 2.27 se muestran los archivos creados por Entity Framework en el proyecto. Estos archivos contienen información que describen el modelo conceptual y el modelo de almacenamiento de la base de datos.



**Figura 2.27.** Archivos generados por Entity Framework

### 2.3.4 CODIFICACIÓN DE LAS CLASES

Entity Framework permite generar modelos de almacenamiento, que son clases que se crean basándose en las tablas generadas en la base de datos. Estas clases permiten la

interacción con los atributos y propiedades de las tablas para gestionar la información de la base de datos.

En la Figura 2.28 se muestra de la clase `tblAdministrador` generada por Entity Framework.

```

12 using System;
13 using System.Collections.Generic;
14
15 public partial class tblAdministrador
16 {
17     0 referencias
18     public int id_Admin { get; set; }
19     0 referencias
20     public string Nombre_Admin { get; set; }
21     0 referencias
22     public string Contraseña { get; set; }
23     0 referencias
24     public string correo { get; set; }
25 }

```

Figura 2.28. Modelo de almacenamiento generado por Entity Framework

Además, se crearon clases adicionales para gestionar las solicitudes de los usuarios y cumplir con los requerimientos del prototipo. En la Figura 2.29 se muestra las clases creadas en el proyecto.

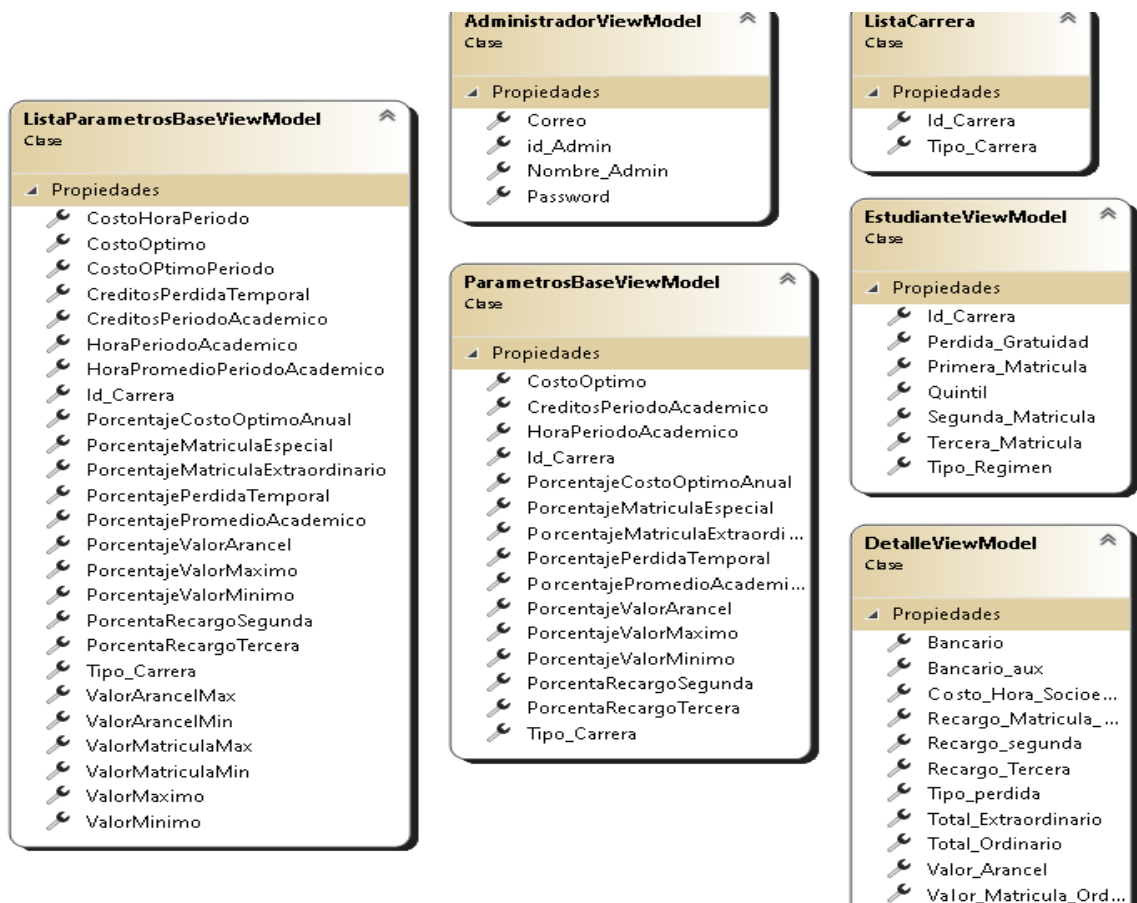
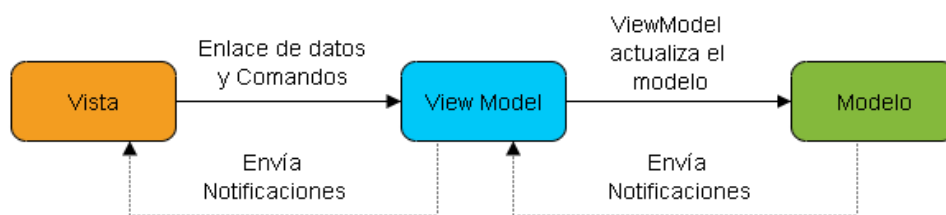


Figura 2.29. Clases usadas en el proyecto



A estas clases adicionales se les conoce como ViewModel y se usan para almacenar datos provenientes de las vistas y adaptarlos a las clases que se generaron por Entity Framework. Un ViewModel es muy similar a un modelo de almacenamiento, con la diferencia de que se usa para simplificar, extender o representar a un modelo existente. Por tanto, al emplear formularios, el uso del ViewModel facilita el manejo de datos que permiten la adecuación de modelos de datos independientemente de la vista.

En la Figura 2.30 se muestra el diagrama conocido como Modelo-Vista-Modelo, en el que se observa la interacción entre el ViewModel con las capas vista y modelo de la arquitectura Modelo – Vista – Controlador.



**Figura 2.30.** Modelo-Vista-Modelo

Es importante mencionar que al emplear clases ViewModel para el manejo de datos desde las vistas, se puede agregar la librería Data Annotation, la cual permite agregar etiquetas a los atributos de la clase agregando características únicas para validar los datos y cumplir con los criterios de aceptación.

Las etiquetas que se utilizaron en el proyecto son:

- `Required` - especifica que la propiedad es un campo obligatorio.
- `DisplayName` - define el texto que se utilizará en los campos del formulario y los mensajes de validación.
- `DataType` - especifica el nombre de un tipo adicional que debe asociarse a un campo de datos.
- `Range` - da un valor máximo y mínimo para un campo numérico.
- `EmailAddress` - valida una dirección de correo electrónico.

En la Figura 2.31 se muestra la codificación de la clase `AdministradorViewModel`. Esta clase almacenará información como: el ID de administrador, el nombre de usuario y la contraseña para autenticarse y acceder a la información de los parámetros base y el correo para recuperar las credenciales de acceso. Así mismo, se muestra el uso de las etiquetas de validación de Data Annotation como por ejemplo `[Required]` para indicar que el valor de la propiedad `Nombre_Admin` es obligatorio y que, en caso de enviar valores vacíos, se

enviará una notificación por efecto de la validación. También, se disponen de atributos de validación como:

- [DataType] para especificar que la propiedad contraseña también es del tipo de dato Password;
- [Display(Name)] para mostrar los nombres de las propiedades de las clases en las vistas y que el usuario deberá llenar; y,
- [EmailAddress] para especificar que el campo deberá tener un formato de correo electrónico.

```
17 public class AdministradorViewModel
18 {
19     public int id_Admin { set; get; }
20
21     [Required]
22     [Display(Name = "Usuario")]
23     public string Nombre_Admin { set; get; }
24
25     [Required]
26     [Display(Name = "Contraseña")]
27     [DataType(DataType.Password)]
28     public string Password { set; get; }
29
30     [Required]
31     [EmailAddress]
32     [Display(Name = "Correo")]
33     public string Correo { set; get; }
```

Figura 2.31. Codificación de la clase AdministradorViewModel

### 2.3.5 CODIFICACIÓN DE LOS CONTROLADORES Y VISTAS

En esta sección se describe la codificación de los controladores y las vistas implementadas en el prototipo.

En la Figura 2.32 se muestra los controladores y vistas que se generaron para cumplir los requerimientos del prototipo. Los controladores se identificarán con los roles de usuario del sistema para que los usuarios realicen peticiones al servidor web de acuerdo con los servicios previamente solicitados en las historias de usuario.

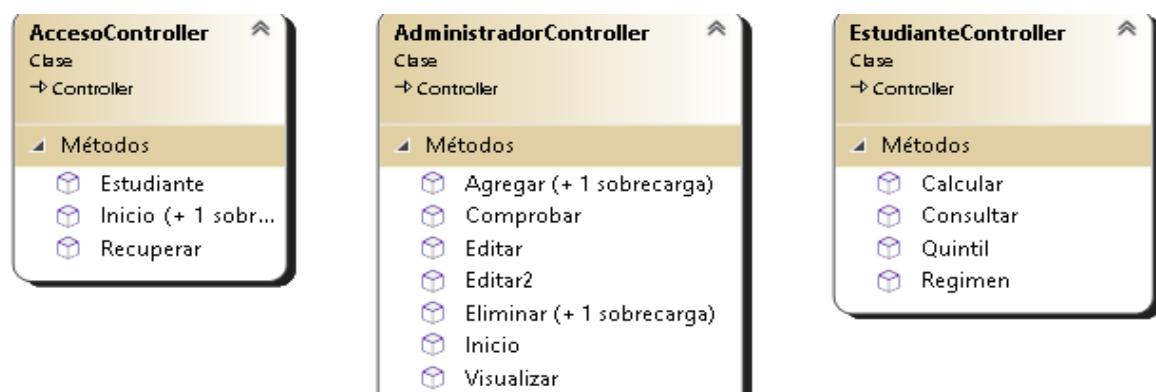


Figura 2.32. Controladores usados en el proyecto

### 2.3.5.1 Vista de Inicio

En la Figura 2.33 se muestra la vista denominada `Inicio`, la cual fue generada para seleccionar el rol de los usuarios en el prototipo. La primera opción permite escoger el rol de administrador, para gestionar la información de los parámetros base y la segunda opción permite escoger el rol de estudiante/aspirante, para consultar el valor a pagar por la matriculación en la EPN.

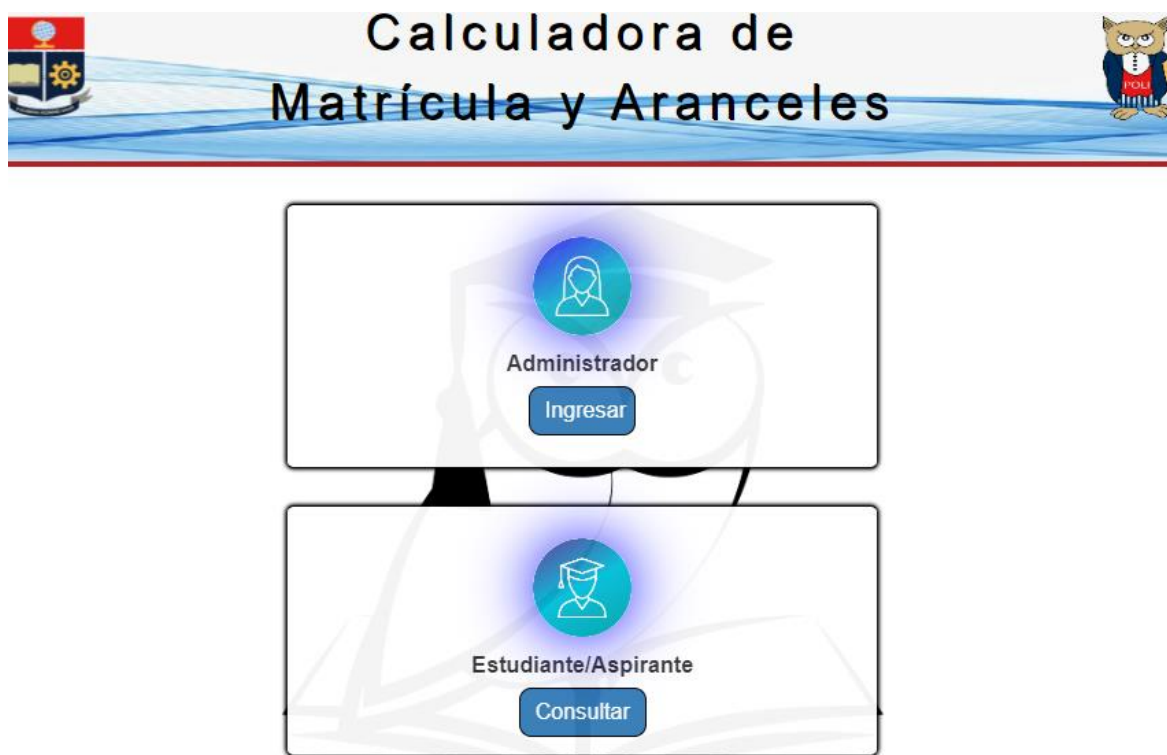


Figura 2.33. Vista Inicio

Para generar la vista `Inicio`, se utilizó un elemento denominado `modal-dialog`. Este elemento permite contener otros elementos como botones, imágenes o etiquetas de texto. En la Figura 2.34 se muestra un segmento del código de la vista para implementar un `modal-dialog`, en el que se emplea la etiqueta HTML denominada `<div>` y el atributo global `class` para su generación en la vista (línea 76). En la línea 79 se marca el inicio del contenido mediante el uso de la clase `modal-content`, el cual contiene elementos como una imagen, la cual se emplea mediante el uso de la etiqueta `<img>` y la propiedad `source` para insertar la imagen (línea 84), una etiqueta de texto el cual se emplea utilizando HTML *helper* y Razor mediante `@Html.Label` en la línea 88 y un botón para mostrar el formulario de *login* mediante el uso de la etiqueta `<a>` y la clase `btn` en la línea

91. Además, para dar el efecto visual de agrandar el `modal-dialog` se empleó código JavaScript y la propiedad `onmouseover`, con lo que el usuario al pasar el *mouse* sobre el `modal-dialog` de la vista, provoca que se aumente su tamaño y cambie de color.

```

76 <div class="modal-dialog text-center" id="modalAdmin" style="justify-content:center ;display:flex;margin:
77 30px auto;width:600px" onmouseout="disminuir('modalAdmin')" onmouseover="agrandar('modalAdmin')">
78 <div class="col-sm-12" style="padding:0px ;width:80%">
79 <div class="modal-content embed-responsive-item" id="contentAdmin" style="background-color:white;
80 border-block-color: black; opacity: 0.96; padding: 0 20px; box-shadow: 0px 0px 3px 1px #000000"
81 onmouseout="normal('contentAdmin')" onmouseover="cambio('contentAdmin')">
82 <br />
83 <div class="col-lg-12">
84 
86 </div>
87 <div>
88 @Html.Label("Administrador", new { @class = "text-center", @style = "margin-top:10px" })
89 </div>
90 <div>
91 <a class="btn btn-primary center-block" title="Accede a la información del costo por carreras
92 de la EPN." id="btnAdmin" onclick="mostrar()" style="width: 87px; border-radius: 10px;
93 border: 1px solid black">Ingresar</a>
94 </div>
95 <br />
96 <div class="form-group" id="FormAdmin" style="display:none">
97 @using (Html.BeginForm("Inicio", "Acceso", FormMethod.Post))
138 <div>
139 <a class="btn-link" role="button" style="color:darkblue;text-decoration:underline"
140 data-target="#recuperarModal" data-toggle="modal">¿Olvidaste tu contraseña?</a>
141 </div>
142 </div>
143 </div>
144 </div>
145 </div>

```

Figura 2.34. Segmento del código de la vista Inicio

Cabe recalcar que se utilizó plantillas para implementar el elemento `modal-dialog`, disponible en el sitio web de Bootstrap. Además, Bootstrap permite dar estilo a los diferentes elementos que se agreguen en la vista.

### 2.3.6 INICIO DE SESIÓN DEL ADMINISTRADOR

Cuando un usuario administrador ingresa al sistema web para el cálculo de matrícula y aranceles de la EPN, se encontrará con la vista `Inicio` y para iniciar sesión debe presionar el botón de Ingresar.

Luego se mostrará un formulario de *login*, el cual contine dos cuadros de texto para ingresar las credenciales de acceso que son: el nombre de usuario y la contraseña. Además, cuenta con un botón para visualizar el texto de la contraseña, un enlace para ocultar el formulario de *login* y un enlace para recuperar las credenciales de acceso, como se muestra en la Figura 2.35.

Calculadora de  
Matrícula y Aranceles

Administrador

Nombre de Usuario

Contraseña

Ingresar Ocultar

[¿Olvidaste tu contraseña?](#)

Figura 2.35. Formulario de *login*

### 2.3.6.1 Autenticación de usuario

Una vez el usuario administrador presiona el botón Ingresar se envían los datos ingresados en el formulario y se procede a realizar la autenticación de la información ingresada en el controlador.

En la Figura 2.36 se muestra un segmento del código de la vista que contiene el formulario que debe ser llenado con las credenciales de usuario, y esta información se enviará al servidor a través del método de acción `Inicio` del controlador `Acceso` mediante el uso del método `POST` de `HTTP` (línea 94).

En las líneas 103 y 113 se definen los campos que deben ser llenados con las credenciales de los administradores, para lo cual se emplea `HTML helpers` y `Razor`; en particular mediante el `helper` `@Html.TextBoxFor` se establece un control para ingresar el nombre de usuario, `@Html.PasswordFor` para la contraseña y también se emplea la propiedad `@Html.ValidationMessageFor` para mostrar las notificaciones de validación (líneas 97 y 122). Además, se agrega la propiedad `placeholder` para indicar al usuario la información que debe colocar en cada campo.

```

94     using (Html.BeginForm("Inicio", "Acceso", FormMethod.Post))
95     {
96         <div class="form-group">
97             @Html.ValidationMessageFor(Model => Model.Nombre_Admin, "", new { @class = "red",
98             @style = "display:inherit;font-style:italic" })
99             <span class="glyphicon glyphicon-user" style="margin-right: -121px; color: black;
100             font-size: 30px; margin-top:1px ;z-index: 2;padding-left:1px;
101             background-color: lightblue; box-shadow: 2px 0px black; border-bottom-left-radius:5px;
102             border-top-left-radius:5px"></span>
103             @Html.TextBoxFor(Model => Model.Nombre_Admin, new { @class = "form-control center-block",
104             @placeholder = "Nombre de Usuario",
105             @style = "float:right;margin-right:72px;text-align:center;padding-left:40px;z-index:1;" +
106             "margin-bottom:3px;font-style:italic;" + "border:2px solid black" })
107         </div>
108         <div class="form-group input-group">
109             <span class="glyphicon glyphicon-lock" style="margin-right: -121px; color: black; font-size:
110             30px;margin-top: 1px; z-index: 2; padding-left: 2px;background-color: lightblue; box-shadow:
111             2px 0px black; border-bottom-left-radius: 5px;border-top-left-radius: 5px "></span>
112             @Html.PasswordFor(Model => Model.Password, new { @class = "form-control center-block"
113             ,@id="contraseña" ,@placeholder = "Contraseña", @style = "float:right;margin-right:" +
114             "-22px;text-align:center;padding-left:40px;z-index:1;border-radius:" + "5px;font-style:italic;"
115             + "border:2px solid black" })
116             <span class="input-group-btn" style="padding:0px 25px">
117                 <button type="button" id="verPassword" class="btn btn-default" onclick="mostrarContraseña()"
118                 style="border-radius: 5px; border: 2px solid black">
119                     <i class="glyphicon glyphicon-eye-close" id="icono" style="font-size:17px"></i>
120                 </button>
121             </span>
122             @Html.ValidationMessageFor(Model => Model.Password, "", new { @class = "red",
123             @style = "display:table-caption;" + "font-style:italic" })
124         </div>
125         <div class="form-group">
126             @Html.TextBoxFor(Model => Model.Correo, new { @class = "form-control center-block", @placeholder
127             = "Correo", @Value = "none", @style = "float:right;margin-right:72px;text-align:center;padding-left:" +
128             "40px;z-index:1;margin-bottom:3px;font-style:italic;" + "border:2px solid black;display:none" })
129             <input type="submit" value="Ingresar" class="btn btn-primary" style="margin-right: 20px;
130             border-radius: 10px;border: 1px solid black" />
131             <a class="btn-link" id="btnHide" onclick="ocultar()" style="color: darkblue;text-decoration:underline">
132             Ocultar</a>
133         </div>
134     }

```

Figura 2.36. Sección del código del formulario *login*

Por otro lado, en la Figura 2.37 se muestra un segmento del código del controlador, en el cual se realiza el proceso de autenticación del usuario mediante la información recibida en el método de acción `Inicio` en la línea 45; en la línea 50 se verifica que la información ingresada en el formulario cumpla con las etiquetas de validación de la clase `AdministradorViewModel` que se muestra en la Figura 2.31. De la línea 55 a 57 se realiza el proceso de autenticación, con el cual se verifica la coincidencia de los datos ingresados con la información que almacenada en la base de datos; con base al resultado, en la línea 59 se verifica si existe o no el usuario para permitir el acceso la información de los parámetros base.

```
44 [HttpPost]
45 0 referencias
46 public ActionResult Inicio(Administrador admin)
47 {
48     try
49     {
50         Administrador admin2 = new Administrador();
51         if (ModelState.IsValid)
52         {
53             using (Models.dbSistemaWebEntities1 db = new Models.dbSistemaWebEntities1())
54             {
55                 var oUser = (from d in db.tblAdministrador
56                             where d.Nombre_Admin == admin.Nombre_Admin.Trim() && d.Contraseña == admin.Password.Trim()
57                             select d).FirstOrDefault();
58
59                 if (oUser == null)
60                 {
61                     ViewBag.Error = "Usuario o contraseña incorrecta ";
62                     return View();
63                 }
64                 else...
65             }
66         }
67     }
68     return RedirectToAction("Inicio", "Administrador", new {id = admin2.id_Admin});
69 }
```

Figura 2.37. Segmento del código para autenticación de usuario

### 2.3.6.2 Recuperación de credenciales por correo

El usuario que ha olvidado sus credenciales de acceso tiene la opción de recuperarlas mediante el enlace *¿Olvidaste tus credenciales?*, la cual se debe presionar para que se muestre una ventana modal, en el cual se debe ingresar el correo electrónico del usuario que desea recuperar las credenciales de acceso, como se muestra en la Figura 2.38.

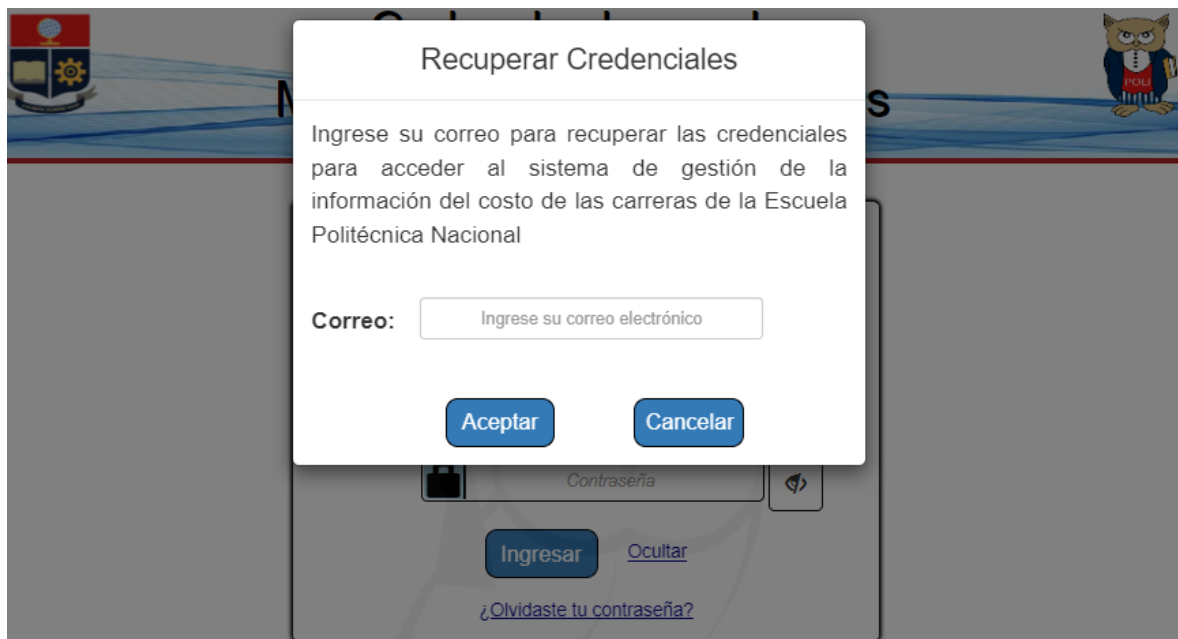


Figura 2.38. Ventana de recuperación de credenciales

Una vez el usuario presiona el botón Aceptar, la información se envía a través de la red, al controlador y en particular se usa en el método de acción `Recuperar`, como se muestra en la Figura 2.39. En las líneas 131 a 133 el correo del usuario se busca en la base de datos para encontrar coincidencias. En la línea 135 se verifica si el usuario existe con base en el resultado de la búsqueda. Seguido de la verificación del usuario, en la línea 137 se crea un objeto del tipo `MailMessage` que permitirá enviar un mensaje de correo electrónico con las credenciales del usuario. En este objeto se especifica la dirección de correo origen y destino, el asunto del mensaje y el cuerpo del mensaje que será enviado a través de un objeto del tipo `SmtpClient` (líneas 149 al 155), para lo cual en el objeto del tipo `SmtpClient` se establece el nombre del servidor (`smtp.gmail.com`) y el puerto de escucha (puerto 587).

```

122 [HttpPost]
123 public ActionResult Recuperar(string correo)
124 {
125     try
126     {
127         if (correo != null)
128         {
129             using (dbSistemaWebEntities1 db = new dbSistemaWebEntities1())
130             {
131                 var oAdmin = (from d in db.tblAdministrador
132                             where d.correo==correo
133                             select d).FirstOrDefault();
134
135                 if (oAdmin!=null)
136                 {
137                     MailMessage mensaje = new MailMessage();
138                     mensaje.From = new MailAddress("jeffersoncargua@gmail.com");
139                     mensaje.To.Add(oAdmin.correo);
140                     mensaje.CC.Add("jeffersoncargua@gmail.com");
141                     mensaje.Subject = "Sistema Web Informativo para el cálculo de matrícula y aranceles -" +
142                                     "Recuperación de Credenciales";
143                     mensaje.Body = "El presente mensaje contiene la recuperación de las credenciales para " +
144                                   "administrar el sistema web informativo para el cálculo de matrícula y aranceles. " +
145                                   "Sus credenciales son: <br />" + "<br /> <b>Nombre de Usuario:</b> " +
146                                   oAdmin.Nombre_Admin + " <br /> <b>Contraseña:</b> " + oAdmin.Contraseña;
147                     mensaje.IsBodyHtml = true;
148
149                     using (SmtpClient smtp = new SmtpClient("smtp.gmail.com", 587))
150                     {
151                         smtp.UseDefaultCredentials = true;
152                         smtp.Credentials = new System.Net.NetworkCredential("jeffersoncargua@gmail.com",
153                                     "abbzygejikkfengxu");
154                         smtp.EnableSsl = true;
155                         smtp.Send(mensaje);
156                     }
157                 }
158             }
159         }
160     }
161 }

```

Figura 2.39. Recuperación de credenciales por correo

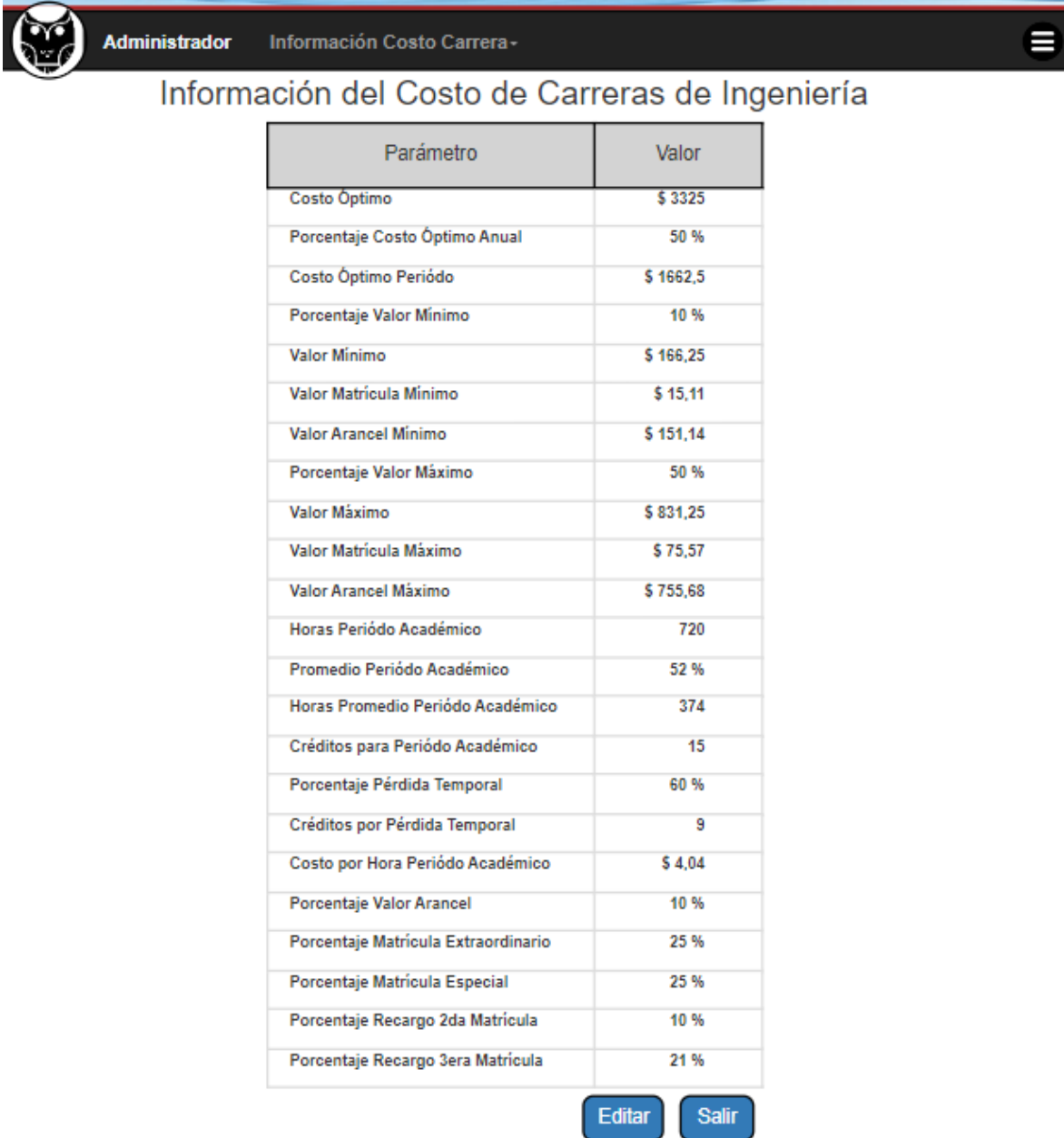
### 2.3.7 GESTIÓN DE LA INFORMACIÓN DE LOS PARÁMETROS BASE

La gestión de la información de los parámetros base para el cálculo de matrícula y aranceles de la EPN son realizadas por el administrador principal y secundario, quienes estarán en capacidad de visualizar y editar la información de dichos parámetros.



### 2.3.7.1 Visualización Parámetros Base

En la Figura 2.40 se muestra la vista que recibe el administrador después de autenticarse, la cual permite visualizar la información de los parámetros base de las carreras de Tecnología e Ingeniería para el cálculo de matrícula y aranceles de la EPN.



Parámetro	Valor
Costo Óptimo	\$ 3325
Porcentaje Costo Óptimo Anual	50 %
Costo Óptimo Período	\$ 1662,5
Porcentaje Valor Mínimo	10 %
Valor Mínimo	\$ 166,25
Valor Matrícula Mínimo	\$ 15,11
Valor Arancel Mínimo	\$ 151,14
Porcentaje Valor Máximo	50 %
Valor Máximo	\$ 831,25
Valor Matrícula Máximo	\$ 75,57
Valor Arancel Máximo	\$ 755,68
Horas Período Académico	720
Promedio Período Académico	52 %
Horas Promedio Período Académico	374
Créditos para Período Académico	15
Porcentaje Pérdida Temporal	60 %
Créditos por Pérdida Temporal	9
Costo por Hora Período Académico	\$ 4,04
Porcentaje Valor Arancel	10 %
Porcentaje Matrícula Extraordinario	25 %
Porcentaje Matrícula Especial	25 %
Porcentaje Recargo 2da Matrícula	10 %
Porcentaje Recargo 3era Matrícula	21 %

[Editar](#) [Salir](#)

**Figura 2.40.** Información de los parámetros base

Una vez que el administrador seleccione el tipo de información que desea visualizar, se enviará la petición al método de acción `Visualizar` del controlador `Administrador`, como se observa en la Figura 2.41.

En la línea 60 se recibe la información del tipo de carrera que el administrador desea visualizar, luego en la línea 64 se crea un objeto denominado `lista` del tipo `ListaParametrosBaseViewModel` que almacenará la información del modelo de Entity Framework `tblInfoCostoCarrera`. En la línea 65 se crea un objeto de la clase `dbSistemaWebEntities`, el cual permite utilizar los modelos generados por Entity Framework e interactuar con la base de datos. Este objeto se utiliza para buscar la información de los parámetros base del modelo `tblInfoCostoCarrera` a partir del tipo de carrera y del método `Find` (línea 68). En las líneas 69 a 93 la información obtenida se almacena en el objeto `lista` para mostrarlos en la vista.

```

60 public ActionResult Visualizar(int Carrera, int administrador)
61 {
62     try
63     {
64         ListaParametrosBaseViewModel lista = new ListaParametrosBaseViewModel();
65
66         using (dbSistemaWebEntities1 db = new dbSistemaWebEntities1())
67         {
68             var oTabla = db.tblInfo_Costo_Carrera.Find(Carrera);
69             lista.Id_Carrera = oTabla.id_Carrera;
70             lista.Tipo_Carrera = oTabla.Tipo_Carrera;
71             lista.CostoOptimo = (float)oTabla.CostoOptimo;
72             lista.CostoOptimoPeriodo = (float)oTabla.CostoOptimoPeriodo;
73             lista.PorcentajeValorMinimo = (float)oTabla.PorcentajeValorMinimo;
74             lista.ValorMinimo = (float)oTabla.ValorMinimo;
75             lista.ValorMatriculaMin = (float)oTabla.ValorMatriculaMin;
76             lista.ValorArancelMin = (float)oTabla.ValorArancelMin;
77             lista.PorcentajeValorMaximo = (float)oTabla.PorcentajeValorMaximo;
78             lista.ValorMaximo = (float)oTabla.ValorMaximo;
79             lista.ValorMatriculaMax = (float)oTabla.ValorMatriculaMax;
80             lista.ValorArancelMax = (float)oTabla.ValorArancelMax;
81             lista.HoraPeriodoAcademico = oTabla.HoraPeriodoAcademico;
82             lista.PorcentajePromedioAcademico = (float)oTabla.PorcentajePromedioAcademico;
83             lista.HoraPromedioPeriodoAcademico = (float)oTabla.HoraPromedioPeriodoAcademico;
84             lista.CreditosPeriodoAcademico = (float)oTabla.CreditosPeriodoAcademico;
85             lista.PorcentajePerdidaTemporal = (float)oTabla.PorcentajePerdidaTemporal;
86             lista.CreditosPerdidaTemporal = (float)oTabla.CreditosPerdidaTemporal;
87             lista.CostoHoraPeriodo = (float)oTabla.CostoHoraPeriodo;
88             lista.PorcentajeMatriculaExtraordinario = (float)oTabla.PorcentajeMatriculaExtraordinario;
89             lista.PorcentajeMatriculaEspecial = (float)oTabla.PorcentajeMatriculaEspecial;
90             lista.PorcentaRecargoSegunda = (float)oTabla.PorcentajeRecargoSegunda;
91             lista.PorcentaRecargoTercera = (float)oTabla.PorcentajeRecargoTercera;
92             lista.PorcentajeValorArancel = (float)oTabla.PorcentajeValorArancel;
93             lista.PorcentajeCostoOptimoAnual = (float)oTabla.PorcentajeCostoOptimoAnual;

```

Figura 2.41. Obtención de la información de los parámetros base

### 2.3.7.2 Edición Parámetros Base

Para la edición de la información de los parámetros base, el administrador debe presionar el botón Editar de la vista mostrada en la Figura 2.40. Seguido a esto, se presentará una vista que contiene una tabla con la información de los parámetros base a editar (ver la Figura 2.42). Se aclara que en la tabla se muestran los parámetros base editables como

costo óptimo, horas por periodo académico, créditos por periodo académico y porcentajes, esto debido a que los valores de los otros parámetros base se obtienen mediante cálculos en el controlador.

Parámetro	Valor
Costo Óptimo (\$)	3325
Porcentaje del Costo Óptimo anual (%)	50
Porcentaje Valor Mínimo (%)	10
Porcentaje Valor Máximo (%)	50
Porcentaje del Valor Arancel por Quintil (%)	5
Hora Período Académico	720
Porcentaje del Promedio Académico (%)	52
Créditos por Período Académico	15
Porcentaje Perdida Temporal (%)	60
Porcentaje Matrícula Extraordinario (%)	25
Porcentaje Matrícula Especial (%)	25
Porcentaje Recargo Segunda Matrícula (%)	10
Porcentaje Recargo Tercera Matrícula (%)	21

**Figura 2.42.** Vista para editar los parámetros base

En la Figura 2.43 se muestra un segmento de código para la edición de la información de los parámetros base. En la línea 126 el controlador recibe la información del tipo de carrera que se desea modificar. Luego se instancia un objeto denominado `parámetros` del tipo `ParametrosBaseViewModel` (línea 128), en el que se almacenará la información del modelo de `tblInfoCostoCarrera` para luego pasarlos a la vista; en el controlador se busca la información a través del `idCarrea` mediante el método `Find` (línea 131). En las líneas 132 a 146 se almacena la información obtenida en el objeto `parametros`.

```
125 [HttpPost]
    0 referencias
126 public ActionResult Editar(int id_Carrera, int administrador)
127 {
128     ParametrosBaseViewModel parametros = new ParametrosBaseViewModel();
129     using (dbSistemaWebEntities db = new dbSistemaWebEntities())
130     {
131         var oTabla = db.tblInfo_Costo_Carrera.Find(id_Carrera);
132         parametros.Id_Carrera = oTabla.id_Carrera;
133         parametros.Tipo_Carrera = oTabla.Tipo_Carrera;
134         parametros.CostoOptimo = (float)oTabla.CostoOptimo;
135         parametros.PorcentajeCostoOptimoAnual = (float)oTabla.PorcentajeCostoOptimoAnual;
136         parametros.PorcentajeValorMinimo = (float)oTabla.PorcentajeValorMinimo;
137         parametros.PorcentajeValorMaximo = (float)oTabla.PorcentajeValorMaximo;
138         parametros.PorcentajeValorArancel = (float)oTabla.PorcentajeValorArancel;
139         parametros.HoraPeriodoAcademico = (float)oTabla.HoraPeriodoAcademico;
140         parametros.PorcentajePromedioAcademico = (float)oTabla.PorcentajePromedioAcademico;
141         parametros.CreditosPeriodoAcademico = (float)oTabla.CreditosPeriodoAcademico;
142         parametros.PorcentajePerdidaTemporal = (float)oTabla.PorcentajePerdidaTemporal;
143         parametros.PorcentajeMatriculaExtraordinario = (float)oTabla.PorcentajeMatriculaExtraordinario;
144         parametros.PorcentajeMatriculaEspecial = (float)oTabla.PorcentajeMatriculaEspecial;
145         parametros.PorcentaRecargoSegunda = (float)oTabla.PorcentajeRecargoSegunda;
146         parametros.PorcentaRecargoTercera = (float)oTabla.PorcentajeRecargoTercera;
147     }
148
149     ViewBag.Id = administrador;
150     return View(parametros);
151 }
```

**Figura 2.43.** Búsqueda y retorno de la información de los parámetros base a editar

Una vez que se realizan los cambios y el administrador presione en el botón Aplicar, la información se envía al controlador. En la Figura 2.44 se muestra un segmento del código, del método de acción `Editar2` que recibe los datos de los parámetros base a actualizar (línea 169), para lo cual primero se evalúa que los datos ingresados estén correctos de acuerdo con las etiquetas de validación a través de `ModelState.IsValid` (línea 171). Seguido a esto se busca la información con base en el `IdCarrera` del objeto `parametros` y el método `Find` (línea 175). Para actualizar los valores se realizan operaciones matemáticas en base a la información enviada y la normativa para el cobro de matrículas y aranceles de la EPN (líneas 176 a 207).

```

169 public ActionResult Editar2(ParametrosBase parametros, int administrador)
170 {
171     if (ModelState.IsValid)
172     {
173         using (dbSistemaWebEntities1 db = new dbSistemaWebEntities1())
174         {
175             var oTabla = db.tblInfo_Costo_Carrera.Find(parametros.Id_Carrera);
176             oTabla.CostoOptimo = parametros.CostoOptimo;
177             oTabla.CostoOptimoPeriodo = parametros.CostoOptimo * parametros.PorcentajeCostoOptimoAnual / 100;
178             oTabla.ValorMinimo = parametros.CostoOptimo * parametros.PorcentajeValorMinimo *
179                 parametros.PorcentajeCostoOptimoAnual / 10000;
180             oTabla.ValorMatriculaMin = (parametros.CostoOptimo * parametros.PorcentajeCostoOptimoAnual *
181                 parametros.PorcentajeValorArancel * parametros.PorcentajeValorMinimo / 1000000) /
182                 (1 + parametros.PorcentajeValorArancel / 100);
183             oTabla.ValorArancelMin = (parametros.CostoOptimo * parametros.PorcentajeCostoOptimoAnual *
184                 parametros.PorcentajeValorMinimo / 10000) / (1 + parametros.PorcentajeValorArancel / 100);
185             oTabla.ValorMaximo = parametros.CostoOptimo * parametros.PorcentajeValorMaximo *
186                 parametros.PorcentajeCostoOptimoAnual / 10000;
187             oTabla.ValorMatriculaMax = (parametros.CostoOptimo * parametros.PorcentajeCostoOptimoAnual *
188                 parametros.PorcentajeValorArancel * parametros.PorcentajeValorMaximo / 1000000) /
189                 (1 + parametros.PorcentajeValorArancel / 100);
190             oTabla.ValorArancelMax = (parametros.CostoOptimo * parametros.PorcentajeCostoOptimoAnual *
191                 parametros.PorcentajeValorMaximo / 10000) / (1 + parametros.PorcentajeValorArancel / 100);
192             oTabla.HoraPeriodoAcademico = (int)parametros.HoraPeriodoAcademico;
193             oTabla.HoraPromedioPeriodoAcademico = Math.Floor(parametros.HoraPeriodoAcademico *
194                 parametros.PorcentajePromedioAcademico / 100);
195             oTabla.CreditosPeriodoAcademico = (int)parametros.CreditosPeriodoAcademico;
196             oTabla.CreditosPerdidaTemporal = Math.Ceiling(parametros.CreditosPeriodoAcademico *
197                 parametros.PorcentajePerdidaTemporal / 100);
198             oTabla.CostoHoraPeriodo = oTabla.CostoOptimoPeriodo / (oTabla.HoraPromedioPeriodoAcademico * 1.1);
199             oTabla.PorcentajeCostoOptimoAnual = parametros.PorcentajeCostoOptimoAnual;
200             oTabla.PorcentajeValorMinimo = parametros.PorcentajeValorMinimo;
201             oTabla.PorcentajeValorMaximo = parametros.PorcentajeValorMaximo;
202             oTabla.PorcentajeValorArancel = parametros.PorcentajeValorArancel;
203             oTabla.PorcentajePromedioAcademico = parametros.PorcentajePromedioAcademico;
204             oTabla.PorcentajePerdidaTemporal = parametros.PorcentajePerdidaTemporal;
205             oTabla.PorcentajeMatriculaExtraordinario = parametros.PorcentajeMatriculaExtraordinario;
206             oTabla.PorcentajeMatriculaEspecial = parametros.PorcentajeMatriculaEspecial;
207             oTabla.PorcentajeRecargoSegunda = parametros.PorcentaRecargoSegunda;

```

Figura 2.44. Actualización de la información de los parámetros base

### 2.3.8 GESTIÓN DE ADMINISTRADORES

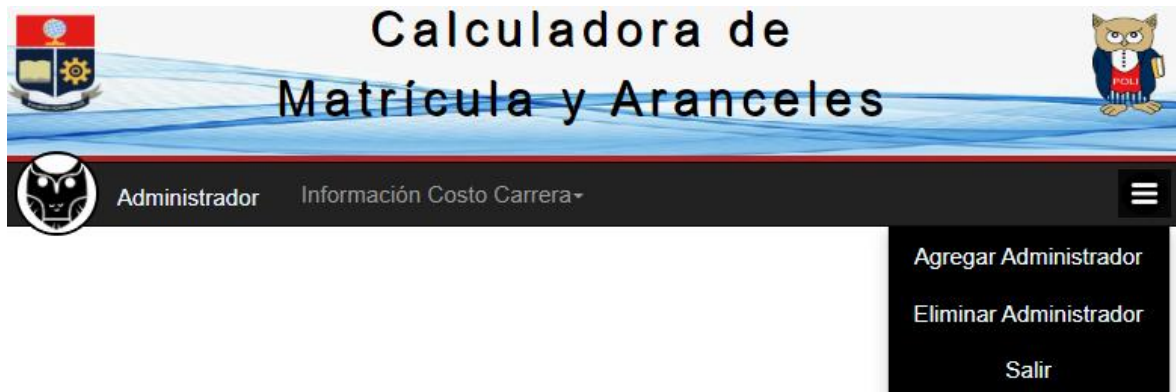
Para la gestión de usuarios, solo el administrador principal del sistema web, será capaz de agregar o eliminar administradores secundarios.

En esta sección se describen las operaciones realizadas para la gestión de usuarios, así como las vistas que se implementaron para realizar dichas operaciones.

#### 2.3.8.1 Administración de usuarios

Para la administración de usuarios se debe ingresar a la vista que se muestra en la Figura 2.45. En esta vista se tiene un botón con forma de ícono de barras ubicado al costado

derecho de la barra de navegación, el cual al ser presionado despliega las opciones de agregar o eliminar administrador.



**Figura 2.45.** Vista para gestionar usuarios

Una vez que se presiona alguna opción, se muestra una ventana modal, en el cual el administrador principal deberá introducir su contraseña para continuar con la administración de los usuarios, como se aprecia en la Figura 2.46.



**Figura 2.46.** Ventana de verificación de contraseña

En la Figura 2.47 se muestra un segmento de código de la vista, en el cual se utiliza la clase `fade` para dar un efecto de fundido en la que la ventana parece estar encima de la vista, el atributo global `aria-hidden` para ocultar la ventana modal de la vista y el atributo global `id` para especificar un identificador único para que la ventana modal aparezca en la vista cuando se presione en una de las opciones (línea 25). El contenido de ventana modal

está dividido en dos partes. La primera división consta de la clase `modal-header` que contiene la etiqueta `<h3>`, donde se define el título de la ventana modal mediante `modal-title` y la etiqueta `<a>`, además, para cerrar la ventana modal se emplea la clase `close` (líneas 31 a 39), y la segunda división consta de la clase `modal-body` que contiene un cuadro de texto para ingresar la contraseña. En la línea 41 se emplea HTML *helper* y Razor para agregar el formulario mediante `@Html.BeginForm` y el método POST de HTTP para enviar los datos al controlador `Administrador`, en el cual se verificará la contraseña.

```

25 <div class="modal fade" id="agregarModal" tabindex="-1" role="dialog"
26     aria-labelledby="agregarModalLabel" aria-hidden="true" style="display:
27     none;padding-right:0px;min-width:1024px">
28     <div class="modal-dialog" role="document" style="display: flex;justify-content:
29     center;width:60%">
30         <div class="modal-content" style="text-align:center;width:50%">
31             <div class="modal-header">
32                 <div style="float:left;width:90%;text-align:center">
33                     <h3 class="modal-title" id="agregarModalLabel">
34                         Agregar Administrador</h3>
35                 </div>
36                 <a href="#" class="close" data-dismiss="modal" aria-label="Close">
37                     <span aria-hidden="true">x</span>
38                 </a>
39             </div>
40             <div class="modal-body form-group">
41                 @using (Html.BeginForm("Comprobar", "Administrador", FormMethod.Post))
42                 {
43                     @Html.Label("Ingresa la clave para agregar un nuevo administrador")
44                     <br />
45                     <br />
46                     <label style="float:left;margin-top:6px">Clave: </label>
47                     <input type="password" class="form-control" name="clave" placeholder
48                         ="Ingresa la Clave" value="" style="float:left;margin-left:
49                         10px;text-align:center;width:75%" />
50                     <input type="text" name="tipo" value="1" style="display:none" />
51                     <input name="admin" value="@Model2" style="display:none" />
52                     <br />
53                     <br />
54                     <br />
55                     <button type="submit" class="btn btn-primary" style="border-radius: 10px;
56                         border: 1px solid black">Acentar</button>

```

**Figura 2.47.** Ventana para ingresar contraseña de administrador principal

En la Figura 2.48 se muestra un segmento de código del controlador, en particular del método de acción `Comprobar` (línea 281). En las líneas 287 al 289 se busca la coincidencia de la contraseña del administrador principal con la contraseña ingresada y en la línea 290 se comprueba el resultado de la búsqueda. En la línea 297 se verifica la operación seleccionada por el administrador principal para luego redirigirlo a la vista correspondiente mediante el método `RedirectToAction( )` (línea 299 o 301).

```
281 public ActionResult Comprobar(string clave, int tipo, int admin)
282 {
283     try
284     {
285         using (Models.dbSistemaWebEntities db = new Models.dbSistemaWebEntities())
286         {
287             var oUser = (from d in db.tblAdministrador
288                         where clave==d.Contraseña && d.Nombre_Admin=="Admin"
289                         select d).FirstOrDefault();
290             if (oUser == null)
291             {
292                 if (tipo==1)
293                 {
294                     return RedirectToAction("Agregar", "Administrador", new { @id= admin });
295                 }
296                 return RedirectToAction("Eliminar", "Administrador", new { @id = admin });
297             }
298         }
299     }
300 }
301
```

Figura 2.48. Segmento de código para comprobar contraseña

### 2.3.8.2 Creación de un nuevo administrador

En la Figura 2.49 se muestra la vista que se implementó para agregar un nuevo administrador. Esta vista contiene un formulario que debe llenarse con los datos del nuevo administrador, la cual dispone de campos para agregar el nombre de usuario, la contraseña, datos que servirán como credenciales para el acceso a la información de los parámetros base y también se agrega el correo electrónico que sirve para la recuperación de las credenciales.

Calculadora de Matrícula y Aranceles

Agregar Nuevo Administrador

**Ingrese las credenciales**

Nombre de Usuario:

Contraseña:

**Ingrese el correo del nuevo usuario**

Correo:

Figura 2.49. Vista para agregar un nuevo administrador

En la Figura 2.50 se muestra un segmento del código utilizado para formulario de la vista. La vista está dividida en tres secciones, en la que se emplea la etiqueta <div> y la clase



form-group, para agrupar la etiqueta y el cuadro de texto en cada sección para ingresar los datos del nuevo administrador (líneas 62 a 69, 72 a 79 y 83 a 91).

En estas secciones se hace uso de los HTML *helpers* y Razor para mostrar y almacenar la información del modelo de vista `AdministradorViewModel` mediante las propiedades `@Html.Label`, `@Html.TextBoxFor` y `@Html.ValidationMessageFor`.

```
58 using (Html.BeginForm("Agregar", "Administrador", FormMethod.Post, new { @style="margin-left:50px"}))
59 {
60     <h3>@Html.Label("Ingrese las credenciales")</h3>
61     <br />
62     <div class="form-group">
63         @Html.Label("Nombre de Usuario:", new { @style = "float:left" })
64         @Html.TextBoxFor(Model => Model.Nombre_Admin, new { @class = "form-control center-block",
65             @placeholder = "Nombre de Usuario", @id="nombre", @style = "float:left;margin-top:-6px;margin-left:" +
66             "10px;margin-right:10px;text-align:center" })
67         @Html.ValidationMessageFor(Model => Model.Nombre_Admin, "", new { @class = "red", @style =
68             "float:right;display:contents" })
69     </div>
70     <br />
71     <br />
72     <div class="form-group">
73         @Html.Label("Contraseña:", new { @style = "float:left" })
74         @Html.TextBoxFor(Model => Model.Password, new { @class = "form-control center-block", @placeholder =
75             "Contraseña", @id = "contraseña", @style = "float:left;margin-top:-6px;margin-left:64px;margin-right:" +
76             "10px;text-align:center" })
77         @Html.ValidationMessageFor(Model => Model.Password, "", new { @class = "red", @style = "float:right;" +
78             "display:contents" })
79     </div>
80     <br />
81     <h3>@Html.Label("Ingrese el correo del nuevo usuario")</h3>
82     <br />
83     <div class="form-group">
84         @Html.Label("Correo:", new { @style = "float:left" })
85         @Html.TextBoxFor(Model => Model.Correo, new { @class = "form-control center-block", @placeholder =
86             "Correo", @id = "correo", @style = "float:left;margin-top:-6px;margin-left:94px;margin-right:10px;" +
87             "text-align:center" })
88         @Html.ValidationMessageFor(Model => Model.Correo, "", new { @class = "red", @style = "float:right;" +
89             "display:contents" })
90         <input name="admin" value="@admin" style="display:none" />
91     </div>
```

**Figura 2.50.** Segmento de código del formulario de registro

Cuando el administrador presione el botón Aceptar, la información ingresada en el formulario se enviará al controlador `Administrador` para añadir un nuevo registro en la base de datos.

En la Figura 2.51 se muestra un segmento del código para agregar un nuevo administrador, mediante el método de acción `Agregar` (línea 353). En línea 357 se evalúa que la información cumpla con las validaciones del modelo `AdministradorViewModel`; en las líneas 361 a 363 se busca coincidencias con la información almacenada en la base de datos, con el propósito de evitar duplicaciones, luego se valida el resultado obtenido en la línea 364. Seguido a esto, en las líneas 371 a 388 se prepara un mensaje para enviarlo por correo electrónico al nuevo administrador, en el que se realiza el mismo procedimiento que se mencionó en la recuperación de credenciales.

A continuación, en la línea 390 se instancia un objeto denominado `oAdministrador` del tipo `tblAdministrador`, clase generada por Entity Framework. Luego se pasan los valores del objeto `AdministradorViewModel` al objeto `oAdministrador`, mismo que servirá para agregar un nuevo registro en la base de datos (líneas 391 a 393). En la línea 394 se agrega el nuevo administrador, mediante el método `Add()` se envían los datos almacenados en el objeto `oAdministrador` a la clase `tblAdministrador` generada por Entity Framework. Finalmente se emplea el método `SaveChanges()` para guardar el nuevo registro en la base de datos.

```

353 public ActionResult Agregar(Administrador administrador,int admin)
354 {
355     try
356     {
357         if (ModelState.IsValid)
358         {
359             using (dbSistemaWebEntities1 db = new dbSistemaWebEntities1())
360             {
361                 var oUser = (from d in db.tblAdministrador
362                             where d.correo == administrador.Correo
363                             select d).FirstOrDefault();
364                 if (oUser != null)
365                 {
366                     return RedirectToAction("Index");
367                 }
368                 MailMessage correo = new MailMessage();
369                 correo.From = new MailAddress("jeffersoncargua@gmail.com");
370                 correo.To.Add(administrador.Correo);
371                 correo.Subject = "Sistema Web Informativo para el cálculo de matrícula y aranceles - " +
372                             "Credenciales de Administrador";
373                 correo.Body = "El presente mensaje contiene las credenciales para administrar el sistema " +
374                             "web informativo para el cálculo de matrícula y aranceles de la Escuela Politécnica " +
375                             "Nacional. Sus credenciales son: <br />" + "<br /> <b>Nombre de Usuario:</b> " +
376                             + administrador.Nombre_Admin + " <br /> <b>Contraseña:</b> " + administrador.Password;
377                 correo.IsBodyHtml = true;
378                 using (SmtpClient smtp = new SmtpClient("smtp.gmail.com", 587))
379                 {
380                     smtp.Credentials = new System.Net.NetworkCredential("jeffersoncargua@gmail.com",
381                                 "abbzygejlkfengxu");
382                     smtp.EnableSsl = true;
383                     smtp.Send(correo);
384                 }
385                 var oAdministrador = new tblAdministrador();
386                 oAdministrador.Nombre_Admin = administrador.Nombre_Admin;
387                 oAdministrador.Contraseña = administrador.Password;
388                 oAdministrador.correo = administrador.Correo;
389                 db.tblAdministrador.Add(oAdministrador);
390                 db.SaveChanges();
391             }
392         }
393     }
394 }
395

```

Figura 2.51. Segmento de código para agregar un nuevo administrador

### 2.3.8.3 Eliminación de un administrador

Para la eliminación de un administrador secundario se presiona el ícono con forma de basurero, el mismo que está ubicado en el lado derecho de cada fila de la tabla en la que se muestra la información de los administradores registrados hasta el momento, como se muestra en la Figura 2.52. Al presionar el ícono para eliminar un administrador se muestra una ventana modal, en la cual se pregunta si está seguro de realizar la operación de eliminación.



**Figura 2.52.** Vista para eliminar administradores

En la Figura 2.53 se muestra un segmento de código de la vista para implementar una tabla con la información de los administradores registrados en la base de datos. La tabla se implementa mediante la etiqueta `<table>` (línea 38). En la línea 39 se emplea la etiqueta `<tr>` para establecer las columnas necesarias en las que se presentará la información de los usuarios, y mediante la etiqueta `<th>` se coloque el texto de cabecera que tendrá cada columna en la tabla; en la línea 40 se define la columna para identificar el nombre de usuario y en la línea 42 se define la columna para identificar el correo del usuario.

Para mostrar la información de los usuarios se empleó Razor, y a través del método `@foreach` se realiza iteraciones para listar los usuarios en la tabla (línea 47). La información de los usuarios se muestra mediante el uso de la etiqueta `<td>` y del objeto `oElemento`; en particular, en la línea 50 se muestra el nombre de usuario, en la línea 51 se muestra el correo del usuario y en la línea 52 se muestra el icono con forma de basurero.

```

38 <table class="table table-bordered table-hover" style="width:65%">
39 <tr>
40 <th scope="col" style="text-align: center; border: 1px solid #000; background-color
: lightgray">Nombre de Usuario</th>
41
42 <th scope="col" style="text-align: center; border: 1px solid #000; background-color
: lightgray">Correo</th>
43
44 <th scope="col" style="text-align: center; border: 1px solid #000; background-color
: lightgray"></th>
45 </tr>
46
47 @foreach (var oElemento in lista)
48 {
49 <tr>
50 <td style="text-align:center;padding-top:16px">@oElemento.Nombre_Admin</td>
51 <td style="text-align:center;padding-top:16px">@oElemento.Correo</td>
52 <td style="text-align:center"><a id="registro" name="registro" role="button"
data-toggle="modal" data-target="#eliminarModal" onclick
="enviar(@oElemento.id_Admin)" title="Eliminar registro">
53 </a></td>
54 </tr>
55 }
56 </table>
57
58

```

**Figura 2.53.** Segmento del código para mostrar información de los usuarios

Una vez se confirma la eliminación se envía el registro al método de acción `Eliminar` (ver Figura 2.54), en el que el controlador `Administrador` se recibe el `id_Admin` para realizar la eliminación del registro (línea 478), luego en la línea 484 se crea un objeto `oAdmin`, en el cual se almacenará la información del usuario a eliminar mediante el método `Find`.

Cada vez que se elimina un registro, se envía un mensaje por correo electrónico al usuario que ha sido removido, mediante el cual se agradece por su colaboración (líneas 486 a 503). Seguido a esto en la línea 505 se procede a eliminar el registro mediante el método `Remove()` y finalmente con la línea 506 se guardan los cambios en la base de datos con el método `SaveChanges()`.

```
478 public ActionResult Eliminar(string id_Admin, int admin)
479 {
480     try
481     {
482         using (dbSistemaWebEntities1 db = new dbSistemaWebEntities1())
483         {
484             var oAdmin = db.tblAdministrador.Find(Convert.ToInt32(id_Admin));
485             MailMessage correo = new MailMessage();
486             correo.From = new MailAddress("jeffersoncargua@gmail.com");
487             correo.To.Add(oAdmin.correo);
488             correo.Subject = "Sistema Web Informativo para el cálculo de matrícula y aranceles - " +
489                 "Notificación Urgente";
490             correo.Body = "Estimado Colaborador, " + "<br /><br />En el presente mensaje se le notifica " +
491                 "que su usuario " +
492                 "a sido eliminado del sistema web para cálculo de matrícula y aranceles. " + "El equipo" +
493                 "le agradece por " +
494                 "sus servicios prestados para el mantenimiento del sistema web informativo para el cobro" +
495                 "de matrícula y " +
496                 "aranceles de la Escuela Politécnica Nacional." + "<br /><br />Tenga un excelente día.";
497             correo.IsBodyHtml = true;
498             using (SmtpClient smtp = new SmtpClient("smtp.gmail.com", 587))
499             {
500                 smtp.Credentials = new System.Net.NetworkCredential("jeffersoncargua@gmail.com",
501                     "abbzygejikfengxu");
502                 smtp.EnableSsl = true;
503                 smtp.Send(correo);
504             }
505             db.tblAdministrador.Remove(oAdmin);
506             db.SaveChanges();

```

Figura 2.54. Segmento de código para eliminar administrador

### 2.3.9 CÁLCULO DE MATRÍCULA Y ARANCELES

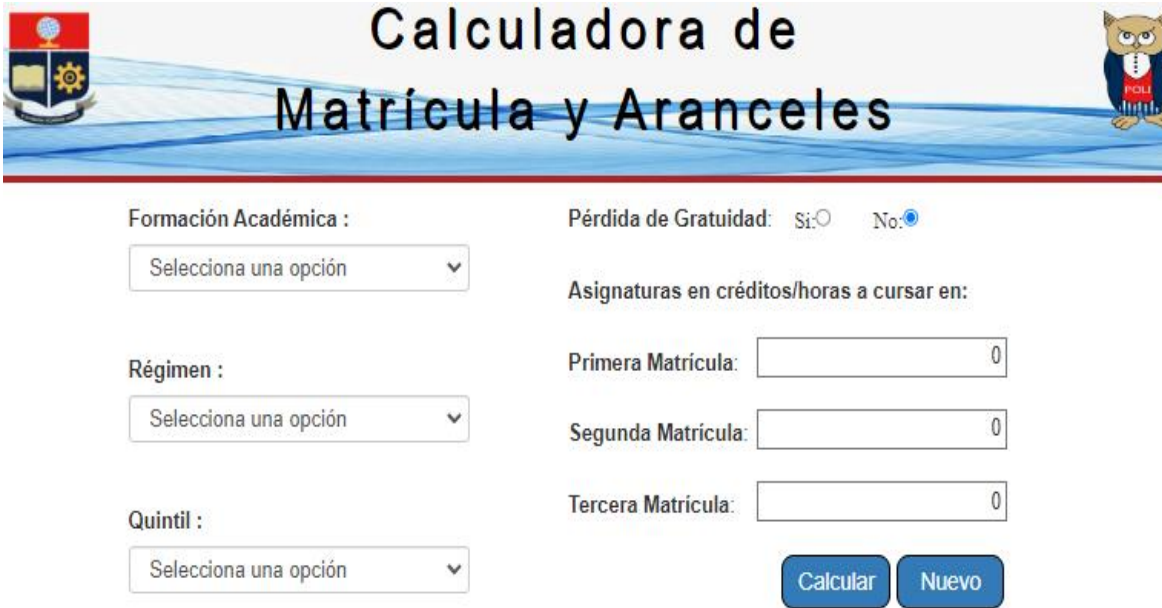
Para el cálculo de matrícula y aranceles en la EPN, el estudiante/aspirante será capaz de consultar y calcular los posibles valores a cancelar por la matriculación.

En esta sección se explica el procedimiento que el estudiante/aspirante de la EPN deberá realizar en las vistas implementadas para conocer los valores a cancelar por la matriculación, de acuerdo a la información que los mismos proporcionen.

Para acceder el usuario deberá escoger el rol de Estudiante/Aspirante en la vista presentada en la Figura 2.33 y presionar en el botón de Consultar para acceder a la vista para ingresar la información para el cálculo.

### 2.3.9.1 Consulta del valor a pagar

Una vez que el estudiante/aspirante ha presionado el botón Consultar, se presentará una vista, la cual contiene un formulario que el usuario debe llenar. En la Figura 2.55 se muestra la parte superior de la vista, donde el usuario debe introducir la información que se le solicita de acuerdo con su formación académica, régimen, quintil, situación de gratuidad y finalmente la cantidad de horas/créditos que va a cursar durante el periodo académico. Además, se incluye un botón para realizar el cálculo del valor a cancelar y otro botón para realizar una nueva consulta.



The screenshot shows the top portion of a web application titled "Calculadora de Matrícula y Aranceles". The header features a university crest on the left and a cartoon owl mascot on the right. The form contains several input fields and buttons:

- Formación Académica :** A dropdown menu with the text "Selecciona una opción".
- Régimen :** A dropdown menu with the text "Selecciona una opción".
- Quintil :** A dropdown menu with the text "Selecciona una opción".
- Pérdida de Gratuidad:** Radio buttons for "Si:" (unselected) and "No:" (selected).
- Asignaturas en créditos/horas a cursar en:** Three input fields labeled "Primera Matricula:", "Segunda Matricula:", and "Tercera Matricula:", each containing the number "0".
- Buttons:** Two blue buttons labeled "Calcular" and "Nuevo".

**Figura 2.55.** Parte superior de la vista para ingresar datos del estudiante/aspirante

La parte inferior de la vista se muestra en la Figura 2.56, en la cual se presenta información que el estudiante/aspirante puede utilizar para consultar la información del quintil socioeconómico al que pertenece e información adicional para conocer los tipos de pérdida de gratuidad de acuerdo a la condición en la que se encuentre el estudiante/aspirante. Además, se incluye un enlace para redirigir al usuario al sistema SAEw de la EPN para poder buscar la información que necesita para realizar el cálculo.

### **Aviso:**

Si no recuerdas el quintil socioeconómico al que perteneces, sigue las siguientes instrucciones:

- Ingresa al SAEW con el siguiente enlace: <https://saew.epn.edu.ec/>
- Accede al sistema con tus credenciales.
- Dirígete al módulo de Información Estudiantil.
- En la pestaña Matriculación selecciona Información de Pagos.
- En Información de Costo de Matrícula selecciona tu carrera y observarás la información del Quintil al que perteneces.

### **Información Adicional:**

- Si reprobaste asignaturas en el período académico previo, tu situación de gratuidad es de Pérdida Parcial de Gratuidad.
- En el caso de las carreras del nuevo régimen, si no te inscribiste en el 60% de los créditos del nivel referencial(15 créditos), tu situación de gratuidad es de Pérdida Temporal de Gratuidad.
- Si acumulaste más del 30% de créditos u horas de asignaturas fallidas, respecto a los créditos u horas totales de tu carrera, tu situación de gratuidad es de Pérdida Definitiva de Gratuidad.
- Si deseas conocer el % de Reprobación de la Carrera visita el [SAEW](#) y revisa tu Currículum Académico.
- Para efectos de cálculo, considera que 1 crédito equivale a 16 horas.

Salir

**Figura 2.56.** Parte inferior de la vista para ingresar datos del estudiante/aspirante

Cuando el estudiante/aspirante presione el botón Calcular, la información se envía al servidor. En la Figura 2.57 se muestra un segmento del código para el calcular el valor a cancelar por el estudiante/aspirante, en el cual se recibe los datos enviados desde la vista a través del objeto del modelo de vista `EstudianteViewModel` (línea 89). Así mismo, se verifica que se cumplan con las etiquetas de validación definidos en los atributos del modelo `EstudianteViewModel` (línea 94). En la línea 96 se verifica el tipo de régimen que selecciono el usuario, donde el valor de 1 corresponde al régimen créditos y el valor de 2 corresponde al régimen horas. Esta validación permite realizar la conversión de créditos a horas, con la finalidad de realizar los cálculos correspondientes del valor a pagar por la matriculación, en el caso de que se haya seleccionado el régimen de créditos (líneas 98 a 100). También, se valida el caso en el que se haya seleccionado el régimen de horas, donde se verifica que se cumpla con los valores adecuados para realizar los cálculos del valor a pagar por la matriculación (línea 105).

```

89 public ActionResult Calcular(EstudianteViewModel estudiante)
90 {
91     try
92     {
93         DetalleViewModel detalles = new DetalleViewModel();
94         if (ModelState.IsValid)
95         {
96             if (estudiante.Tipo_Regimen == 1)
97             {
98                 estudiante.Primer_Matricula = 16 * estudiante.Primer_Matricula;
99                 estudiante.Segunda_Matricula = 16 * estudiante.Segunda_Matricula;
100                estudiante.Tercera_Matricula = 16 * estudiante.Tercera_Matricula;
101            }
102            if(estudiante.Tipo_Regimen ==2 && (estudiante.Primer_Matricula+estudiante.Segunda_Matricula+
103                estudiante.Tercera_Matricula)>0)
104            {
105                if ( (estudiante.Primer_Matricula+estudiante.Segunda_Matricula+estudiante.Tercera_Matricula)
106                    %16 != 0)
107                {
108                    List<ListaCarrera> listal = null;
109                    using (dbSistemaWebEntities db = new dbSistemaWebEntities())

```

**Figura 2.57.** Obtención del régimen para el cálculo de matrícula y aranceles

Seguido a esto, en la Figura 2.58 se muestran las líneas de código utilizadas para evaluar los datos ingresados por el estudiante/aspirante para realizar el cálculo. En la línea 135 se crea un objeto `oEstudiante`, el cual almacenará la información de los parámetros base para calcular el valor que el estudiante/aspirante deberá pagar por la matriculación, y mediante el `IdCarrera` y el método `Find` se busca la información en la base de datos.

Para evaluar la pérdida de gratuidad del estudiante y realizar los cálculos correspondientes al valor a pagar por la matriculación, se hace uso de condicionales para determinar el tipo de pérdida de acuerdo con la información ingresada por el usuario (líneas 153, 168, 186 y 202).

```

135 var oEstudiante = db.tblInfo_Costo_Carrera.Find(estudiante.Id_Carrera);
136 detalles.Costo_Hora_Socioeconomico = (float)(oEstudiante.CostoHoraPeriodo * estudiante.Quintil * 0.1);
137 if ((estudiante.Primer_Matricula + estudiante.Segunda_Matricula + estudiante.Tercera_Matricula) == 0)
151 else
152 {
153     if (estudiante.Perdida_Gratuidad == true)
166     else
167     {
168         if ((estudiante.Segunda_Matricula + estudiante.Tercera_Matricula) > 0 && ((estudiante.Primer_Matricula
169             + estudiante.Segunda_Matricula + estudiante.Tercera_Matricula)/16 <= oEstudiante.CreditosPerdidaTemporal
170             || (estudiante.Primer_Matricula + estudiante.Segunda_Matricula + estudiante.Tercera_Matricula) <=
171             oEstudiante.CreditosPerdidaTemporal*16))
184     else
185     {
186         if ((estudiante.Segunda_Matricula + estudiante.Tercera_Matricula) > 0)
200     else
201     {
202         if (((estudiante.Primer_Matricula + estudiante.Segunda_Matricula + estudiante.Tercera_Matricula)
203             / 16) <= oEstudiante.CreditosPerdidaTemporal || ((estudiante.Primer_Matricula +
204             estudiante.Segunda_Matricula + estudiante.Tercera_Matricula) <= oEstudiante.CreditosPerdidaTemporal
205             * 16))
218     else

```

**Figura 2.58.** Evaluación del tipo de pérdida de gratuidad

En la Figura 2.59 se muestra un segmento de código para realizar el cálculo, mediante el cual se crea un objeto `detalles` del tipo `DetalleViewModel` para almacenar los cálculos realizados y después mostrarlos en la vista. Luego de verificar el tipo de gratuidad, se realizan los cálculos de acuerdo a la información de los parámetros base obtenidos en la línea 135. En las líneas 155 a 173 se realizan operaciones matemáticas necesarias para calcular el valor que el estudiante deberá cancelar por la matriculación.

```

133 using (dbSistemaWebEntities db = new dbSistemaWebEntities())
134 {
135     var oEstudiante = db.tblInfo_Costo_Carrera.Find(estudiante.Id_Carrera);
136     detalles.Costo_Hora_Socioeconomico = (float)(oEstudiante.CostoHoraPeriodo * estudiante.Quintil * 0.1);
137     if ((estudiante.Primer_Matricula + estudiante.Segunda_Matricula + estudiante.Tercera_Matricula) == 0)
151     else
152     {
153         if (estudiante.Perdida_Gratuidad == true)
154         {
155             detalles.Tipo_perdida = "Pérdida Definitiva";
156             detalles.Valor_Matricula_Ordinario = decimal.Round((decimal)(oEstudiante.ValorMatriculaMin*
157                 estudiante.Quintil), 2);
158             detalles.Valor_Arancel = decimal.Round((decimal)((estudiante.Primer_Matricula +
159                 estudiante.Segunda_Matricula + estudiante.Tercera_Matricula) *
160                 (detalles.Costo_Hora_Socioeconomico)), 2);
161             detalles.Recargo_segunda = decimal.Round((decimal)(estudiante.Segunda_Matricula *
162                 detalles.Costo_Hora_Socioeconomico * oEstudiante.PorcentajeRecargoSegunda / 100), 2);
163             detalles.Recargo_Tercera = decimal.Round((decimal)(estudiante.Tercera_Matricula *
164                 detalles.Costo_Hora_Socioeconomico * oEstudiante.PorcentajeRecargoTercera / 100), 2);
165             detalles.Bancario = 1;
166             detalles.Bancario_aux = 1;
167             detalles.Total_Ordinario = (decimal)(detalles.Valor_Matricula_Ordinario + detalles.Valor_Arancel
168                 + detalles.Recargo_segunda + detalles.Recargo_Tercera + detalles.Bancario);
169             detalles.Recargo_Matricula_Extraordinaria = decimal.Round((decimal)(detalles.Valor_Matricula_Ordinario
170                 * (decimal)oEstudiante.PorcentajeMatriculaExtraordinario / 100), 2);
171             detalles.Total_Extraordinario = (decimal)(detalles.Valor_Matricula_Ordinario +
172                 detalles.Recargo_Matricula_Extraordinaria + detalles.Valor_Arancel + detalles.Recargo_segunda
173                 + detalles.Recargo_Tercera + detalles.Bancario_aux);
174         }
    }
}

```

**Figura 2.59.** Cálculo del valor a cancelar

En la Figura 2.60 se muestra un segmento del código de la vista, mediante el cual se presenta la implementación de un menú desplegable para mostrar las opciones que tendrá que seleccionar el usuario, mediante el uso de HTML *helpers* y Razor haciendo uso de `@Html.DropDownListFor` (línea 29). Este menú desplegable se usa para mostrar la información sobre la formación académica, régimen y quintil que el usuario deberá seleccionar para realizar el cálculo (línea 29, 36 y 43).



```

27 <div style="width:30%;padding-right:20px;margin-right:40px">
28   @Html.Label("Formación Académica :")
29   @Html.DropDownListFor(Model => Model.Id_Carrera, items, "Selecciona una opción",
30     new { @class = "form-control input", @name = "carrera" })
31   @Html.ValidationMessageFor(Model => Model.Id_Carrera, "", new { @class = "red",
32     @style = "color:red;font-style:italic" })
33   <br />
34   <br />
35   @Html.Label("Régimen :")
36   @Html.DropDownListFor(Model => Model.Tipo_Regimen, (List<SelectListItem>)ViewBag.ListaRegimen,
37     "Selecciona una opción", new { @class = "form-control input", @name = "regimen" })
38   @Html.ValidationMessageFor(Model => Model.Tipo_Regimen, "", new { @class = "red", @style =
39     "color:red;font-style:italic" })
40   <br />
41   <br />
42   @Html.Label("Quintil :")
43   @Html.DropDownListFor(Model => Model.Quintil, (List<SelectListItem>)ViewBag.ListaQuintil,
44     "Selecciona una opción", new { @class = "form-control input", @name = "quintil" })
45   @Html.ValidationMessageFor(Model => Model.Quintil, "", new { @class = "red", @style =
46     "color:red;font-style:italic" })
47   <br />

```

Figura 2.60. Menú desplegable para selección de información

### 2.3.9.2 Información del Pago de Matrícula

En la Figura 2.61 se muestra una vista con toda la información del cálculo realizado en el controlador `Estudiante`, en el cual el usuario podrá observar el detalle de los cálculos de: matrícula, arancel, recargo en segunda o tercera matrícula y el valor total a pagar. Este detalle está dividido en Matrícula Ordinaria y Matrícula Extraordinaria/Especial. En el caso de Matrícula Extraordinaria/Especial se agrega un cálculo adicional que es el valor del recargo por matrícula extraordinaria o especial.

Información del Pago de Matrícula																																									
Resultados del Cálculo																																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">Matrícula Ordinaria</th> </tr> <tr> <td colspan="2">Condición de Gratuidad: <b>Pérdida Parcial</b></td> </tr> <tr> <th style="text-align: left;">Detalle</th> <th style="text-align: right;">Valor</th> </tr> </thead> <tbody> <tr> <td>Valor Matrícula</td> <td style="text-align: right;">15,11</td> </tr> <tr> <td>Valor Arancel</td> <td style="text-align: right;">32,33</td> </tr> <tr> <td>Recargo 2da Matrícula</td> <td style="text-align: right;">3,23</td> </tr> <tr> <td>Matrícula</td> <td style="text-align: right;">0</td> </tr> <tr> <td>Recargo 3era Matrícula</td> <td style="text-align: right;">1</td> </tr> <tr> <td>Bancario</td> <td></td> </tr> <tr> <td>Total a pagar: \$</td> <td style="text-align: right; border: 1px solid red;">51,67</td> </tr> </tbody> </table>	Matrícula Ordinaria		Condición de Gratuidad: <b>Pérdida Parcial</b>		Detalle	Valor	Valor Matrícula	15,11	Valor Arancel	32,33	Recargo 2da Matrícula	3,23	Matrícula	0	Recargo 3era Matrícula	1	Bancario		Total a pagar: \$	51,67	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">Matrícula Extraordinaria/Especial</th> </tr> <tr> <td colspan="2">Condición de Gratuidad: <b>Pérdida Parcial</b></td> </tr> <tr> <th style="text-align: left;">Detalle</th> <th style="text-align: right;">Valor</th> </tr> </thead> <tbody> <tr> <td>Valor Matrícula</td> <td style="text-align: right;">15,11</td> </tr> <tr> <td>Recargo Matrícula Extraordinaria</td> <td style="text-align: right;">3,78</td> </tr> <tr> <td>Valor Arancel</td> <td style="text-align: right;">32,33</td> </tr> <tr> <td>Recargo 2da Matrícula</td> <td style="text-align: right;">0</td> </tr> <tr> <td>Recargo 3era Matrícula</td> <td style="text-align: right;">1</td> </tr> <tr> <td>Bancario</td> <td></td> </tr> <tr> <td>Total a pagar: \$</td> <td style="text-align: right; border: 1px solid red;">55,45</td> </tr> </tbody> </table>	Matrícula Extraordinaria/Especial		Condición de Gratuidad: <b>Pérdida Parcial</b>		Detalle	Valor	Valor Matrícula	15,11	Recargo Matrícula Extraordinaria	3,78	Valor Arancel	32,33	Recargo 2da Matrícula	0	Recargo 3era Matrícula	1	Bancario		Total a pagar: \$	55,45
Matrícula Ordinaria																																									
Condición de Gratuidad: <b>Pérdida Parcial</b>																																									
Detalle	Valor																																								
Valor Matrícula	15,11																																								
Valor Arancel	32,33																																								
Recargo 2da Matrícula	3,23																																								
Matrícula	0																																								
Recargo 3era Matrícula	1																																								
Bancario																																									
Total a pagar: \$	51,67																																								
Matrícula Extraordinaria/Especial																																									
Condición de Gratuidad: <b>Pérdida Parcial</b>																																									
Detalle	Valor																																								
Valor Matrícula	15,11																																								
Recargo Matrícula Extraordinaria	3,78																																								
Valor Arancel	32,33																																								
Recargo 2da Matrícula	0																																								
Recargo 3era Matrícula	1																																								
Bancario																																									
Total a pagar: \$	55,45																																								
<input type="button" value="Salir"/>																																									

Figura 2.61. Vista con la información del pago por matrícula

En la Figura 2.62 se muestra un segmento del código de la vista, mediante el cual se presenta la información de los cálculos realizados del valor a pagar por la matriculación. En esta estructura se utiliza un esquema similar al de la Figura 2.53, con la diferencia de que se utilizan elementos adicionales como la etiqueta `<dl>` para listar los nombres de los valores calculados (línea 34 y 43). Los nombres de estos valores están definidos mediante la etiqueta `<dt>` junto a la etiqueta `<label>`, las cuales se usan para listar y mostrar el nombre de los valores y presentarlos en la tabla (líneas 35 a 39).

Para mostrar los valores calculados, en las líneas 44 a 48 se emplea Razor y el método `Convert.ToString()` para convertir los datos de tipo `float` a datos tipo `string`, con el fin de mostrar los valores calculados en la vista a través de las etiquetas `<label>`.

```

22 <div class="form-group">
23     @Html.Label("Condición de Gratuidad: ")
24     @Html.Label(Convert.ToString(detalle.Tipo_perdida), new { @style = "color:red" })
25     <div class="row">
26         <div class="col-sm-12" style="justify-content:center;align-items:center;display:flex">
27             <table class="table table-condensed table-striped" style="width:275px;margin-bottom:-20px">
28                 <tr>
29                     <th style="background-color: lightgray">Detalle</th>
30                     <th style="padding-left: 60px; background-color: lightgray">Valor</th>
31                 </tr>
32                 <tr>
33                     <td style="text-align:left">
34                         <dl>
35                             <dt>@Html.Label("Valor Matricula")</dt>
36                             <dt>@Html.Label("Valor Arancel")</dt>
37                             <dt>@Html.Label("Recargo 2da Matricula")</dt>
38                             <dt>@Html.Label("Recargo 3era Matricula")</dt>
39                             <dt>@Html.Label("Bancario")</dt>
40                         </dl>
41                     </td>
42                     <td style="text-align:right">
43                         <dl>
44                             <dt><label>@Convert.ToString(detalle.Valor_Matricula_Ordinario)</label></dt>
45                             <dt><label>@Convert.ToString(detalle.Valor_Arancel)</label></dt>
46                             <dt><label>@Convert.ToString(detalle.Recargo_segunda)</label></dt>
47                             <dt><label>@Convert.ToString(detalle.Recargo_Tercera)</label></dt>
48                             <dt><label>@Convert.ToString(detalle.Bancario)</label></dt>
49                         </dl>
50                     </td>
51                 </tr>
52             </table>

```

**Figura 2.62.** Segmento para mostrar información del valor a pagar

### 3 RESULTADOS Y DISCUSIÓN

En este capítulo se presentan las pruebas de funcionamiento para examinar la correcta ejecución de los requerimientos descritos en las historias de usuario. Para ello, se realizaron pruebas de los controladores y las vistas implementadas en el prototipo.

Para las pruebas de validación de datos se ingresó información en las vistas implementadas en el prototipo, en las que se valida que no existan campos vacíos o se ingresen datos no permitidos. Así mismo, se muestran mensajes informativos cuando se ingresan datos que no cumplen con las validaciones establecidas en cada uno de los modelos implementados en el prototipo.

También se incluyen los resultados obtenidos de las pruebas de validación realizadas a quince estudiantes de la EPN, las mismas que fueron planteadas en forma de encuesta, y permiten validar el funcionamiento del prototipo desde la perspectiva del usuario. Finalmente, una vez obtenidas las respuestas de la encuesta se procedió a mejorar y corregir los errores encontrados en el prototipo.

#### 3.1 PRUEBAS DE FUNCIONAMIENTO

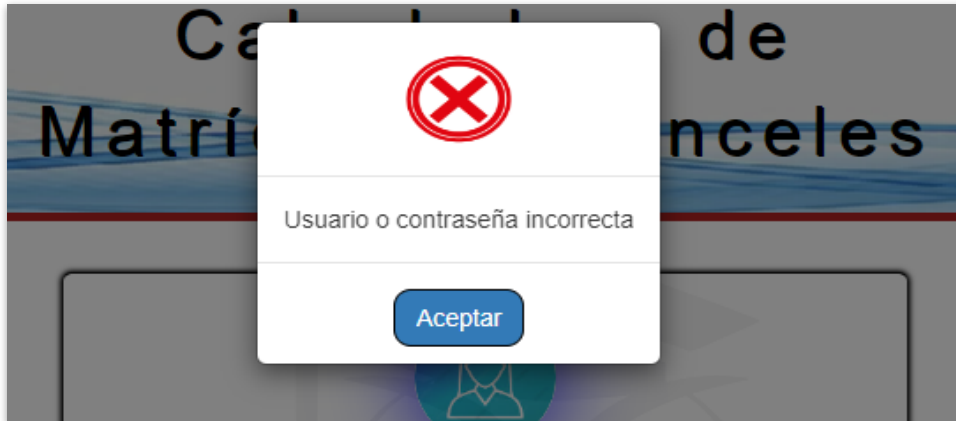
##### 3.1.1 AUTENTICACIÓN DE USUARIO

Para la autenticación de usuario, en el formulario *login* de la vista se presentan campos que deben ser llenados, y en caso de dejar los campos vacíos y presionar el botón de Ingresar, el usuario verá una notificación que le indica que los campos son obligatorios, como se muestra en la Figura 3.1.



Figura 3.1. Validación de los campos en la vista de *login*

También puede presentarse el caso de que el usuario ingrese sus credenciales de forma incorrecta, entonces se mostrará una ventana modal indicando que los datos ingresados están incorrectos, como se muestra en la Figura 3.2.



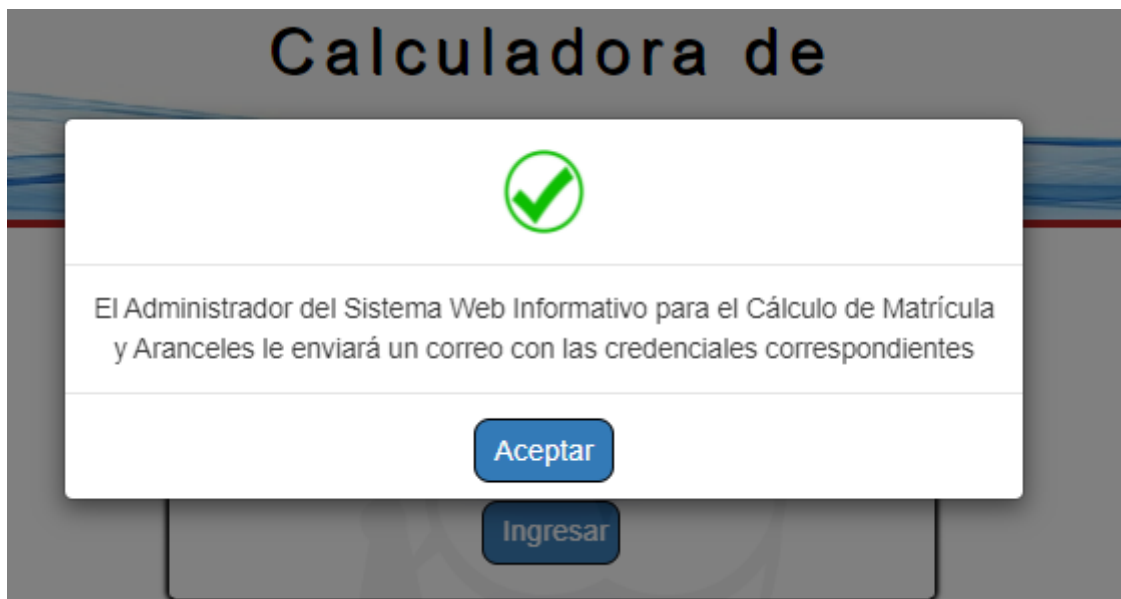
**Figura 3.2.** Mensaje de error de credenciales incorrectas

### 3.1.2 RECUPERACIÓN DE CREDENCIALES POR CORREO

La recuperación de las credenciales de usuario por correo electrónico se realiza por medio del enlace *¿Olvidaste tus credenciales?* del formulario de *login*. Este enlace muestra una ventana modal, como se muestra en la Figura 3.3, en la cual se debe ingresar el correo electrónico del usuario, y una vez que se verifique el correo electrónico, se muestra un mensaje indicando que se enviará un correo con las credenciales de acceso al usuario, como se muestra en la Figura 3.4.

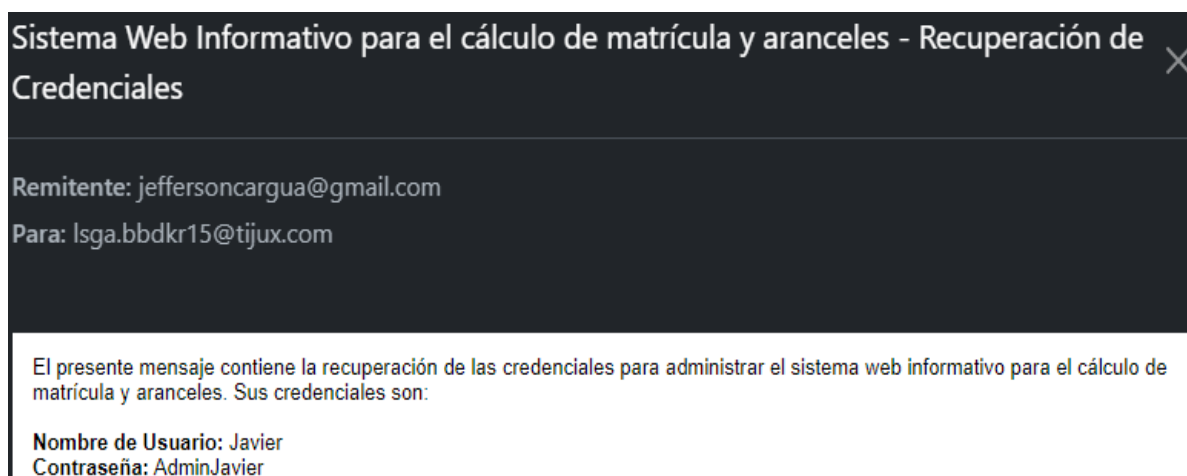


**Figura 3.3.** Ventana para recuperar las credenciales por correo



**Figura 3.4.** Confirmación del correo

A continuación, el usuario podrá chequear su correo electrónico, en el que encontrará un correo que le notificará la recuperación de las credenciales de acceso, como se muestra en la Figura 3.5.



**Figura 3.5.** Correo recibido con la recuperación de las credenciales del usuario

De darse el caso en que un usuario ingrese un correo que no coincida con los correos registrados en la base de datos, se mostrará un mensaje como el de la Figura 3.6.

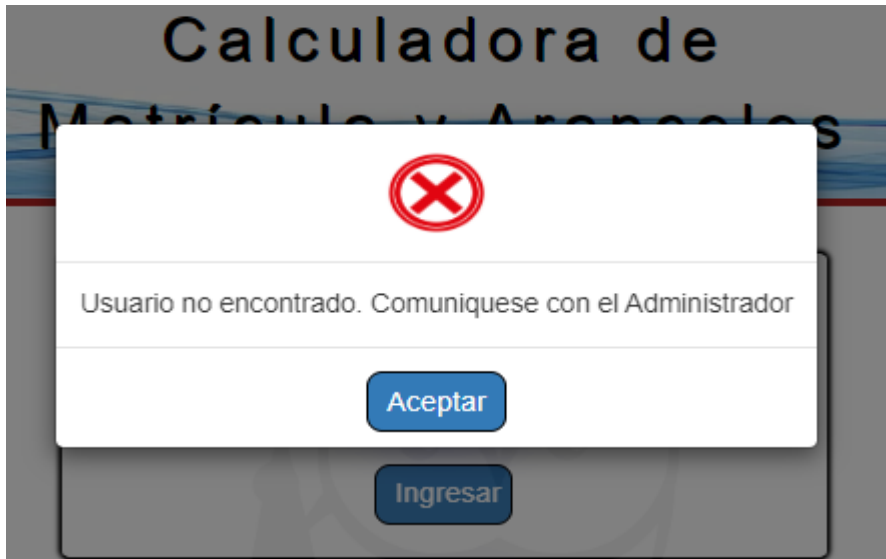


Figura 3.6. Usuario no encontrado

### 3.1.3 CREACIÓN DE UN NUEVO ADMINISTRADOR

Para la creación de un nuevo administrador, se comprueba que los campos del formulario no se envíen vacíos, como se indica en la Figura 3.7. De esta manera se observa mensajes que indican que los campos deben ser llenados obligatoriamente por el usuario. También, se valida el formato del correo electrónico del usuario, y en caso de que no se cumpla con el formato se muestra un mensaje indicando al usuario que el formato del correo ingresado no es válido, como se ve en la Figura 3.8.

Calculadora de Matrícula y Aranceles

Agregar Nuevo Administrador

**Ingrese las credenciales**

Nombre de Usuario:  El campo Usuario es obligatorio.

Contraseña:  El campo Contraseña es obligatorio.

**Ingrese el correo del nuevo usuario**

Correo:  El campo Correo es obligatorio.

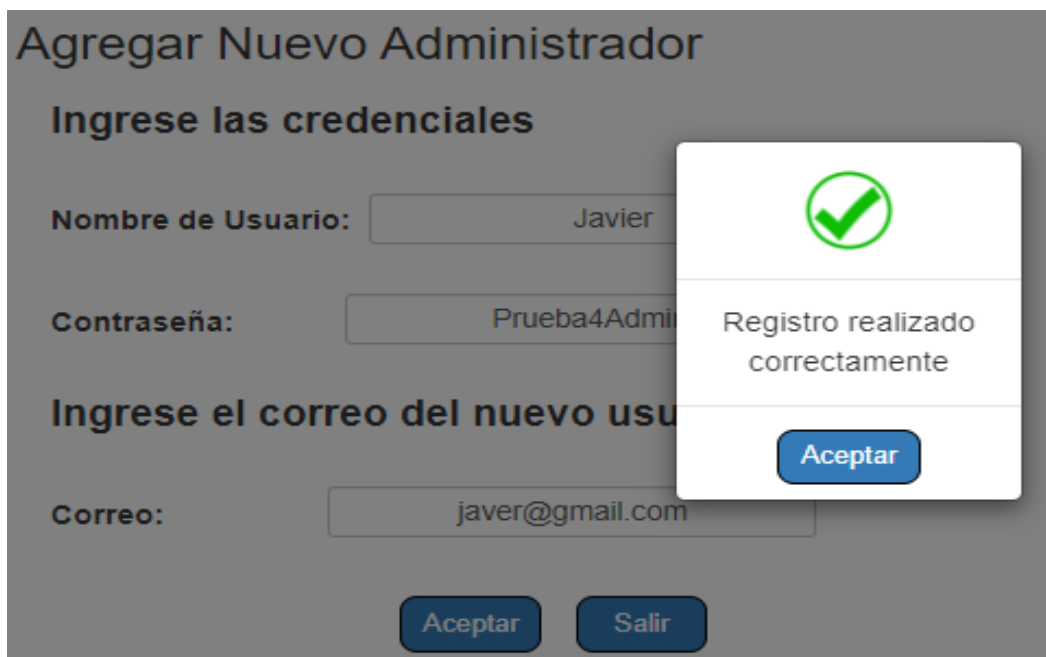
Figura 3.7. Validación de datos del nuevo usuario

### Ingrese el correo del nuevo usuario

Correo:  El campo Correo no es una dirección de correo electrónico válida.

**Figura 3.8.** Validación del formato del correo electrónico


Una vez que la información del nombre de usuario, contraseña y correo electrónico se ingresen correctamente y se presione el botón Aceptar, en la vista se mostrará un mensaje indicando que el registro se ha realizado correctamente, como el de la Figura 3.9. Se puede comprobar si se realizó el registro revisando el listado de usuarios registrados, como se muestra en la Figura 3.10. Además, se envía un correo al usuario registrado con sus credenciales, el cual podrá revisar en su correo electrónico y verá un mensaje como el de la Figura 3.11.



The screenshot shows a registration form titled "Agregar Nuevo Administrador" with the sub-header "Ingrese las credenciales". The form includes fields for "Nombre de Usuario" (filled with "Javier"), "Contraseña" (filled with "Prueba4Admi"), and "Correo" (filled with "javer@gmail.com"). A modal dialog box is overlaid on the form, displaying a green checkmark icon, the text "Registro realizado correctamente", and an "Aceptar" button. The form's "Aceptar" and "Salir" buttons are visible at the bottom.

**Figura 3.9.** Registro exitoso del nuevo usuario

## Información de los Administradores del Sistema

Nombre de Usuario	Correo	
Jeff	jeff@gmail.com	
Juan	juan@gmail.com	
Javier	javier@gmail.com	
Javier	javer@gmail.com	

[Salir](#)

Figura 3.10. Comprobación del nuevo registro

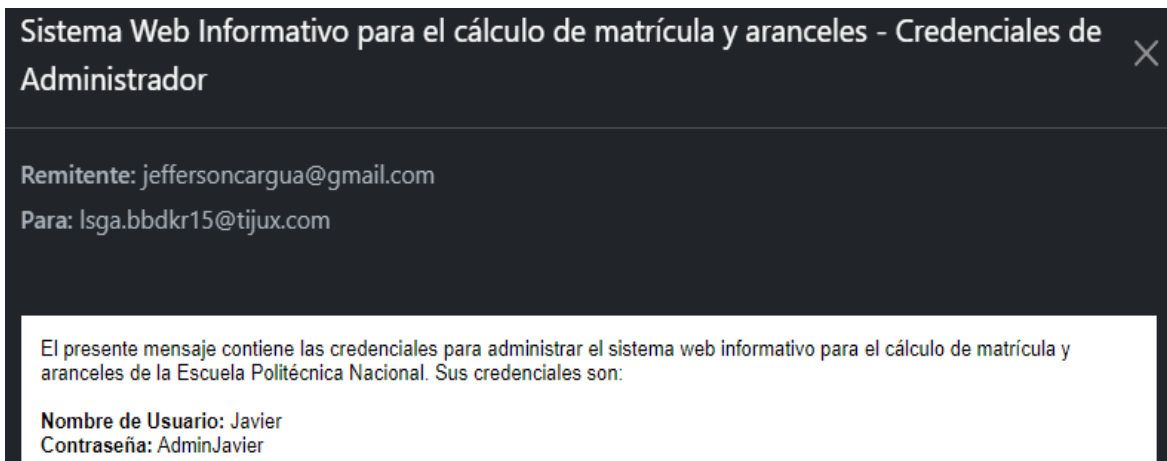


Figura 3.11. Correo recibido con las credenciales

De darse el caso en que el usuario este registrado y se ha ingresado nuevamente sus datos en un nuevo registro, se mostrará un mensaje como el de la Figura 3.12.

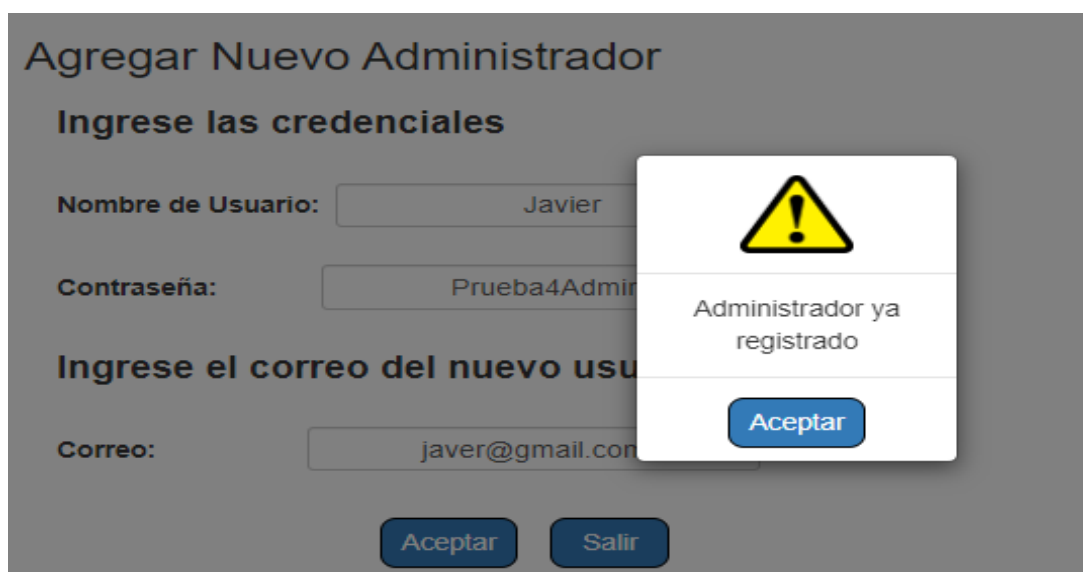


Figura 3.12. Validación de usuario registrado



### 3.1.4 ELIMINACIÓN DE UN ADMINISTRADOR

Para eliminar un administrador registrado en el sistema se muestra una ventana modal, mediante la cual se pregunta si está seguro de eliminar el registro seleccionado. En esta ventana se muestran las opciones de Aceptar o Cancelar, para continuar o no, con la eliminación del registro, como se muestra en la Figura 3.13. En el caso de presionar el botón Aceptar se elimina el registro y se envía un correo al usuario que ha sido eliminado, en cual se indica que ha sido removido del sistema, como se muestra en la Figura 3.14.



Figura 3.13. Mensaje de advertencia de eliminación de usuario

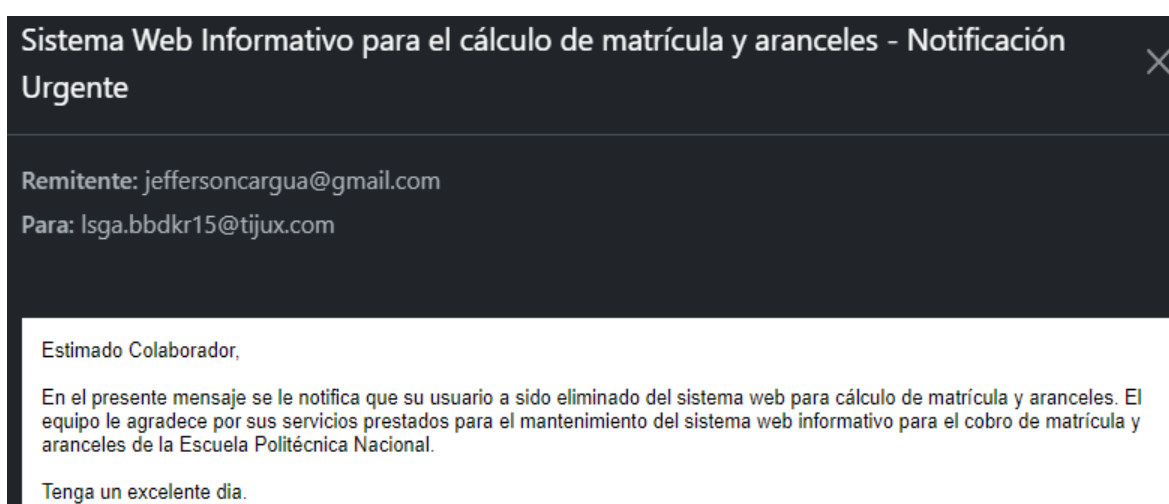


Figura 3.14. Correo enviado al usuario removido del sistema

### 3.1.5 EDICIÓN DE LOS PARÁMETROS BASE

Para editar la información de los parámetros base se tiene una vista que cuenta una tabla que contiene la información que se puede editar, en esta vista se valida que no se dejen campos vacíos durante la edición o se ingresen datos no permitidos, como se muestra en la Figura 3.15.

Parámetro	Valor
Costo Óptimo (\$)	3325hknkm <i>El campo Costo Óptimo (\$) debe ser un número.</i>
Porcentaje del Costo Óptimo anual (%)	50
Porcentaje Valor Mínimo (%)	200 <i>El valor debe estar entre 0 y 100</i>
Porcentaje Valor Máximo (%)	50
Porcentaje del Valor Arancel por Quintil (%)	<i>Debe ingresar el valor del Porcentaje Arancel</i>
Hora Período Académico	kjhk20 <i>El campo Hora Período Académico debe ser un número.</i>
Porcentaje del Promedio Académico (%)	52
Créditos por Período Académico	15
Porcentaje Pérdida Temporal (%)	60
Porcentaje Matrícula Extraordinario (%)	25
Porcentaje Matrícula Especial (%)	ukg <i>El campo Porcentaje Matrícula Especial (%) debe ser un número.</i>
Porcentaje Recargo Segunda Matrícula (%)	10
Porcentaje Recargo Tercera Matrícula (%)	<i>Debe ingresar el valor del Porcentaje de Recargo</i>

**Figura 3.15.** Validación de datos en la edición

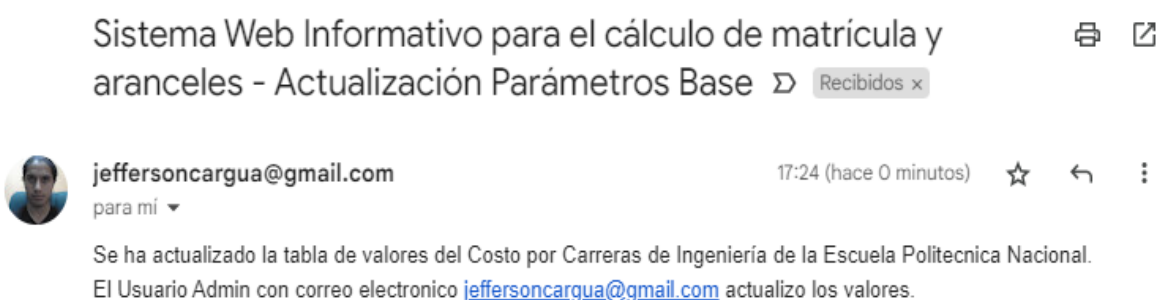
Para comprobar los cambios realizados en la información de los parámetros base se muestra una vista que contiene una tabla con la información recuperada de la base de datos, como se ve en la Figura 3.16. Además, se envía un correo al administrador principal indicando la información de los parámetros base que se modificó y que usuario realizó dicha modificación, como se observa en la Figura 3.17.

### Información del Costo de Carreras de Ingeniería

Parámetro	Valor
Costo Óptimo	\$ 3325
Porcentaje Costo Óptimo Anual	50 %
Costo Óptimo Período	\$ 1662,5
Porcentaje Valor Mínimo	10 %
Valor Mínimo	\$ 166,25
Valor Matrícula Mínimo	\$ 15,11
Valor Arancel Mínimo	\$ 151,14
Porcentaje Valor Máximo	50 %
Valor Máximo	\$ 831,25
Valor Matrícula Máximo	\$ 75,57
Valor Arancel Máximo	\$ 755,68
Horas Período Académico	720
Promedio Período Académico	52 %
Horas Promedio Período Académico	374
Créditos para Período Académico	15
Porcentaje Pérdida Temporal	60 %
Créditos por Pérdida Temporal	9
Costo por Hora Período Académico	\$ 4,04
Porcentaje Valor Arancel	10 %
Porcentaje Matrícula Extraordinario	25 %
Porcentaje Matrícula Especial	25 %
Porcentaje Recargo 2da Matrícula	10 %
Porcentaje Recargo 3era Matrícula	21 %

Editar
Salir

**Figura 3.16.** Vista de la información de los parámetros base



**Figura 3.17.** Correo recibido al editar la información de los parámetros base

### 3.1.6 CÁLCULO DEL VALOR A PAGAR

Para realizar el cálculo del valor a pagar por la matriculación se verifica que los campos para ingresar la información de: formación académica, régimen, quintil y las asignaturas en primera, segunda y tercera matrícula sean llenados. De no ser este caso se muestran mensajes indicando que los campos deben ser llenados de forma obligatoria para poder realizar el cálculo, como se muestra en la Figura 3.18. Además, se muestra un mensaje indicando la forma de ingresar datos en los campos de las asignaturas, como se ve en la Figura 3.19.

Calculadora de Matrícula y Aranceles

Formación Académica :  
Selecciona una opción  
*El campo Tipo Carrera es obligatorio.*

Régimen :  
Selecciona una opción  
*El campo Régimen es obligatorio.*

Quintil :  
Selecciona una opción  
*El campo Quintil es obligatorio.*

Pérdida de Gratuidad: Si:  No:

Asignaturas en créditos/horas a cursar en:

Primera Matrícula:  Ejemplo: 16 *El campo Primera Matrícula es obligatorio.*  
*Ingrese solo Horas*

Segunda Matrícula:  Ejemplo: 0 *El campo Segunda Matrícula es obligatorio.*  
*Ingrese solo Horas*

Tercera Matrícula:  Ejemplo: 0 *El campo Tercera Matrícula es obligatorio.*  
*Ingrese solo Horas*

Calcular Nuevo

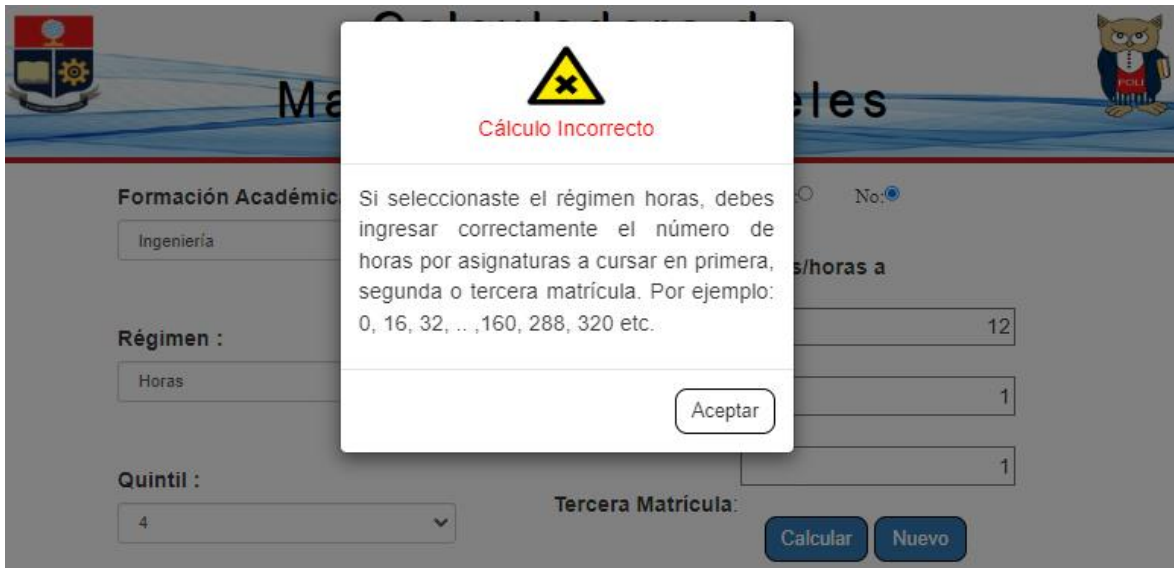
**Figura 3.18.** Validación de los campos del formulario de consulta

Primera Matrícula:  0

Ingrese la cantidad de horas/créditos a cursar en primera matrícula. Por ejemplo, Créditos : 10 u Horas: 160

**Figura 3.19.** Mensaje de ayuda para el ingreso de valores en las asignaturas

De presentarse el caso de que el usuario ha seleccionado el régimen horas e ingresado valores de forma errónea, se presentará un mensaje como el de la Figura 3.20. De esta manera el usuario debe ingresar la información solicitada de forma correcta.



**Figura 3.20.** Advertencia de valores incorrectos

Para comprobar que la información ingresada en el formulario se evalúa correctamente en el controlador, se ingresaron datos que permitan verificar el cálculo del valor a pagar en función de los tipos de pérdida de gratuidad. En la Figura 3.21 se muestra el cálculo del valor a pagar cuando el usuario cuenta con gratuidad.

**Información del Pago de Matrícula**  
Resultados del Cálculo

Matrícula Ordinaria	Matrícula Extraordinaria/Especial																														
Condición de Gratuidad: <b>Con Gratuidad</b>																															
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 70%;">Detalle</th> <th style="width: 30%;">Valor</th> </tr> </thead> <tbody> <tr><td>Valor Matrícula</td><td style="text-align: right;">0</td></tr> <tr><td>Valor Arancel</td><td style="text-align: right;">0</td></tr> <tr><td>Recargo 2da Matrícula</td><td style="text-align: right;">0</td></tr> <tr><td>Recargo 3era Matrícula</td><td style="text-align: right;">0</td></tr> <tr><td>Bancario</td><td></td></tr> <tr> <td>Total a pagar: \$</td> <td style="text-align: right; border: 1px solid red;">0</td> </tr> </tbody> </table>	Detalle	Valor	Valor Matrícula	0	Valor Arancel	0	Recargo 2da Matrícula	0	Recargo 3era Matrícula	0	Bancario		Total a pagar: \$	0	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 70%;">Detalle</th> <th style="width: 30%;">Valor</th> </tr> </thead> <tbody> <tr><td>Valor Matrícula</td><td style="text-align: right;">0</td></tr> <tr><td>Recargo Matrícula Extraordinaria</td><td style="text-align: right;">15,11</td></tr> <tr><td>Valor Arancel</td><td style="text-align: right;">0</td></tr> <tr><td>Recargo 2da Matrícula</td><td style="text-align: right;">0</td></tr> <tr><td>Recargo 3era Matrícula</td><td style="text-align: right;">1</td></tr> <tr><td>Bancario</td><td></td></tr> <tr> <td>Total a pagar: \$</td> <td style="text-align: right; border: 1px solid red;">16,11</td> </tr> </tbody> </table>	Detalle	Valor	Valor Matrícula	0	Recargo Matrícula Extraordinaria	15,11	Valor Arancel	0	Recargo 2da Matrícula	0	Recargo 3era Matrícula	1	Bancario		Total a pagar: \$	16,11
Detalle	Valor																														
Valor Matrícula	0																														
Valor Arancel	0																														
Recargo 2da Matrícula	0																														
Recargo 3era Matrícula	0																														
Bancario																															
Total a pagar: \$	0																														
Detalle	Valor																														
Valor Matrícula	0																														
Recargo Matrícula Extraordinaria	15,11																														
Valor Arancel	0																														
Recargo 2da Matrícula	0																														
Recargo 3era Matrícula	1																														
Bancario																															
Total a pagar: \$	16,11																														

[Salir](#)

**Figura 3.21.** Valores a pagar con gratuidad

En la Figura 3.22 se muestra el valor a pagar cuando el usuario tome asignaturas en segunda o tercera matrícula, en este caso el usuario cuenta con pérdida parcial de gratuidad.

**Información del Pago de Matrícula**  
Resultados del Cálculo

Matrícula Ordinaria		Matrícula Extraordinaria/Especial	
Condición de Gratuidad: <b>Pérdida Parcial</b>		Condición de Gratuidad: <b>Pérdida Parcial</b>	
Detalle	Valor	Detalle	Valor
Valor Matrícula	60,45	Valor Matrícula	60,45
Valor Arancel	129,31	Recargo Matrícula Extraordinaria	15,11
Recargo 2da Matrícula	12,93	Valor Arancel	12,93
Recargo 3era Matrícula	0	Recargo 2da Matrícula	0
Bancario	1	Recargo 3era Matrícula	1
Bancario		Bancario	
<b>Total a pagar: \$</b>	<b>203,69</b>	<b>Total a pagar: \$</b>	<b>218,80</b>

[Salir](#)

**Figura 3.22.** Valores a pagar con pérdida de gratuidad parcial

De darse el caso en que el usuario registre una cantidad de créditos u horas inferior a la cantidad de horas u créditos a tomar por periodo, el usuario tendrá una pérdida parcial + temporal de gratuidad, como se muestra en la Figura 3.23.

**Información del Pago de Matrícula**  
Resultados del Cálculo

Matrícula Ordinaria		Matrícula Extraordinaria/Especial	
Condición de Gratuidad: <b>Pérdida Temporal + Parcial</b>		Condición de Gratuidad: <b>Pérdida Temporal + Parcial</b>	
Detalle	Valor	Detalle	Valor
Valor Matrícula	45,34	Valor Matrícula	45,34
Valor Arancel	174,57	Recargo Matrícula Extraordinaria	11,34
Recargo 2da Matrícula	3,88	Valor Arancel	174,57
Recargo 3era Matrícula	0	Recargo 2da Matrícula	3,88
Bancario	1	Recargo 3era Matrícula	0
Bancario		Bancario	1
<b>Total a pagar: \$</b>	<b>224,79</b>	<b>Total a pagar: \$</b>	<b>236,13</b>

[Salir](#)

**Figura 3.23.** Valores a pagar con pérdida de gratuidad temporal + parcial

También se presenta el caso en que el usuario no cuenta con gratuidad, por lo que su condición es de pérdida definitiva de gratuidad y deberá cancelar por todas las asignaturas que tome en el periodo académico, como se muestra en la Figura 3.24.

## Información del Pago de Matrícula

Resultados del Cálculo

Matrícula Ordinaria		Matrícula Extraordinaria/Especial	
Condición de Gratuidad: <b>Pérdida Definitiva</b>		Condición de Gratuidad: <b>Pérdida Definitiva</b>	
Detalle	Valor	Detalle	Valor
Valor Matrícula	45,34	Valor Matrícula	45,34
Valor Arancel	174,57	Recargo Matrícula Extraordinaria	11,34
Recargo 2da Matrícula	3,88	Valor Arancel	3,88
Recargo 3era Matrícula	0	Recargo 2da Matrícula	0
Bancario	1	Recargo 3era Matrícula	1
Bancario		Bancario	
Total a pagar: \$	224,79	Total a pagar: \$	236,13

Salir

Figura 3.24. Valores a pagar con pérdida de gratuidad definitiva

### 3.2 PRUEBAS DE VALIDACIÓN

Las pruebas de validación del prototipo se realizaron con quince estudiantes de la EPN.

Los usuarios estudiantes realizaron el proceso de consulta del valor a pagar por la matriculación, ingreso de datos, cálculo del valor a cancelar y visualización del detalle de los valores calculados de la matrícula, aranceles, recargos y el total a pagar.

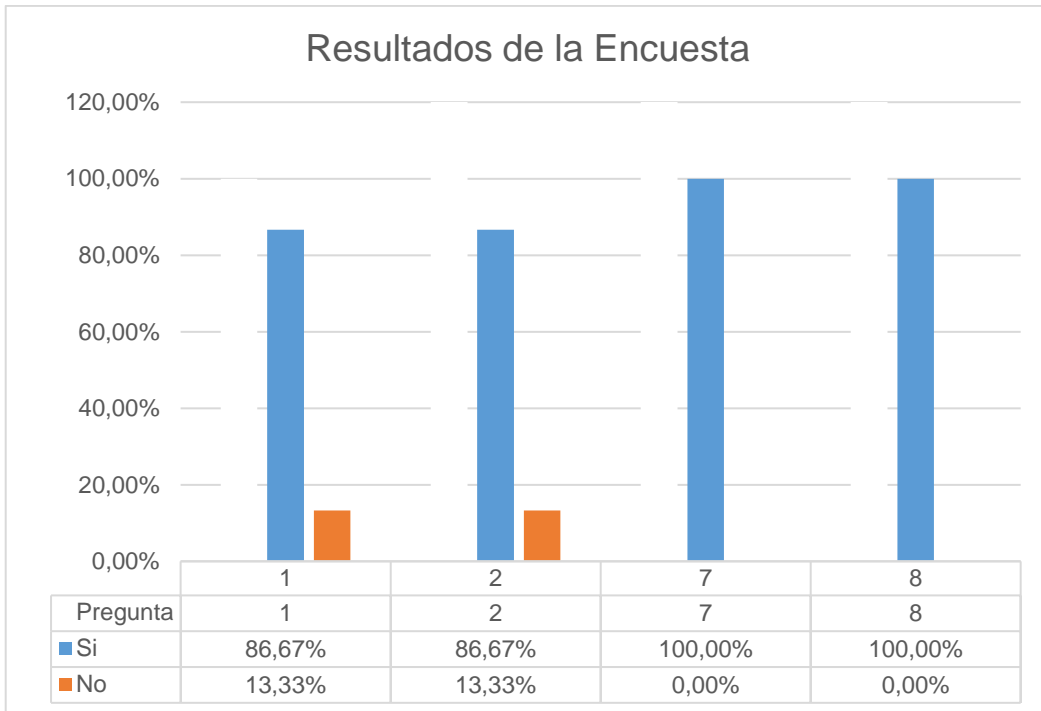
Para las pruebas de validación del lado del administrador, se otorgaron credenciales temporales a quince usuarios administradores.

Los usuarios administradores realizaron el proceso de inicio de sesión, visualización y edición de la información de los parámetros base.

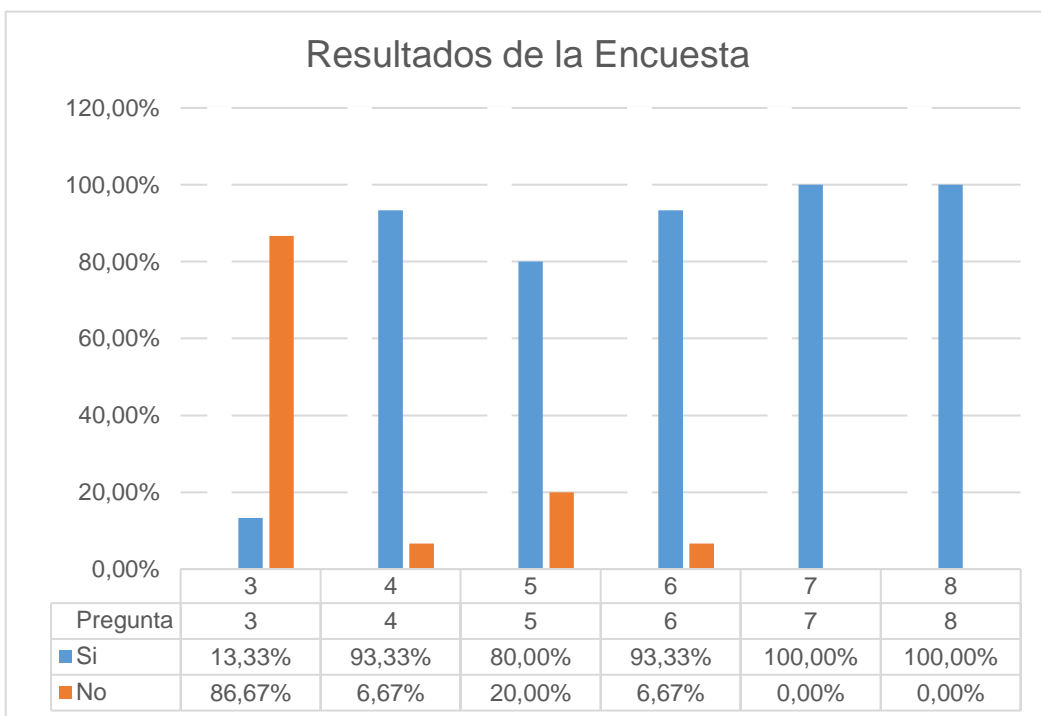
Finalmente, el usuario administrador principal, el cual es el primer usuario creado, probó las funcionalidades de gestión de usuarios del sistema web, para lo cual realizó el proceso de creación y eliminación de usuarios. Cabe recalcar que este usuario es el único que puede hacerlo a través de su contraseña.

Una vez finalizadas las pruebas de validación, los usuarios llenaron una encuesta acerca de la funcionalidad del prototipo de acuerdo con el rol que desempeñaron.

En la Figura 3.25 y en la Figura 3.26 se muestran diagramas de barras que indican los porcentajes de los resultados obtenidos de la encuesta según la ponderación de los usuarios y con base a las dos posibles respuestas: si o no.



**Figura 3.25.** Resultados de las pruebas de validación – Administradores



**Figura 3.26.** Resultados de las pruebas de validación - Estudiantes



La pregunta uno fue: ¿Cómo administrador, pudo iniciar sesión en el sistema?, para comprobar la autenticación de usuarios con un resultado del sí con el 86,67% y un no del 13,33%.

La pregunta dos fue: ¿Cómo administrador, pudo acceder al sistema web para visualizar y editar la información de los parámetros base adecuadamente?, para comprobar que la información de los parámetros base se puede ver y actualizar desde el prototipo con un resultado del sí con un 86,67% y un no del 13,33%.

La pregunta tres fue: ¿Cómo estudiante de la EPN, tuvo inconvenientes al ingresar sus datos en el formulario de consulta para calcular el valor a cancelar por su matriculación?, para comprobar que el ingreso de información permite realizar el cálculo del valor a cancelar con un resultado del no con un 86,67% y un sí del 13,33%.

La pregunta cuatro fue: ¿Cómo estudiante de la EPN, el sistema le ayudó a consultar los valores a cancelar por su matriculación?, para comprobar que los cálculos realizados proporcionan de información útil al estudiante con un resultado del sí con el 93,33% y con un no del 6,67%.

La pregunta cinco fue: ¿Cómo estudiante de la EPN, considera que el sistema proporciona información clara sobre el valor a cancelar por su matriculación?, para comprobar que la información presentada sea entendida por el estudiante con un resultado del sí con 80% y un no del 20%.

La pregunta seis fue: ¿Cómo estudiante de la EPN, recomendaría usar el sistema web para el cálculo de matrícula y aranceles a sus compañeros de la EPN?, para comprobar que el sistema proporciona información útil a los estudiantes con un resultado del sí con el 93,33% y un no del 6,67%.

La pregunta siete fue: ¿Le resultó fácil la navegación en el sistema web para el cálculo de matrícula y aranceles?, para comprobar que la navegación en el prototipo sea sencilla para los usuarios con un resultado del sí con el 100%.

La pregunta ocho fue: ¿Cuán satisfecho/a se encuentra luego de haber probado el sistema web para el cálculo de matrícula y aranceles?, para comprobar si cumple con los requerimientos de usuario con un resultado del sí con el 100%.

Una vez receptado los resultados y mostrados tanto en la Figura 3.25 y la Figura 3.26 se pudo comprobar que los requerimientos de los usuarios se han cumplido exitosamente.

En la Tabla 3.1 se muestran los requerimientos que se cumplieron al finalizar las pruebas de validación.

**Tabla 3.1.** Revisión del administrador

Usuarios	Requerimientos del Sistema	Cumplimiento	
		Si	No
Administrador Principal	Crear y eliminar administradores secundarios.	✓	
Administrador Principal y Secundario	Acceder a la información de los parámetros base de las carreras de nivel de grado como de nivel técnico-tecnológico. Por simplicidad a lo largo del documento se referirá a las carreras de grado como carreras de Ingeniería y a las carreras del nivel técnico-tecnológico como carreras de Tecnología	✓	
	Recuperar las credenciales de acceso.	✓	
	Consultar la información de los parámetros base de las carreras tanto del nivel de grado como del nivel técnico-tecnológico.	✓	
	Modificar la información de los parámetros base de las carreras tanto del nivel de grado como del nivel técnico-tecnológico.	✓	
Estudiante	Consultar los valores a cancelar por concepto de matrícula y aranceles.	✓	
	Ingresar información para realizar los cálculos.	✓	
	Calcular el valor total a cancelar.	✓	
	Visualizar el detalle de los cálculos y el valor a cancelar.	✓	

### 3.3 CORRECCIÓN DE ERRORES

Al finalizar las pruebas de validación y recibir las respuestas de la encuesta, se determinó los errores, así como se definieron las mejoras que fueron implementadas.

En la Tabla 3.2 se muestran las correcciones de la revisión del administrador.

**Tabla 3.2.** Revisión del administrador

Correcciones
Se debe agregar un botón para visualizar la contraseña y un botón para recuperar las credenciales de usuarios en el formulario de <i>login</i> .
Se debe mostrar todos los mensajes de validación en español en la tabla para editar la información de los parámetros base.
Mejorar el aspecto de las vistas, formulario <i>login</i> y tablas de información.

En la Figura 3.27 se muestra el formulario de *login* antes de realizar los cambios descritos en la información de la Tabla 3.1, en el que se observa el aspecto que tenía, así como la ausencia de los botones para visualizar la contraseña y recuperar las credenciales.

Los cambios descritos en la Tabla 3.1 se muestran en la Figura 3.28, en el que se muestra una mejora en el aspecto del formulario y de los botones, así como los botones para visualizar la contraseña y para recuperar las credenciales.



The image shows a login form for an administrator. At the top center is a circular icon with a person silhouette. Below it, the word "Administrador" is written in bold. There are two input fields: "Nombre de Usuario" and "Contraseña". Below the fields are two buttons: "Ingresar" (highlighted in blue) and "Ocultar" (a plain text link).

**Figura 3.27.** Formulario de *login* antes de los cambios



The image shows the updated login form. It features a blue circular icon with a person silhouette. Below it, the word "Administrador" is written in bold. The "Nombre de Usuario" field now has a user silhouette icon on the left. The "Contraseña" field has a lock icon on the left and a refresh/clear icon on the right. Below the fields are two buttons: "Ingresar" (highlighted in blue) and "Ocultar" (a plain text link). At the bottom, there is a blue link: "¿Olvidaste tus credenciales?".

**Figura 3.28.** Formulario de *login* después de los cambios

En la Figura 3.29 se presenta la tabla con la información de los parámetros base a editar, en el que se observan los mensajes en inglés y español, cuando se produce un error en la validación de datos.

La misma tabla con los cambios realizados se observa en la Figura 3.30. Se aprecia que los mensajes se encuentran en español y con un color de fuente diferente para mostrar con mayor énfasis los errores en la validación de datos.

Información de Parámetros Base de Carreras de Ingeniería		
Parámetro	Valor	Observaciones
Costo Óptimo (\$)	3325	The field Porcentaje del Costo Óptimo anual (%) must be a number.
Porcentaje del Costo Óptimo anual (%)	sdf	The field Porcentaje del Costo Óptimo Maximo (%) must be a number.
Porcentaje del Costo Óptimo Minimo (%)	10	
Porcentaje del Costo Óptimo Maximo (%)	50sdf	El campo Horas Semestre Promedio es obligatorio.
Porcentaje del Valor Arancel por Quintil (%)	10	The field Porcentaje Horas Semestre Promedio (%) must be a number.
Horas Semestre Promedio		El campo Porcentaje Recargo Segunda Matricula (%) es obligatorio.
Porcentaje Horas Semestre Promedio (%)	1sdf5	
Porcentaje Recargo Segunda Matricula (%)		
Porcentaje Recargo Tercera Matricula (%)	21	

**Figura 3.29.** Tabla para editar la información de los parámetros base antes de los cambios

Información del Costo de Carreras de Ingeniería		
Parámetro	Valor	Observaciones
Costo Óptimo (\$)	3325hknkm	El campo Costo Óptimo (\$) debe ser un número.
Porcentaje del Costo Óptimo anual (%)	50	
Porcentaje Valor Minimo (%)	200	El valor debe estar entre 0 y 100
Porcentaje Valor Máximo (%)	50	
Porcentaje del Valor Arancel por Quintil (%)		Debe ingresar el valor del Porcentaje Arancel
Hora Período Académico	kjhk20	El campo Hora Período Académico debe ser un número.
Porcentaje del Promedio Academico (%)	52	
Créditos por Período Académico	15	
Porcentaje Perdida Temporal (%)	60	
Porcentaje Matrícula Extraordinario (%)	25	
Porcentaje Matrícula Especial (%)	ukg	El campo Porcentaje Matrícula Especial (%) debe ser un número.
Porcentaje Recargo Segunda Matrícula (%)	10	
Porcentaje Recargo Tercera Matrícula (%)		Debe ingresar el valor del Porcentaje de Recargo

**Figura 3.30.** Tabla para editar la información de los parámetros base después de los cambios

En la Tabla 3.3 se muestran las correcciones de la revisión del estudiante.

**Tabla 3.3.** Revisión del estudiante

<b>Correcciones</b>
No permitir el ingreso de caracteres distintos a números enteros en los campos para ingresar la cantidad de créditos/horas en las asignaturas en primera, segunda y tercera matrícula.
Incluir información que permita conocer el tipo de pérdida de gratuidad al que puede pertenecer un estudiante.
Colocar instrucciones para consultar el quintil al que pertenece un estudiante de la EPN.
Agregar un enlace al sistema SAEw para consultar el quintil y el porcentaje de reprobación de un estudiante.

En la Figura 3.31 se muestra los campos para el ingreso de la cantidad de horas/créditos antes de realizar los cambios descritos en la Tabla 3.2, en el que se observa que permite ingresar caracteres no numéricos.

Una vez realizados los cambios, en la Figura 3.32 se muestra un ejemplo de los datos que se permiten ingresar. Además, al pretender ingresar caracteres alfabéticos o especiales no se escriben en los campos.

<b>Primera Matrícula</b>	<input type="text" value="gkuubjbjbj"/>	The field Primera Matrícula must be a number.
<b>Segunda Matrícula</b>	<input type="text" value="hbj,hj"/>	The field Segunda Matrícula must be a number.
<b>Tercera Matrícula</b>	<input type="text" value="lugbkuygbh"/>	The field Tercera Matrícula must be a number.

**Figura 3.31.** Campos de ingreso de créditos/horas antes de los cambios

**Asignaturas en créditos/horas a cursar en:**

<b>Primera Matrícula:</b>	<input type="text" value="Ejemplo: 16"/>
	Ingrese solo Créditos
<b>Segunda Matrícula:</b>	<input type="text" value="Ejemplo: 0"/>
	Ingrese solo Créditos
<b>Tercera Matrícula:</b>	<input type="text" value="Ejemplo: 0"/>
	Ingrese solo Créditos

**Figura 3.32.** Campos de ingreso de créditos/horas después de los cambios

En la Figura 3.33 se muestra los cambios realizados en la vista de consulta, en el que se agrega información que permite consultar el quintil y el porcentaje de reprobación de materias de un estudiante a través de un enlace al SAEW de la EPN, así como la información de acuerdo al tipo de pérdida de gratuidad.

### **Aviso:**

**Si no recuerdas el quintil socioeconómico al que perteneces, sigue las siguientes instrucciones:**

- Ingresa al SAEW con el siguiente enlace: <https://saew.epn.edu.ec/>
- Accede al sistema con tus credenciales.
- Dirígete al módulo de Información Estudiantil.
- En la pestaña Matriculación selecciona Información de Pagos.
- En Información de Costo de Matrícula selecciona tu carrera y observarás la información del Quintil al que perteneces.

### **Información Adicional:**

- Si reprobaste asignaturas en el período académico previo, tu situación de gratuidad es de Pérdida Parcial de Gratuidad.
- En el caso de las carreras del nuevo régimen, si no te inscribiste en el 60% de los créditos del nivel referencial(15 créditos), tu situación de gratuidad es de Pérdida Temporal de Gratuidad.
- Si acumulaste más del 30% de créditos u horas de asignaturas fallidas, respecto a los créditos u horas totales de tu carrera, tu situación de gratuidad es de Pérdida Definitiva de Gratuidad.
- Si deseas conocer el % de Reprobación de la Carrera visita el [SAEW](#) y revisa tu Currículum Académico.
- Para efectos de cálculo, considera que 1 crédito equivale a 16 horas.

Salir

**Figura 3.33.** Información agregada para consultar el quintil y conocer los tipos de pérdida de gratuidad

## 4 CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentan las conclusiones y recomendaciones obtenidas a lo largo del desarrollo del prototipo.

### 4.1 CONCLUSIONES

- Al finalizar el presente Trabajo de Titulación se logró obtener un prototipo de sistema web informativo para el cálculo de matrícula y aranceles.
- Para el desarrollo del prototipo de este Trabajo de Titulación se empleó el *framework* ASP.NET que permite desarrollar de aplicaciones web mediante el uso de la tecnología .NET que brinda herramientas que facilitan el desarrollo web como Razor para la creación de vistas sencillas y dinámicas, el patrón de diseño Modelo-Vista-Controlador, Entity Framework para la comunicación con la base de datos y SQL Server. Todo esto facilitó el desarrollo del prototipo permitiendo presentar un producto entregable.
- Como parte del desarrollo del prototipo se levantó información sobre la normativa actual para el cobro de matrícula y aranceles de la EPN que permite determinar el valor que un estudiante debe cancelar por la matriculación, esto debido a la pérdida de gratuidad, identificando cómo funciona y que parámetros son tomados en cuenta.
- Los requisitos funcionales se cumplieron de forma que el sistema permite gestionar la información de los parámetros base, administrar usuarios, consultar y calcular el valor a pagar por la matriculación, entre otras funciones.
- Para la estructuración del código fuente del prototipo se utilizó la arquitectura MVC, el cual permitió mantener un código ordenado y de fácil entendimiento al momento de implementar funcionalidades o realizar correcciones.
- Se utilizó el gestor de paquetes de NuGet para agregar automáticamente Bootstrap al proyecto utilizado para aplicar diseños en los elementos empleados en las vistas y Entity Framework para generar los modelos de almacenamiento de la base de datos.
- La creación de clases ViewModel, permitió agregar etiquetas de validación mediante la librería Data Annotations, con las cuales se generan avisos y comprobaciones de datos para cumplir con las solicitudes de los usuarios.
- Para el desarrollo de las vistas del prototipo se empleó la herramienta Razor que permitió mezclar código escrito en C# con etiquetas HTML. Así mismo, se agregó

la herramienta de Bootstrap que permitió mejorar el aspecto visual de las vistas, y también se agregó código escrito en JavaScript que permitió dar estilo y efectos dinámicos a los elementos de las vistas.

- Para la gestión de la información de los parámetros base del sistema web se usaron rutas personalizadas para introducir parámetros específicos, los cuales llevan la información solicitada para probar el funcionamiento de visualización y edición.
- Para poder presentar información al usuario sin que se tenga que salir de la vista actual se utilizó ventanas modales y JQuery. Esto permite generar una ventana flotante que se muestra sobre la vista actual, lo cual permite al usuario interactuar con mensajes o formularios, evitando salir de la vista actual o cargar otras vistas.
- Los resultados obtenidos de las pruebas de validación como se puede apreciar en el capítulo tres, demostraron que los administradores pudieron iniciar sesión, visualizar y editar la información de los parámetros base, así mismo los estudiantes pudieron ingresar información que les permitió conocer el valor a pagar por la matriculación.
- Mediante Microsoft Azure DevOps se usó la metodología ágil Kanban que permitió definir el proceso de desarrollo del sistema web, en el que el tablero implementado permitía conocer el estado de las tareas, lo cual fue útil para conocer cuales estaban pendientes, en cuales se estaba trabajando y cuales estaban completas, así mismo permitía conocer las tareas que se debía revisar y corregir.



## 4.2 RECOMENDACIONES

- Se recomienda que se continúe con el desarrollo del prototipo agregando nuevas funcionalidades por ejemplo permitiendo consultar la información del quintil y porcentaje de reprobación usando el número de cédula para que el estudiante o aspirante pueda conocer esta información sin la necesidad de salir del sistema.
- Dado a que el sistema proporciona un cálculo fijo a cancelar por la matriculación se recomienda para proyectos futuros agregar una funcionalidad que permita generar un cálculo para pago por partes por la matriculación.
- Se podría mejorar la apariencia de las interfaces de usuario para que el sistema sea más intuitivo e interactivo con el usuario para una mejor experiencia.
- Se recomienda crear clases ViewModel para los modelos que necesiten validaciones ya que de esta manera se controlan los eventos generados en los modelos de las vistas y se evitan problemas que afecten el desempeño del sistema o la base de datos.
- Utilizar Bootstrap en las aplicaciones web, ya que cuenta con una gran variedad de clases que permite aplicar diseños a los elementos que se utilizan en el documento HTML para presentar vistas con estilos avanzados.
- Utilizar una cuenta de correo electrónico que cuente con una contraseña de aplicaciones, de tal forma que se pueda otorgar seguridad en el envío de correos electrónicos.
- Desarrollar una función que permita facilitar y automatizar la generación de contraseñas que se enviarán a los administradores secundarios.
- Se realizaron pruebas de compatibilidad sobre los navegadores web más utilizados: Google Chrome, Mozilla Firefox, Opera y Microsoft Edge, sin embargo, se recomienda realizar pruebas en otros navegadores para comprobar el funcionamiento del sistema web, sobre todo por las funcionalidades escritas en JavaScript.
- Utilizar el patrón Modelo-Vista-Controlador en proyectos de aplicaciones web, ya que permite separar las interfaces de usuario, la lógica de la base de datos y negocio, con el objetivo de que el sistema tenga una estructura robusta y escalable.
- Emplear Azure DevOps para el desarrollo de proyectos debido a que permite emplear la metodología ágil Kanban y enlazarlo con el entorno de desarrollo web ASP.NET en Visual Studio, lo que facilita la realización de las tareas asignadas y conocer el estado del tablero.

## 5 REFERENCIAS BIBLIOGRÁFICAS

- [1] «Reglamento de cobro de matrícula y aranceles de las carreras de grado, tecnología superior y nivelación de la Escuela Politécnica Nacional,» Escuela Politécnica Nacional, 29 01 2019. [En línea]. Disponible: <https://www.epn.edu.ec/wp-content/uploads/2020/05/Codificación-de-Reglamento-de-Cobro-de-Matrículas-enero-2019.pdf>.
- [2] «Guía para el Estudiante,» Escuela Politécnica Nacional, 17 04 2020. [En línea]. Disponible: [https://www.epn.edu.ec/wp-content/uploads/2020/04/Guía\\_para\\_Estudiente.pdf](https://www.epn.edu.ec/wp-content/uploads/2020/04/Guía_para_Estudiente.pdf).
- [3] J. A. Senso, «Tecnologías web para los servicios de información,» Universidad de Granada, 28 09 2016. [En línea]. Disponible: <https://blogs.ugr.es/tecweb/los-sistemas-informacion-basados-la-web/>.
- [4] L. Gilibets, «Qué es la metodolgia Kanban y cómo utilizarla,» 11 11 2020. [En línea]. Disponible: <https://www.iebschool.com/blog/metodologia-kanban-agile-scrum/>.
- [5] Kanban Tool , «Metodología Kanban,» Shore Labs, 2009. [En línea]. Disponible: <https://kanbantool.com/es/metodologia-kanban>.
- [6] Kanbanize, «¿Qué es un tablero Kanban y cómo utilizarlo?,» [En línea]. Disponible: <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban>.
- [7] F. Escolar, «Método Kanban con Azure DevOps,» 26 02 2019. [En línea]. Disponible: <https://www.developerro.com/2019/02/26/azure-devops-kanban/>.
- [8] Kanban Tool, «Tarjeta Kanban,» Shore Labs, 01 04 2020. [En línea]. Disponible: <https://kanbantool.com/es/guia-kanban/tarjeta-kanban>.
- [9] M. Rehkopf, «Historias de usuario,» Atlassian, [En línea]. Disponible: <https://www.atlassian.com/es/agile/project-management/user-stories>.
- [10] Microsoft, «What is ASP.NET?,» [En línea]. Disponible: <https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet>.
- [11] G. Warren, «Overview of .NET Framework,» 15 09 2021. [En línea]. Disponible: <https://learn.microsoft.com/en-us/dotnet/framework/get-started/overview>.
- [12] G. Warren, «Get started with .NET Framework,» 22 09 2021. [En línea]. Disponible: <https://learn.microsoft.com/en-us/dotnet/framework/get-started/>.
- [13] M. Wenzel, «Entity Framework overview,» 15 09 2021. [En línea]. Disponible: <https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/ef/overview>.

- [14] «What is Entity Framework?,» Entity Framework Tutorial, 2020. [En línea]. Disponible: <https://www.entityframeworktutorial.net/what-is-entityframework.aspx>.
- [15] R. Anderson, «Información general de ASP.NET,» 5 10 2021. [En línea]. Disponible: <https://learn.microsoft.com/es-es/aspnet/overview>.
- [16] «Qué es MVC?,» Desarrolloweb6, 28 07 2020. [En línea]. Disponible: <https://desarrolloweb.com/articulos/que-es-mvc.html>.
- [17] «Modelo vista controlador (MVC),» Universidad de Alicante, [En línea]. Disponible: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>.
- [18] «Qué es HTML5 y por qué se usa en las páginas webs más actuales,» Quality Devs, 9 11 2020. [En línea]. Disponible: <https://www.qualitydevs.com/2020/11/09/que-es-html5/>.
- [19] A. Lozada, «¿Qué es JavaScript y cómo funciona?,» Horizonte, 22 10 2020. [En línea]. Disponible: <https://soyhorizonte.com/blog/que-es-javascript-y-como-funciona/>.
- [20] L. Chuburu, «Qué es JQuery y cómo implementarlo,» Laura Churubu, [En línea]. Disponible: <https://laurachuburu.com.ar/que-es-jquery-y-como-implementarlo/>.
- [21] ECURED, «CSS3,» 22 10 2021. [En línea]. Disponible: <https://www.ecured.cu/CSS3>.
- [22] «ASP.NET MVC - Bootstrap,» Tutorials Point, [En línea]. Disponible: [https://www.tutorialspoint.com/asp.net\\_mvc/asp.net\\_mvc\\_bootstrap.htm](https://www.tutorialspoint.com/asp.net_mvc/asp.net_mvc_bootstrap.htm).
- [23] Hireline, «¿Qué hace un Desarrollador Backend?,» [En línea]. Disponible: <https://hireline.io/mx/enciclopedia-de-perfiles-ti/perfil-de-desarrollador-backend>.
- [24] Cevallos Salas, E. E. (2016). Procesamiento de observaciones meteorológicas para presentación mediante una aplicación web responsiva. pp. 35. Quito : EPN. Repositorio Digital Institucional de la Escuela Politécnica Nacional, [En línea]. Disponible: <http://bibdigital.epn.edu.ec/handle/15000/15262>.
- [25] Gualli Godoy, F. J. (2019). Desarrollo de un sistema web prototipo para facturación electrónica de una ferretería. pp. 22. Quito : EPN. Repositorio Digital Institucional de la Escuela Politécnica Nacional, [En línea]. Disponible: <http://bibdigital.epn.edu.ec/handle/15000/20396>.

## **ANEXOS**

ANEXO A. Historias de usuario

ANEXO B. Diagrama de actividades

ANEXO C. Diagrama de secuencias

ANEXO D. Sketches de la interfaz de usuario

ANEXO E. Script para la generación de la base de datos

ANEXO F. Código fuente del prototipo

ANEXO G. Manual de usuario del prototipo

# ANEXO A

En este anexo se presentan las historias de usuario elaboradas para el prototipo.

**USER STORY 49**

49 Gestión de usuarios

JEFFERSON GUILLERMO CARGUA MEDI... 0 comments Add tag

State	New	Area	DesarrolloDeUnPrototipoDeSistemaWebInformativoCalcul
Reason	New	Iteration	DesarrolloDeUnPrototipoDeSistemaWebInformativoCalcul

**Description**

Como administrador quiero agregar o eliminar usuarios administradores para que estos puedan hacer uso del sistema web.

**Acceptance Criteria**

- La **contraseña del administrador debe ser autenticada**, caso contrario no se podrá crear o eliminar usuarios y se mostrará un mensaje de error.
- Los **nuevos usuarios no deben existir en la base de datos**, en caso contrario se mostrará un mensaje de error.

Si la contraseña es correcta, el administrador podrá crear o eliminar usuarios. Además, si se crea o elimina correctamente se enviará un correo a usuario.

**Planning**

Story Points

Priority 2

Risk

**Classification**

Figura A.1. Historia de usuario - Gestión de usuarios en el prototipo

**USER STORY 19**

19 Almacenamiento de la Información de Parámetros Base

JEFFERSON GUILLERMO CARGUA MEDI... 0 comments Add tag

State	New	Area	DesarrolloDeUnPrototipoDeSistemaWebInformativoCalcul
Reason	Moved to the...	Iteration	DesarrolloDeUnPrototipoDeSistemaWebInformativoCalcul

**Description**

Como administrador, quiero que el sistema web este conectado a una base de datos para extraer y cambiar valores de los parámetros de cobro de matrícula y aranceles con la finalidad de mantener información válida.

**Acceptance Criteria**

- La **información de los parámetros deben existir y estar almacenados en una base de datos**, en caso contrario se mostrará un mensaje de error.

De tener la información completa, el administrador podrá obtener la información de los parámetros de cobro de matrícula y aranceles.

**Planning**

Story Points

Priority 1

Risk 1 - High

**Classification**

Figura A.2. Historia de usuario – Conexión de la base de datos y el prototipo

USER STORY 48

## 48 Gestión de la Información de los Parámetros Base

JEFFERSON GUILLERMO CARGUA MEDI... 0 comments Add tag

State	New	Area	DesarrolloDeUnPrototipoDeSistemaWebInformativoCalcul
Reason	New	Iteration	DesarrolloDeUnPrototipoDeSistemaWebInformativoCalcul

### Description

Como administrador quiero ser capaz de gestionar la información de los parámetros base para verificar que la información sea correcta para el cálculo de los valores a pagar por la matriculación de los estudiantes en las carreras de ingeniería y tecnología.

### Acceptance Criteria

- Los **valores de los parámetros deben tener valor** , caso contrario se mostrará un mensaje de error.
- Los **valores de los parámetros deben ser de tipo numérico con punto flotante** , caso contrario se mostrará un mensaje de error.

Si toda la información es correcta, el administrador podrá visualizar y actualizar los valores de los parámetros de cobro de matrícula y aranceles.

### Planning

Story Points

Priority

2

Risk

### Classification

Figura A.3. Historia de usuario – Edición de la información de los parámetros base en el prototipo

USER STORY 66

## 66 Experiencia del Administrador

JEFFERSON GUILLERMO CARGUA MEDI... 0 comments Add tag

State	New	Area	DesarrolloDeUnPrototipoDeSistemaWebInformativoCalcul
Reason	New	Iteration	DesarrolloDeUnPrototipoDeSistemaWebInformativoCalcul

### Description

Como administrador, quiero que el sistema web funcione de manera rápida y eficaz para realizar el mantenimiento de la información de los parámetros base.

### Acceptance Criteria

*Click to add Acceptance Criteria*

### Discussion

### Planning

Story Points

Priority

2

Risk

### Classification

Figura A.4. Historia de usuario – Experiencia del Administrador

## 17 Cálculo de valores a pagar por matriculación

JEFFERSON GUILLERMO CARGUA MEDI... 0 comments Add tag

State	New	Area	DesarrolloDeUnPrototipoDeSistemaWebInformativoCalc..
Reason	Moved to the...	Iteration	DesarrolloDeUnPrototipoDeSistemaWebInformativoCalc..

### Description

Como estudiante de la EPN, quiero obtener información del valor total a pagar por la matriculación de acuerdo con el quintil al que pertenezco y de las asignaturas que cursaré en el semestre.

### Acceptance Criteria

- La **información del quintil debe tener un valor**, en caso contrario se mostrará un mensaje de error.
- La **información del quintil debe ser un valor del tipo numérico entero**, en caso contrario se mostrará un mensaje de error.
- La **cantidad de horas / créditos deben ser del tipo numérico entero**, en caso contrario se mostrará un mensaje de error.
- La **cantidad de horas / créditos deben tener valor**, en caso contrario se mostrará un mensaje de error.
- La **información de los parámetros para el cálculo deberá existir en una base de datos**, en caso contrario se mostrará un mensaje de error.

Si el estudiante proporciona toda la información correctamente, el estudiante podrá calcular el valor a pagar.

### Planning

Story Points

Priority

2

Risk

### Classification

**Figura A.5.** Historia de usuario – Cálculo del valor a pagar por matriculación en el prototipo

## ANEXO B

En este anexo se muestran los diagramas de actividades realizados para el prototipo.

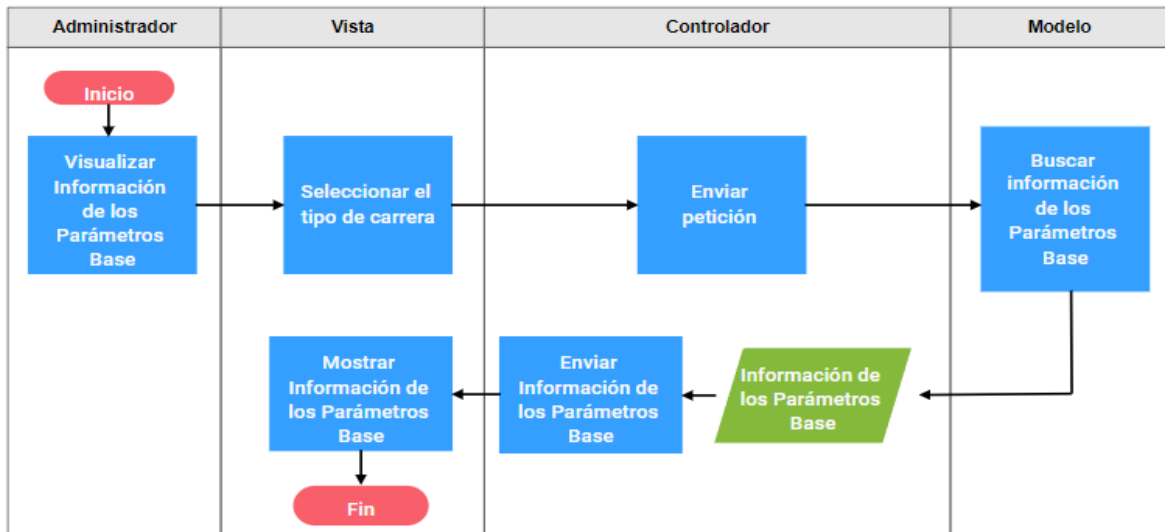


Figura B.11. Diagrama de actividades - Visualizar la información de los parámetros base

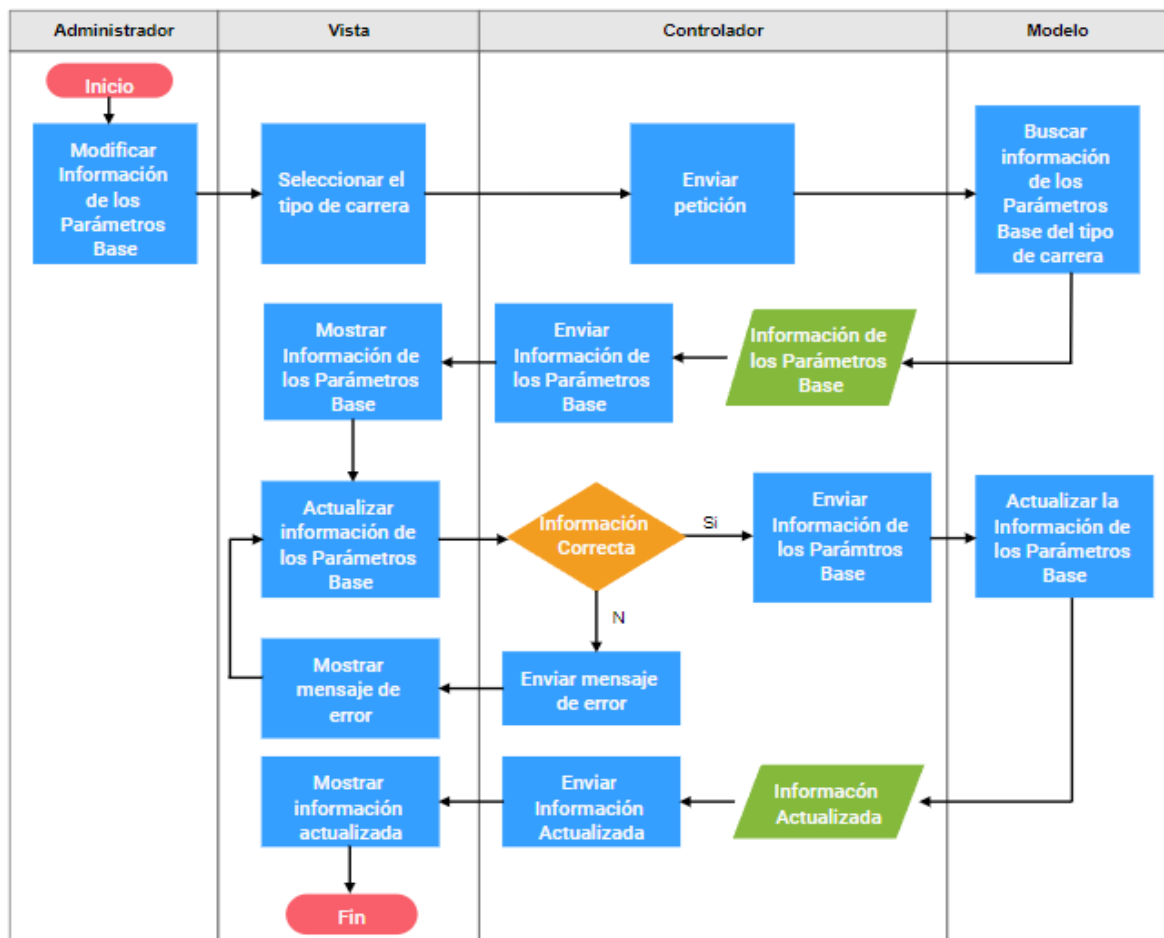


Figura B.2. Diagrama de actividades - Modificar la información de los parámetros base



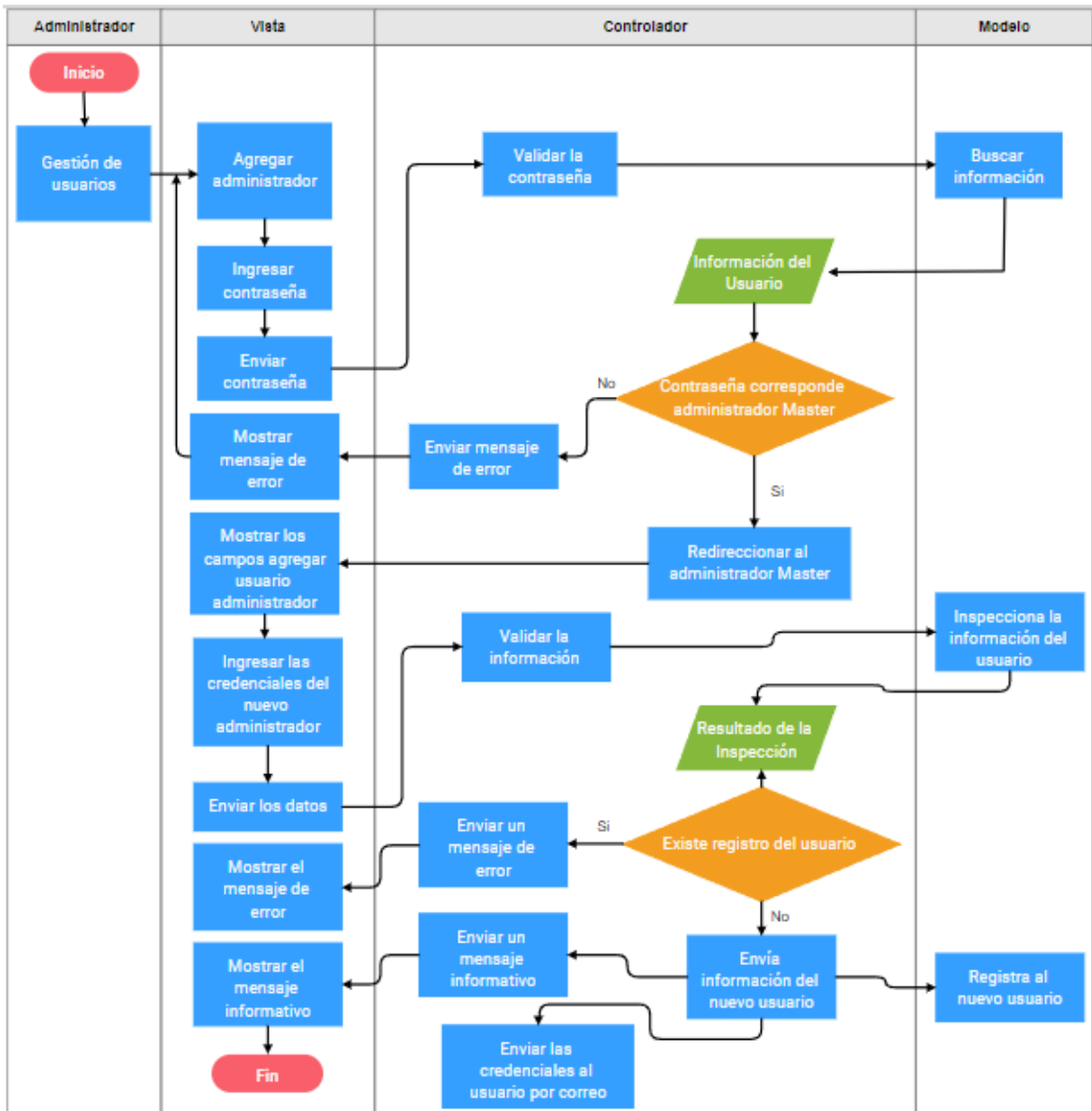


Figura B.3. Diagrama de actividades – Agregar nuevos administradores

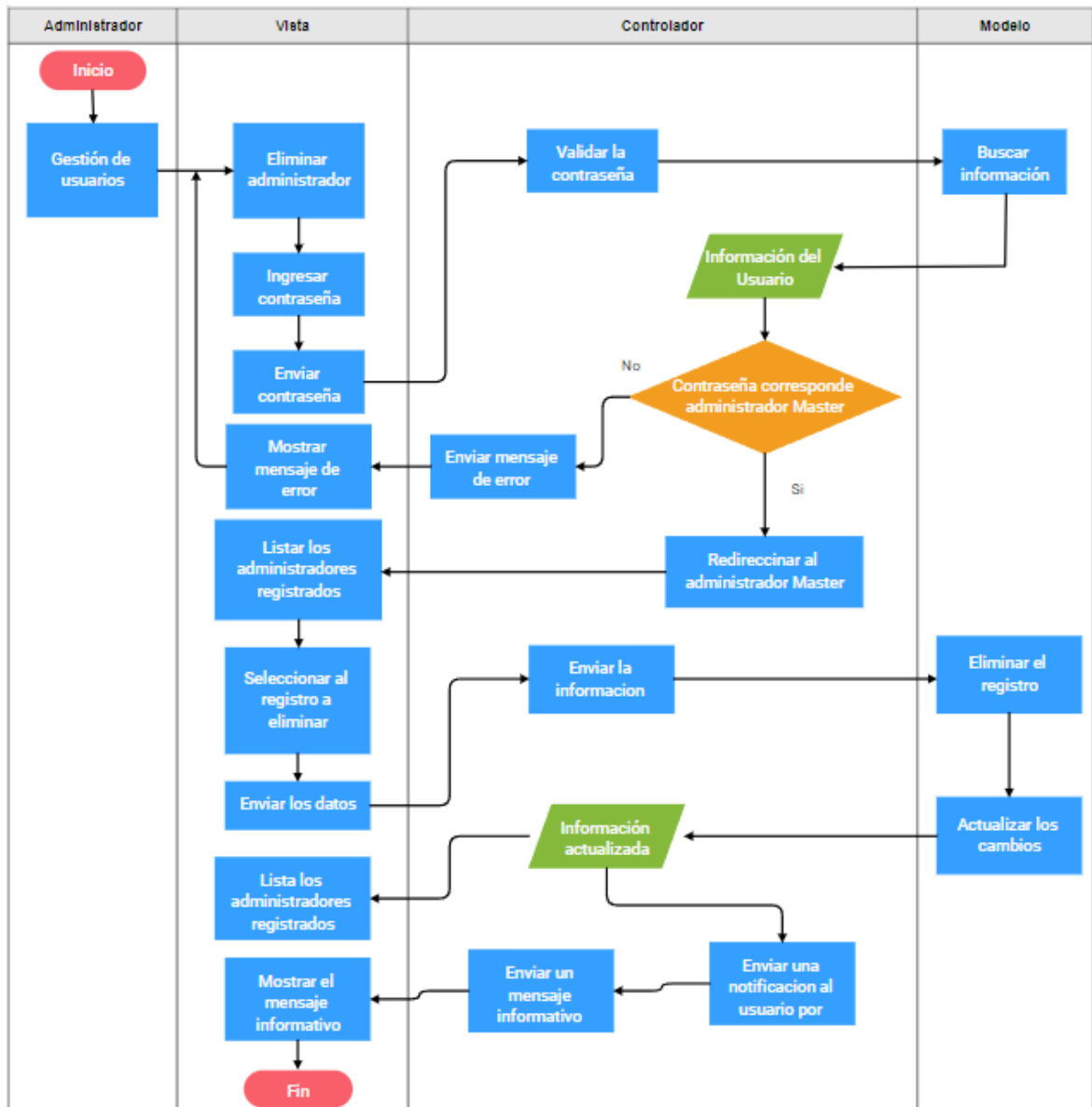
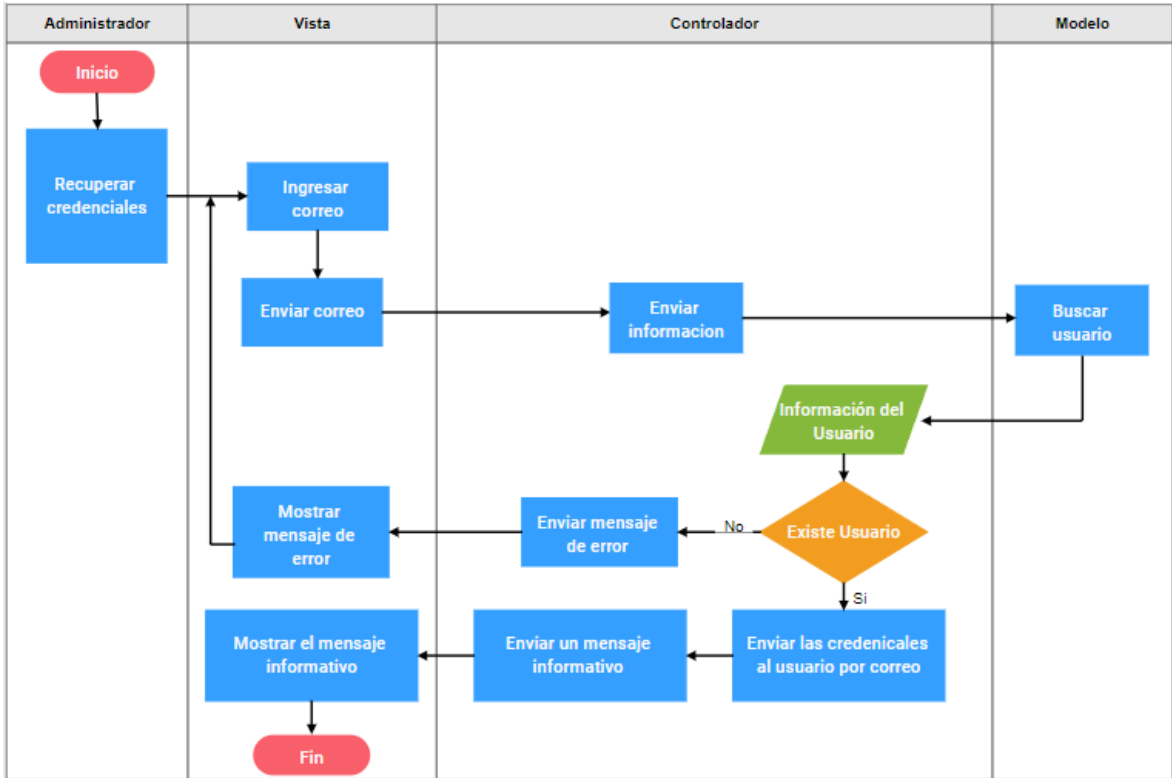


Figura B.4. Diagrama de Actividades – Eliminar administradores



**Figura B.5.** Diagrama de Actividades – Recuperar credenciales de acceso por correo

# ANEXO C

En este anexo se muestran los diagramas de secuencia realizados para el prototipo.

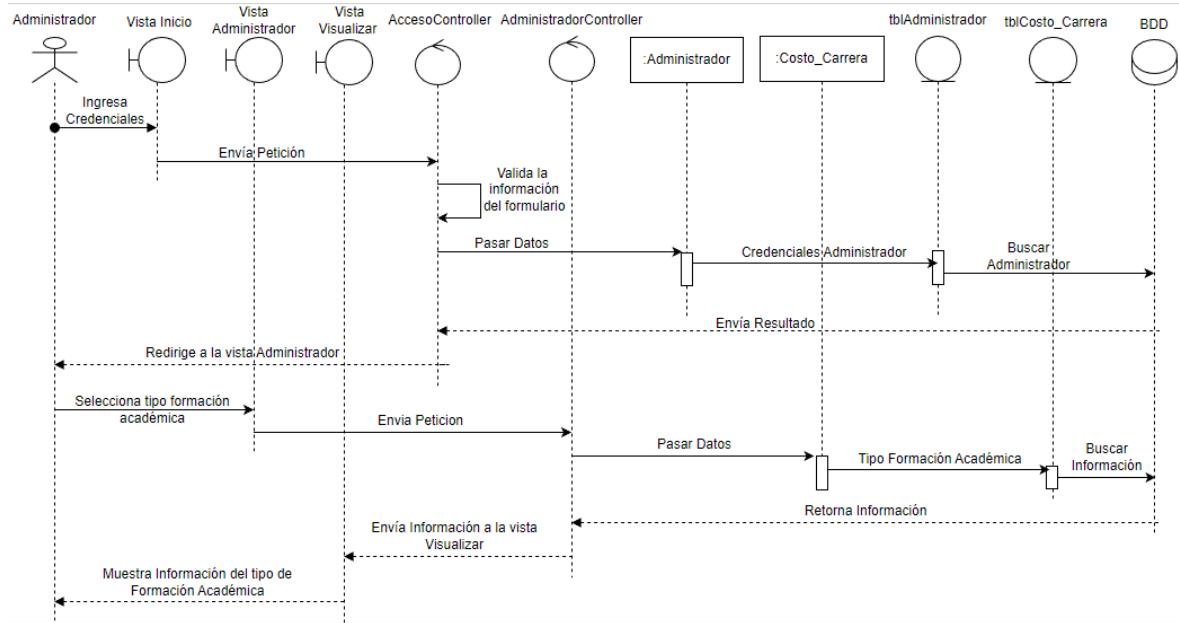


Figura C.1. Diagrama de secuencia – Visualizar la información de los parámetros base

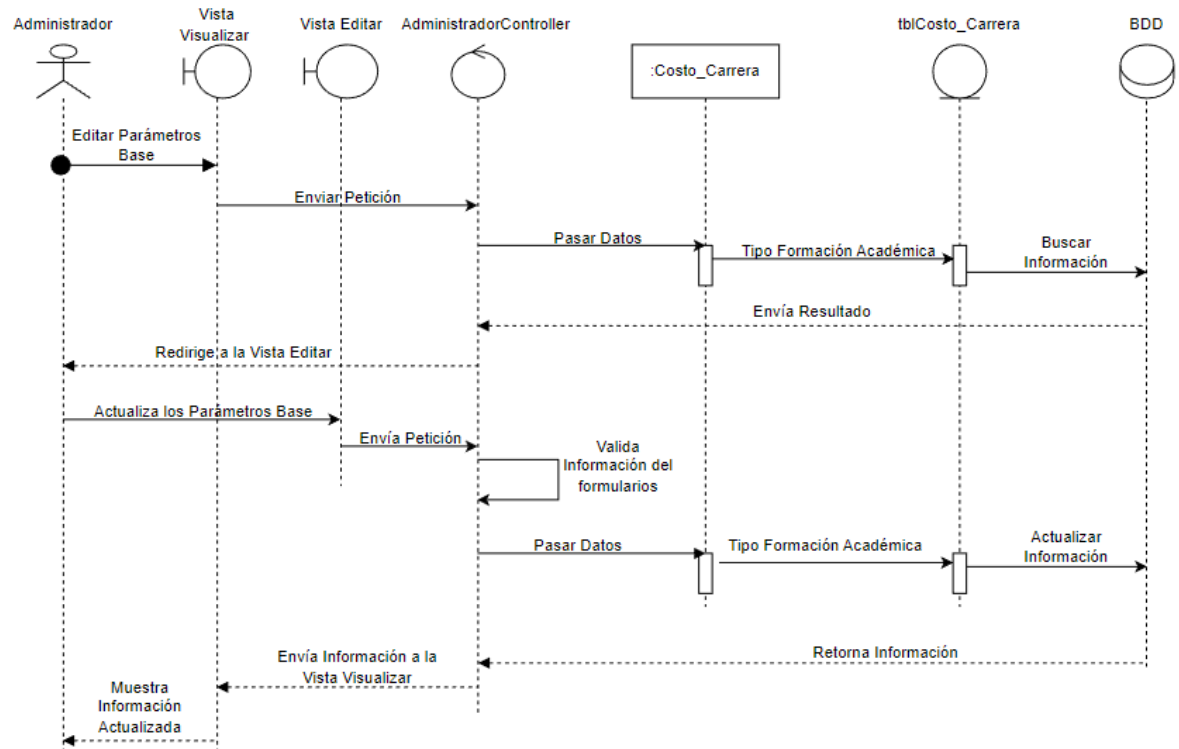
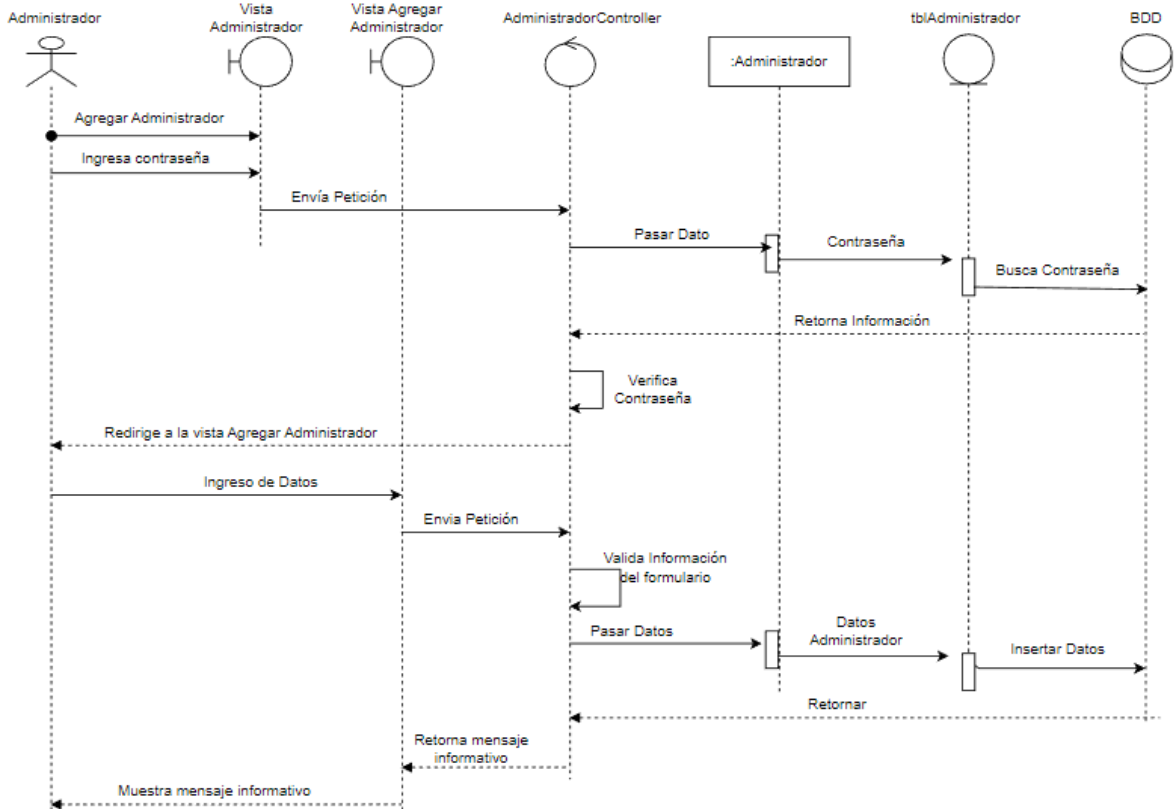
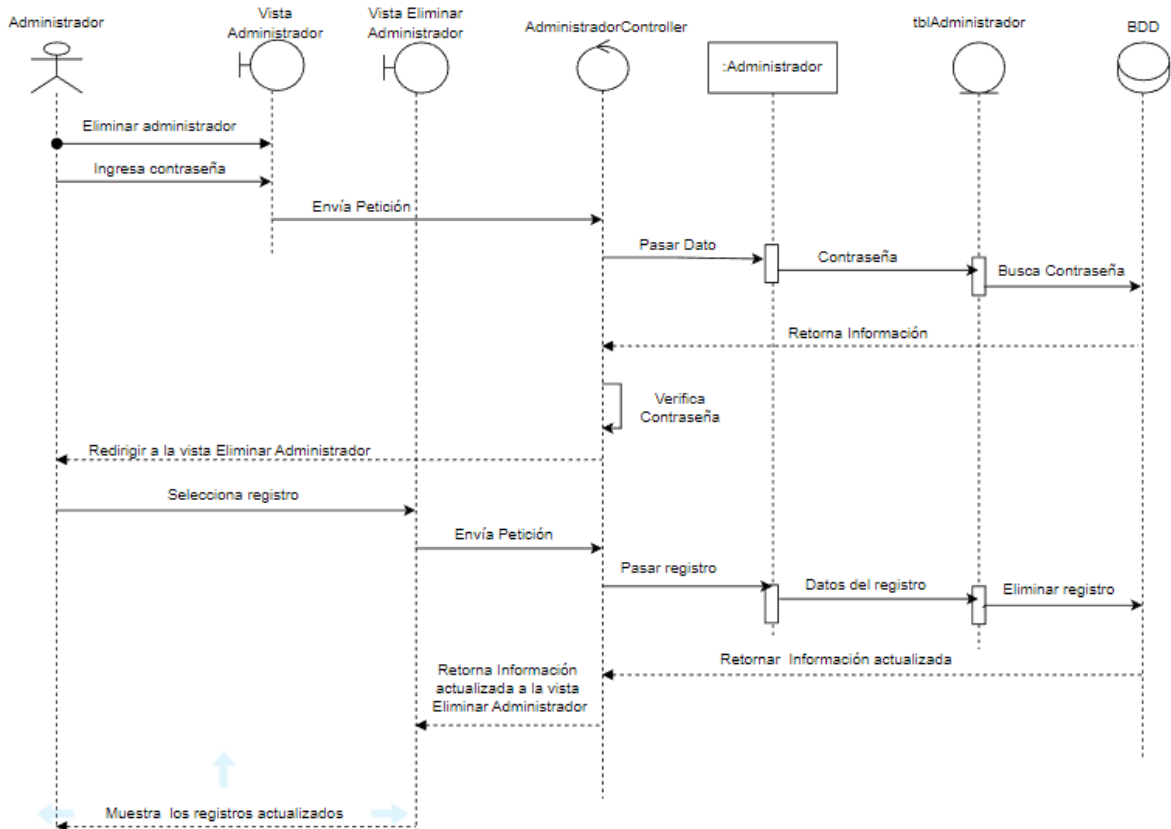


Figura C.2. Diagrama de secuencia – Modificar la información de los parámetros base



**Figura C.3.** Diagrama de secuencia – Agregar nuevo administrador



**Figura C.4.** Diagrama de secuencia – Eliminar administrador

## ANEXO D

En este anexo se muestran los diseños de los sketches para las vistas del prototipo.

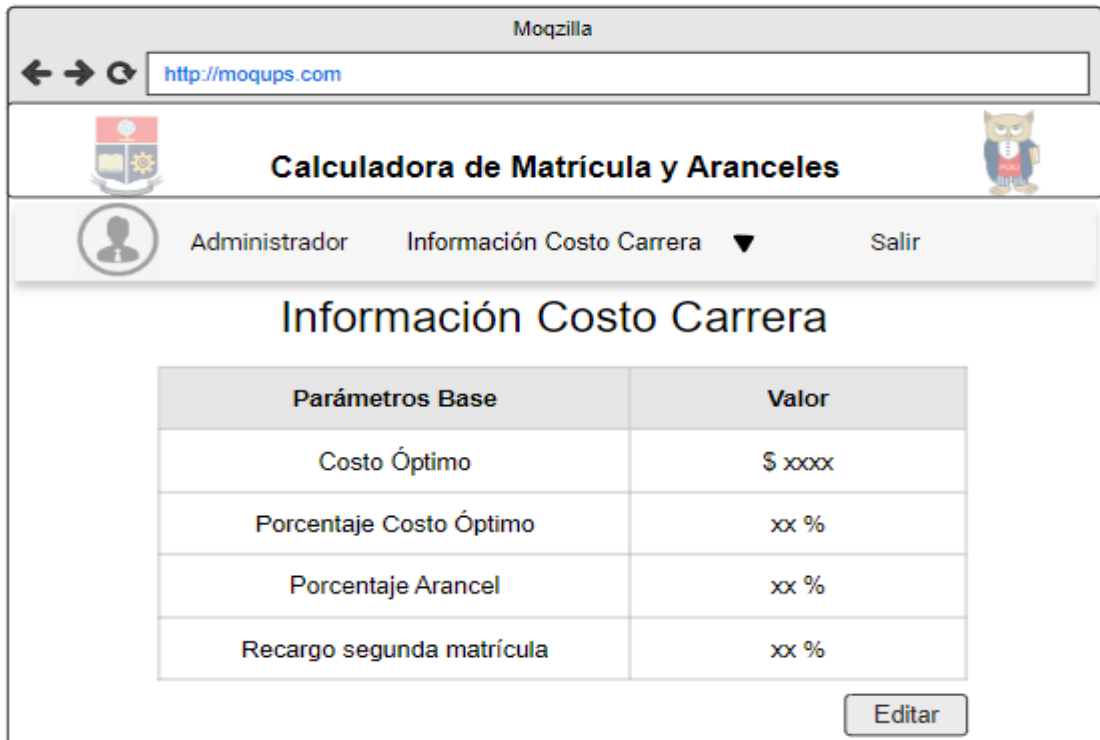


Figura D.1. Sketch para visualizar la información de los parámetros base

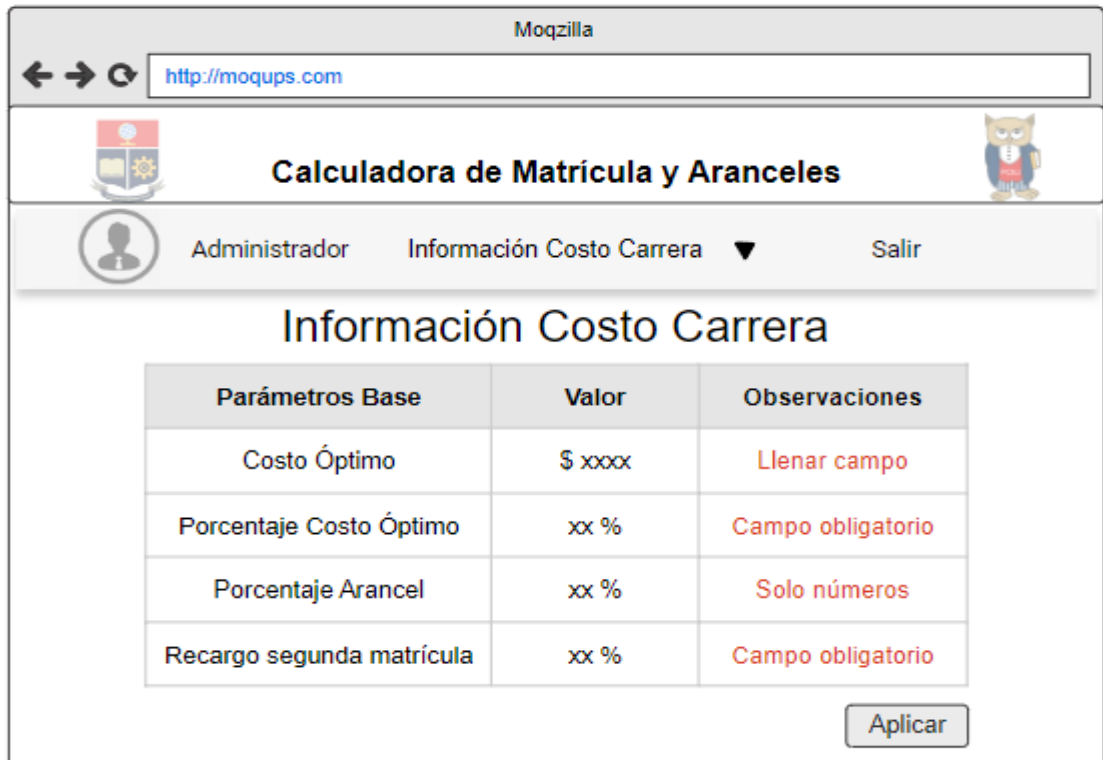




Figura D.2. Sketch para editar la información de los parámetros base

Moqzilla

← → ↻ <http://moqups.com>



## Calculadora de Matrícula y Aranceles



Formación Académica:  Gratuidad:  Si  No

Tipo de Régimen:

Nivel Socioeconómico:

Asignaturas en:

Primera Matrícula:


Segunda Matrícula:

Tercera Matrícula:


**Figura D.3.** Sketch para calcular el valor a pagar por la matriculación en la EPN

Moqzilla

← → ↻ <http://moqups.com>



## Calculadora de Matrícula y Aranceles



### Resultados del cálculo

Matrícula Ordinaria		Matrícula Extraordinaria	
Detalle	Valor	Detalle	Valor
Valor Matrícula	\$ xxx	Valor Matrícula	\$ xxx
Valor Arancel	\$ xxx	Valor Arancel	\$ xxx
Recargo segunda matrícula	\$ xxx	Recargo matrícula extraordinaria	\$ xxx
Total a pagar	\$ xxx	Total a pagar	\$ xxx

**Figura D.4.** Sketch para visualizar el valor a pagar por la matriculación en la EPN

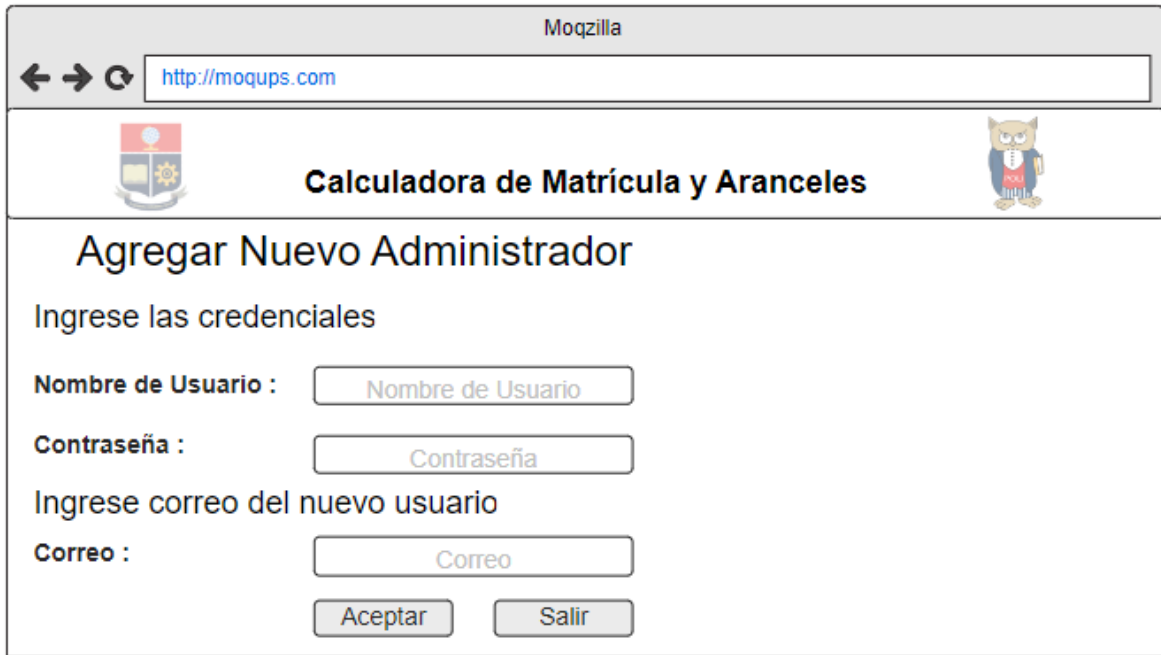


Figura D.5. Sketch para agregar un nuevo administrador

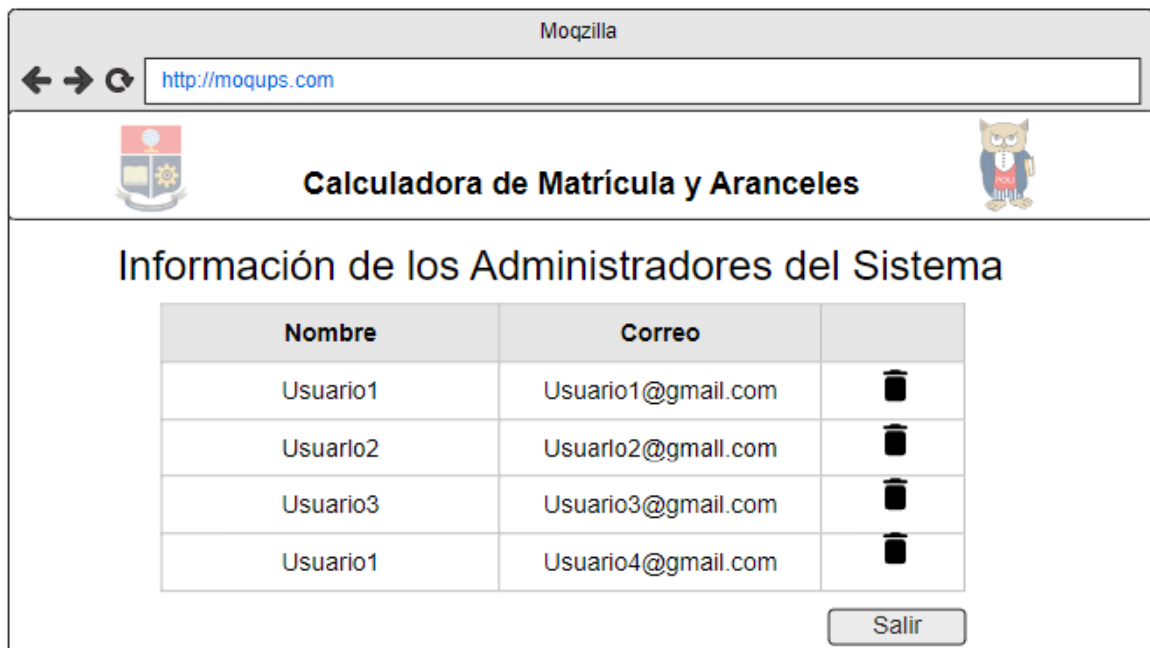


Figura D.6. Sketch para eliminar administradores



## **ANEXO E**

Este anexo se adjunta de manera digital y contiene el script en SQL necesario para la generación de la base de datos del prototipo.

## **ANEXO F**

Este anexo se adjunta de manera digital y contiene los archivos del código fuente creados en el desarrollo del prototipo.

## **ANEXO G**

Este anexo se adjunta de manera digital y contiene el manual de usuario del prototipo.

## **ORDEN DE EMPASTADO**