

Manual Técnico Módulo de TouchScreen

Duván Peñata, Ilian Maussa, Jarol García, Jhordan Flórez, Jonathan Daza, Kevin Gil,

Meijan García, Santiago Cárdenas y Santiago Peñaloza

Facultad de Ingeniería, Universitaria de Colombia

Ingeniería de Sistemas

2023

Contenido

I	Introducción	4
II	Objetivo.....	5
2.1	Objetivos Específicos	5
III	Definición de Tipo de Producto	6
3.1	Software educativo basado en tecnología de pantalla táctil para la enseñanza en estudiantes con capacidades especiales	6
3.2	Diseño De Un Sistema Prototipo De Pantalla De Estructura Touch Screen	6
3.3	Desarrollo de robot móvil para aplicaciones en eventos publicitarios	7
IV	Requerimientos.....	8
4.1	Requerimientos Funcionales.....	8
4.2	Requerimientos No Funcionales.....	10
V	Metodología	11
5.1	Metodología RUP	11
5.2	Metodología SCRUM.....	14
VI	Cronograma.....	15
VII	Presupuesto.....	16
VIII	Definición de actores y roles	17
IX	Mapa de Procesos.....	17
X	Casos de Uso	18
10.1	Caso de Uso de Alto Nivel.....	18
10.2	Caso de Uso Extendido.....	19
10.2.1	Caso de Uso Extendido Certificado.....	19
10.2.2	Caso de Uso Extendido Estado de Cuenta.....	20
10.2.3	Caso de Uso Solicitudes	21
10.2.4	Caso de Uso Extendido Paz y Salvo.....	22
XI	Diagrama de Flujo	23
XII	Modelamiento de la Base de Datos	24
XIII	Diagrama de Clases	34
XIV	Diagrama de Estado.....	35
XV	Pre-Diseño	36
XVI	Arquitectura de Software.....	38
16.1	Arquitectura de Frontend	38
16.1.1	Vista Principal	38

16.1.2	Vista Certificados	39
16.1.3	Vista Paz y Salvos	39
16.1.4	Vista Solicitudes	40
16.1.5	Vista Estado de Cuenta.....	40
16.2	Arquitectura Backend	41
XVII	Librerías y Complementos.....	44
XVIII	Resultados.....	45
XIX	Referencias	47

I Introducción

En el presente manual técnico se visualiza toda la información necesaria para el desarrollo del módulo denominado “*TouchScreen*”, el cual representa el ultimo modulo del proyecto general “*Diseño y Desarrollo de Software para la automatización y mejora de los procesos administrativos y académicos de la Universitaria de Colombia*”. Donde, se establecen los objetivos, requerimientos, esquemas y presupuesto para el mismo y de esta forma cumplir con la satisfacción total del cliente: Universitaria de Colombia, automatizando procesos estudiantiles.

II Objetivo

Desarrollar una aplicación móvil para los estudiantes que se pueda utilizar en un touchscreen, y así los estudiantes puedan auto-gestionar los temas más importantes tanto para ellos como para la universidad.

2.1 Objetivos Específicos

- Implementar los temas más importantes para los estudiantes en un solo lugar, donde puedan acceder desde su celular y hacer petición a cada uno de los servicios que ofrece Universitaria de Colombia.
- Notificar al estudiante sobre las noticias más importantes para los estudiantes de Universitaria de Colombia, además de usar el blog estudiantil “*Entérate*”.
- Generar el carnet virtual para los estudiantes.
- Automatizar el proceso de visualización de notas de forma factible y rápida.
- Automatizar los procesos de certificados y pagos, para que los estudiantes puedan realizarlos desde el dispositivo móvil.
- Automatizar el proceso de gestión estudiantil, mediante el uso de un QR otorgado para cada estudiante en su carnet virtual.

III Definición de Tipo de Producto

3.1 Software educativo basado en tecnología de pantalla táctil para la enseñanza en estudiantes con capacidades especiales

El objetivo de este proyecto es implementar un software basado en pantallas táctiles para la enseñanza de los estudiantes con capacidades especiales de la Unidad educativa de sordos y del Instituto de educación especial de Ibarra, Ecuador con la finalidad de facilitar la enseñanza tanto en las aulas de clase como en el hogar. El software permitirá mejorar el aprendizaje de los estudiantes con capacidades especiales. Se presenta una encuesta aplicada a personas involucradas a través del cual se identifican las necesidades.



3.2 Diseño De Un Sistema Prototipo De Pantalla De Estructura Touch Screen

Los Objetivos de este proyecto se dividen en:

- Aplicar una metodología para el desarrollo del proyecto de tesis.
- Realizar la documentación necesaria del producto que se desea hacer para así tener la información necesaria para la elaboración de este sistema de información.

- Investigar sobre las diferentes tecnologías que existen para el diseño y construcción de las pantallas táctiles.
- Evaluar las empresas existentes que se dedican a diseñar y construir las pantallas de estructura touch screen.



3.3 Desarrollo de robot móvil para aplicaciones en eventos publicitarios

Se diseñó y desarrolló un conjunto de prototipos de robots semi-humanoides que puedan moverse en recintos cerrados con presencia humana dotados de sensores para su navegación y dispositivos de divulgación de información. Estos diseños se han construido con estrategia de teleoperación a través de redes Wi-Fi, Zigbee, entre otros. La interacción del dispositivo con el ser humano se lleva a cabo a través de sensores de movimiento, pantalla touchscreen, animatrónica, entre otros.



IV Requerimientos

4.1 Requerimientos Funcionales

Nombre	RF1 - Paz y salvos
Resumen	En este apartado se podrá obtener un certificado el cual denotará que el usuario se encuentra al día con los pagos de la institución, este se podrá obtener por medio de una impresión.
Entradas	
Número de identificación del usuario, semestre, carrera	
Resultados	
Impresión de Certificado Paz y salvos	

Nombre	RF2 - Estado de cuenta
Resumen	En este apartado se podrá validar como se encuentra el usuario referente a los pagos de la institución, se podrá verificar las cuotas pendientes, pagadas o próximas a pagar
Entradas	
Número de identificación del usuario, semestre, carrera	
Resultados	
Visualización de datos	

Nombre	RF3 – Gestión de seguridad y acceso a la aplicación
Resumen	El sistema requiere gestionar el ingreso al sistema junto a los permisos de cada usuario con los cuales tendrá diferentes opciones a la hora de la recreación de actividades y evaluación de actividades
Entradas	
Identificación, Nombre, Apellido, Cedula, Teléfono, Rol, Email, Contraseña	
Resultados	
Datos de usuario	

Nombre	RF4 – Base de Datos
Resumen	El sistema requiere debe utilizar la misma base de datos que se utiliza para la plataforma de notas, centralizando la misma para cada funcionalidad.
Entradas	
API	
Resultados	
Recepción y envío de datos	

Nombre	RF5 - Certificados
Resumen	En este apartado se podrá obtener un certificado dependiendo de la necesidad del usuario, el cual según el pago realizado de este se podrá obtener de manera impresa
Entradas	
Tipo de certificado, carrera, periodo	
Resultados	
Impresión de Certificados	

Nombre	RF6 - Solicitudes
Resumen	Se podrán realizar los diferentes tipos de solicitudes (homologaciones, aplazamientos, inscripción de materias, entre otras) donde se enviará a un Google spreadsheet
Entradas	
Tipo de Solicitud, periodo, carrera, mensaje, archivo	
Resultados	
Creación y envío de solicitud	

4.2 Requerimientos No Funcionales

N	Tipo	Descripción
1	Producto	La aplicación debe funcionar correctamente en la mayoría de dispositivos móviles sin requerir de un hardware con alto nivel de computo
2	Producto	La aplicación debe ser compatible con IOS y Android
3	Producto	La aplicación debe tener una alta integración con diferentes lenguajes teniendo como base el inglés y el español
4	Producto	La aplicación debe tener una interfaz altamente amigable e interactiva que facilite el entendimiento de las funciones que se encuentren dentro de esta
5	Seguridad	El sistema cerrará la sesión del usuario luego de cerrar la aplicación
6	Usabilidad	El diseño debe adaptarse a la pantalla de la mayoría de dispositivos (smartphones, tablets)
7	Legal	Garantizar al usuario, el pleno y efectivo ejercicio del derecho de hábeas data la cual trata sobre el derecho que tiene toda persona de conocer, actualizar y rectificar toda aquella información que se le relacione.
8	Infraestructura	La aplicación deberá tener como limite el consumo de 500 Mb de RAM para evitar la ralentización del dispositivo
9	Infraestructura	La aplicación deberá usar apropiadamente el CPU del dispositivo para no afectar tareas básicas del dispositivo móvil
10	Infraestructura	La aplicación deberá estar optimizada para requerir del menor espacio posible a la hora de instalarse

V Metodología

5.1 Metodología RUP

El "Rational Unified Process", o en español, El Proceso Unificado de Rational, desarrollado por Philippe Kruchten, Ivar Jacobson y otros como complemento al UML. Es el proceso que cuenta y tiene la capacidad de acomodar una gran variedad de procesos, según describe Benedicto Canaza, en su libro artículo "Ingeniería de Software, Metodologías ágiles RUP" en el 2009. En el presente proyecto la metodología RUP se utilizó para la construcción de los manuales de usuario, técnicos y en si toda la documentación del proyecto en general. Se baso al mismo tiempo en las cuatro fases fundamentales para el ciclo de vida del software descrito por los creadores de RUP:

1. **Incepción:** Donde se sintetiza la visión del proyecto, limitando los alcances e identificando los procesos más importantes, precisos y críticos, además, de tener en cuenta el conocimiento en los costos, cronograma y riesgos
2. **Elaboración:** Según la misma metodología RUP, en esta fase, se debe analizar el problema y definir el proyecto estructurando su arquitectura, infraestructura y consolidando los requerimientos del proyecto, utilizando casos de uso y otros diagramas UML para complementar el análisis general de los riesgos y gastos planteados desde el inicio de la fase.
3. **Construcción:** Todos los componentes definidos de la fase anterior, se empiezan a implementar y verificar, teniendo en cuenta el control de costos, calendario y calidad. Generando las versiones generales de cada etapa de prueba como ALFA, BETA... Entre otros. Por último, se compila una versión de entrega probada y facturada.
4. **Transición:** Se le conoce como transición, ya que el software pasa a las manos del usuario final, a este producto se le corrigen los últimos errores y se agregan los detalles

minuciosos para el proyecto, además, se entrenan los usuarios y el software se empieza a implementar y distribuir.

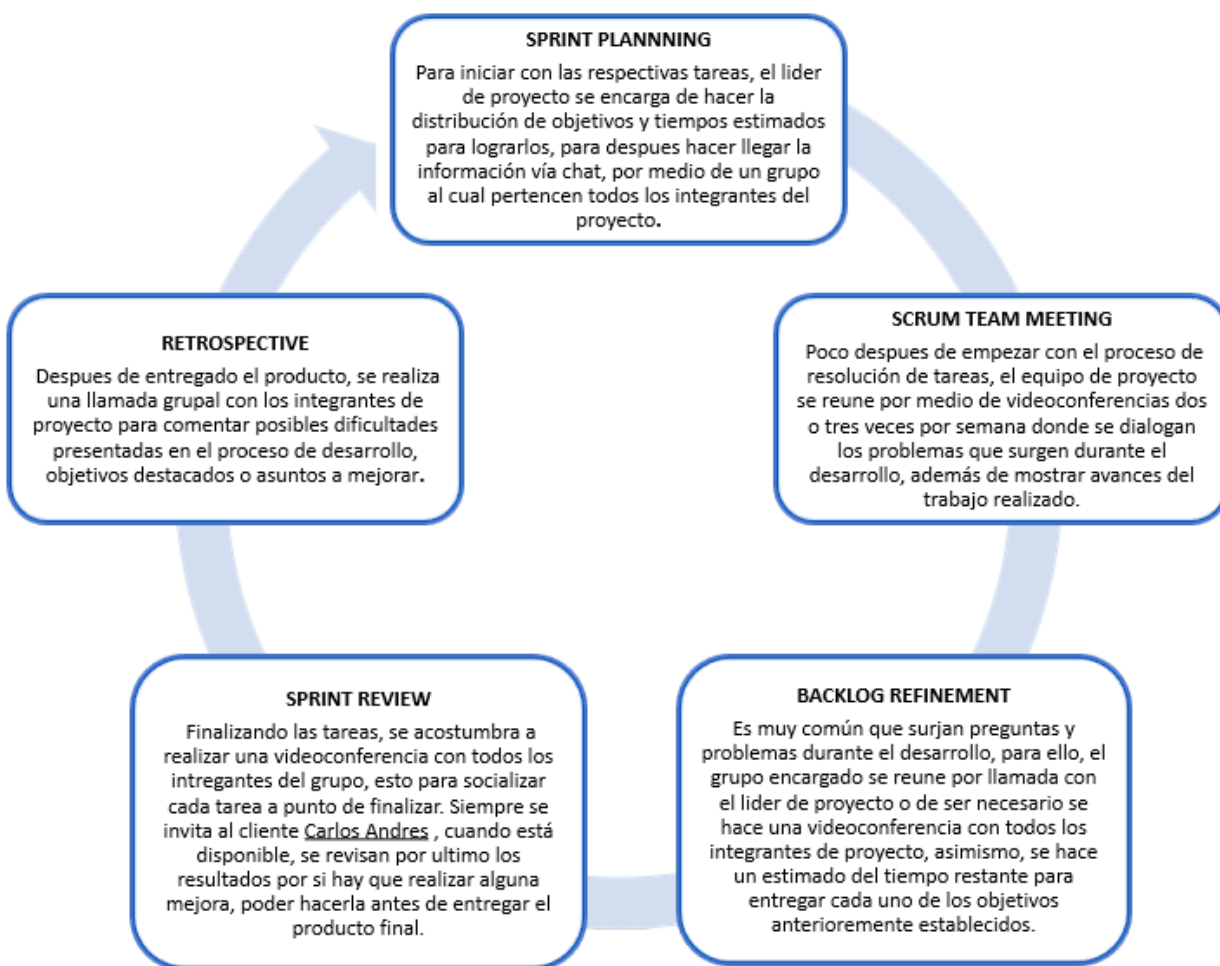
Fase	Nro. Iteraciones	Duración
Fase de Inicio	1	12 semanas
Fase de Elaboración	1	8 semanas
Fase de Construcción	1	16 semanas
Fase de Transición	1	4 semanas

Descripción	Hito
Fase de Inicio	<p>Durante esta fase de iniciación, se deben iniciar las siguientes actividades y/o flujos de trabajo y sus respectivos productos</p> <ul style="list-style-type: none"> ● Gestión de Proyectos: Plan de Desarrollo de Software (este documento) y Lista de Riesgos ● Requisitos: se elaborará un glosario, un modelo de casos de uso y la visión del proyecto <p>Basándose en las entrevistas hechas al cliente y posteriormente de modelarlos mediante diagramas y casos de uso se realiza el desarrollo de esta fase.</p>
Fase de elaboración	En la presente fase de elaboración el modelo conceptual de la base

	<p>de datos y el prototipo no funcional del software se revisan y perfeccionan. Los procesos que se iniciaron antes de esta fase se continuaron analizando hasta iniciar las siguientes actividades y/o flujos de trabajo:</p> <ul style="list-style-type: none"> ● Análisis / diseño: Modelo conceptual de datos. ● Implementación: Prototipos no funcionales de interfaces de usuario. <p>Estos 2 flujos, en las siguientes fases se deben ir refinando cada iteración</p>
<p>Fase de Construcción</p>	<p>Durante la fase de construcción, se realiza el análisis y diseño de casos de uso. Esta fase termina cuando la aplicación es funcional y operativa, cumpliendo con los requisitos al inicio del proyecto.</p>
<p>Fase de Transición</p>	<p>En esta etapa, se han preparado dos versiones para garantizar la implementación completa y los cambios en el sistema anterior, incluida la prueba de la experiencia del usuario y el suministro de documentación para servir como guías de instalación y funcionalidad del software.</p> <p>En esta fase se realiza el despliegue cada uno de los cinco aplicativos.</p>

5.2 Metodología SCRUM

Por los tiempos de entrega, durante el desarrollo de este proyecto y cada uno de sus correspondientes módulos, fue necesario utilizar la metodología SCRUM, una metodología, una metodología ágil desarrollada por **Jeff Sutherland y Ken Schwaber**, donde se trabajaba constantemente en equipo colaborativamente utilizando buenas prácticas para obtener los mejores resultados posibles.



VII Presupuesto

PRESUPUESTO		
Etapa	%Estimado/prioridad	Días
Analisis	20%	58
Diseño Conceptual/Logico	30%	107
Quality Assurance	30%	107
Pruebas de usuario	10%	30
Implementación	10%	20
Desarrollo	total	322
Nivel de incertidumbre	10%	80
	Total	402

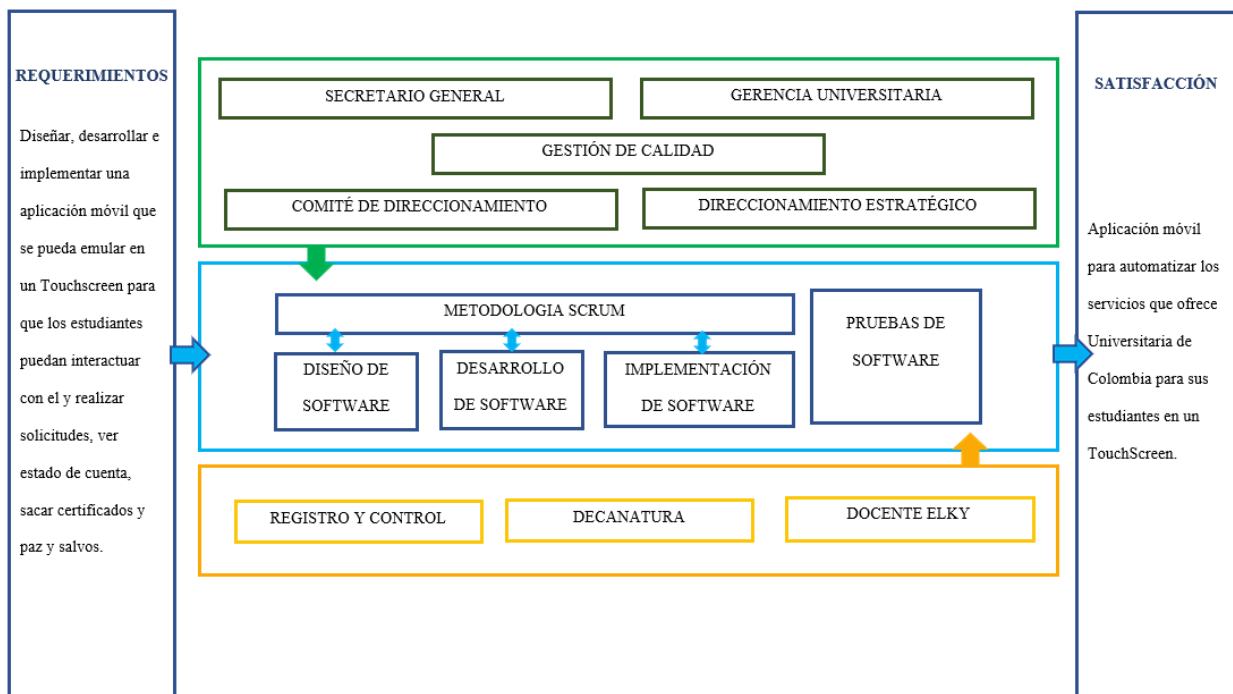
Costo Diario	\$40,000.00
Valor Total	\$16,080,000.00

PRESUPUESTO IMPLEMENTACIÓN	
Descripción	Costo
Impresora multifuncional HP laser	\$1,219,000.00
Licencia de SQLServer	\$2,280,000.00
Licencia de Visual Studio 2022	\$3,050,000.00
Servidor Sommee, especial para ASP NET	\$1,500,000.00
Total	\$8,049,000.00

VIII Definición de actores y roles

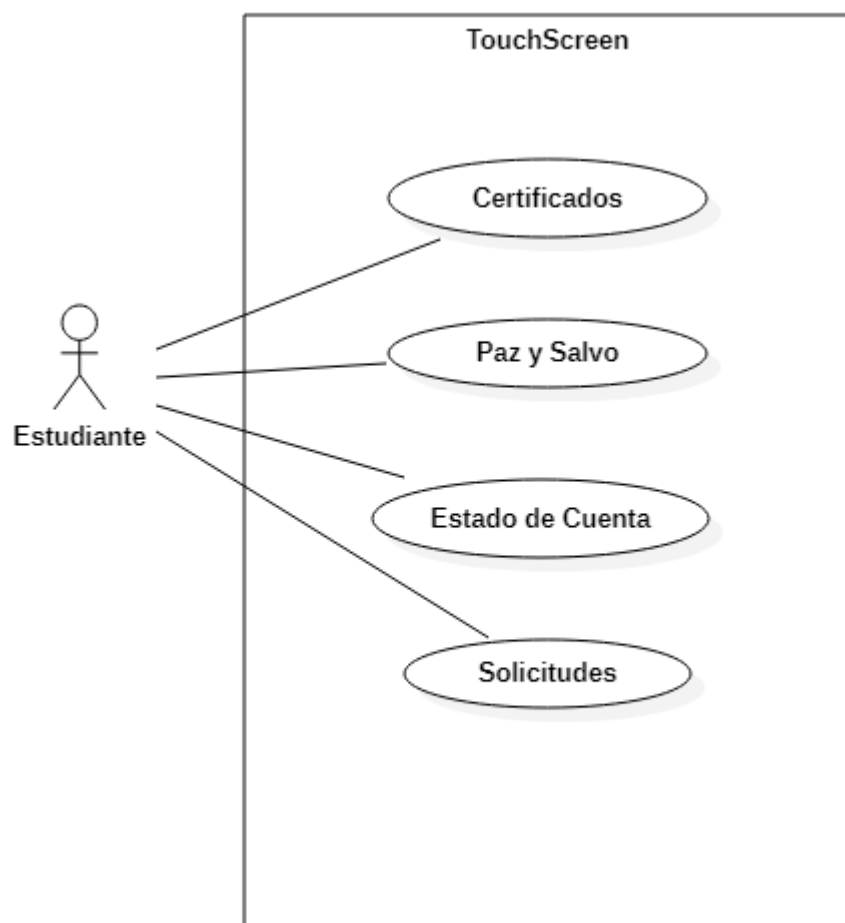
Actor	Estudiante
Descripción	El estudiante puede realizar las distintas acciones que se encuentran en la aplicación una vez inicie sesión
Responsabilidades	<ul style="list-style-type: none"> · Consultar el estado de sus pagos · Consultar notas · Ver noticias actualizadas en la universitaria de Colombia · Visualizar su carné de manera digital

IX Mapa de Procesos



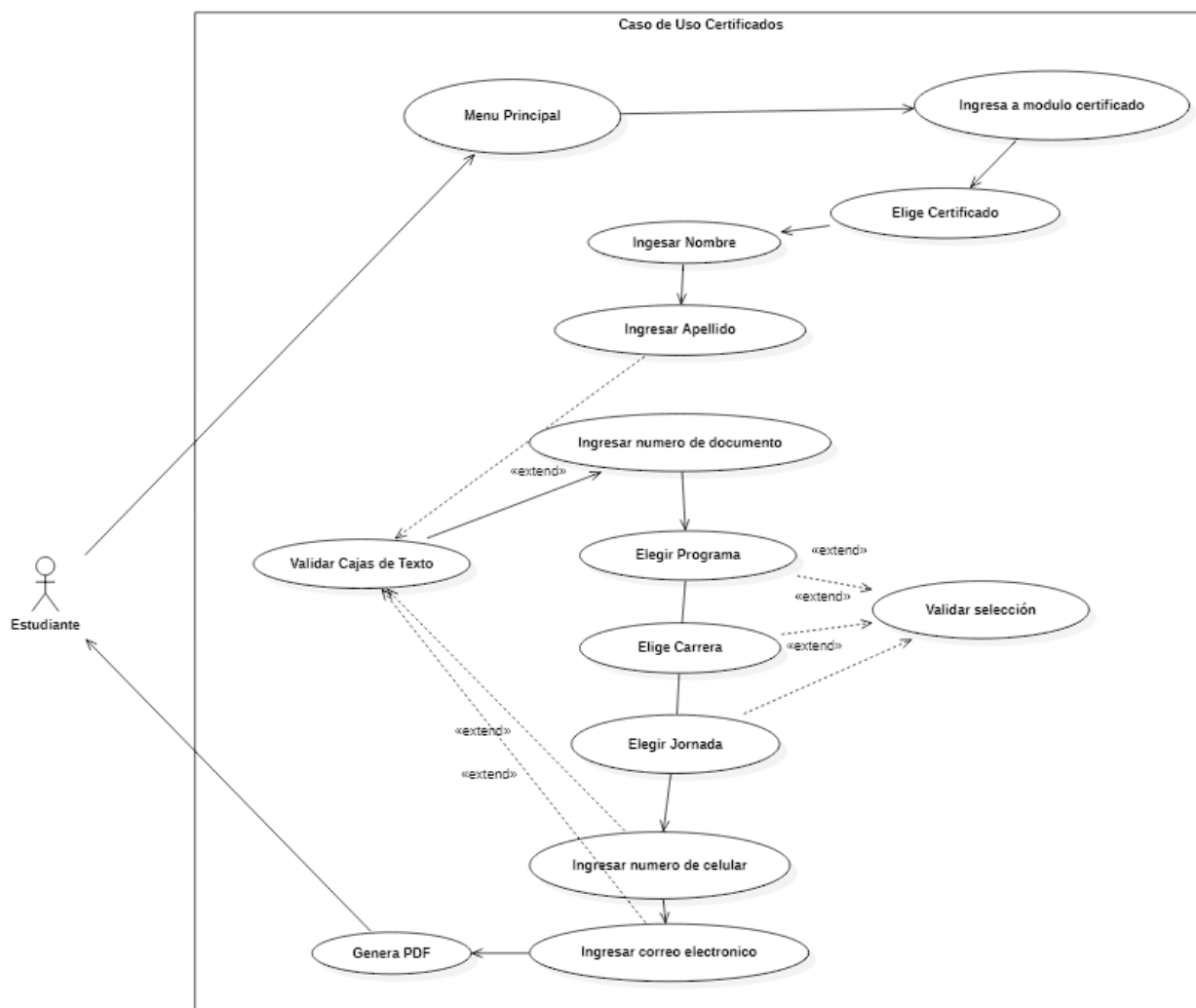
X Casos de Uso

10.1 Caso de Uso de Alto Nivel

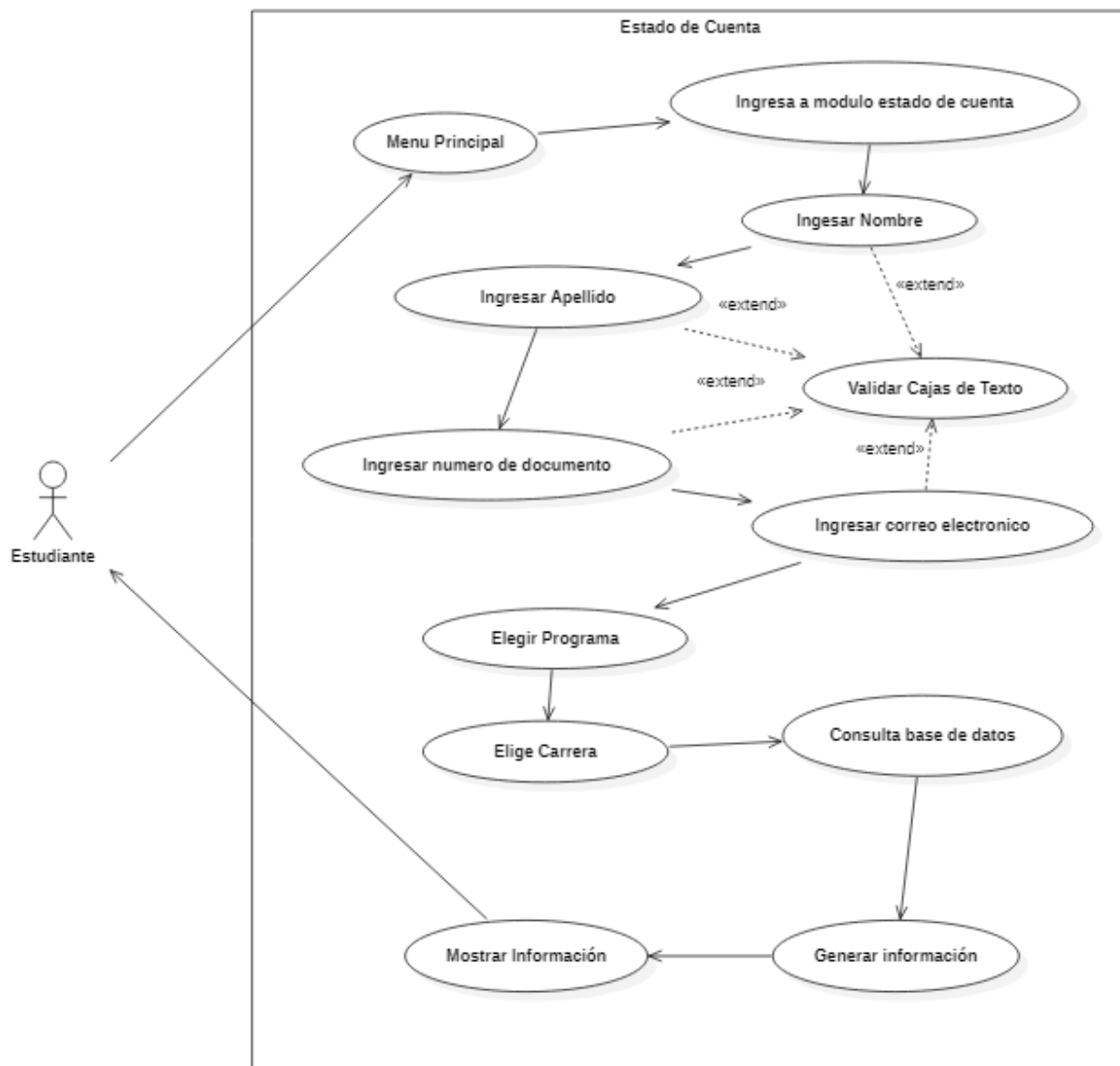


10.2 Caso de Uso Extendido

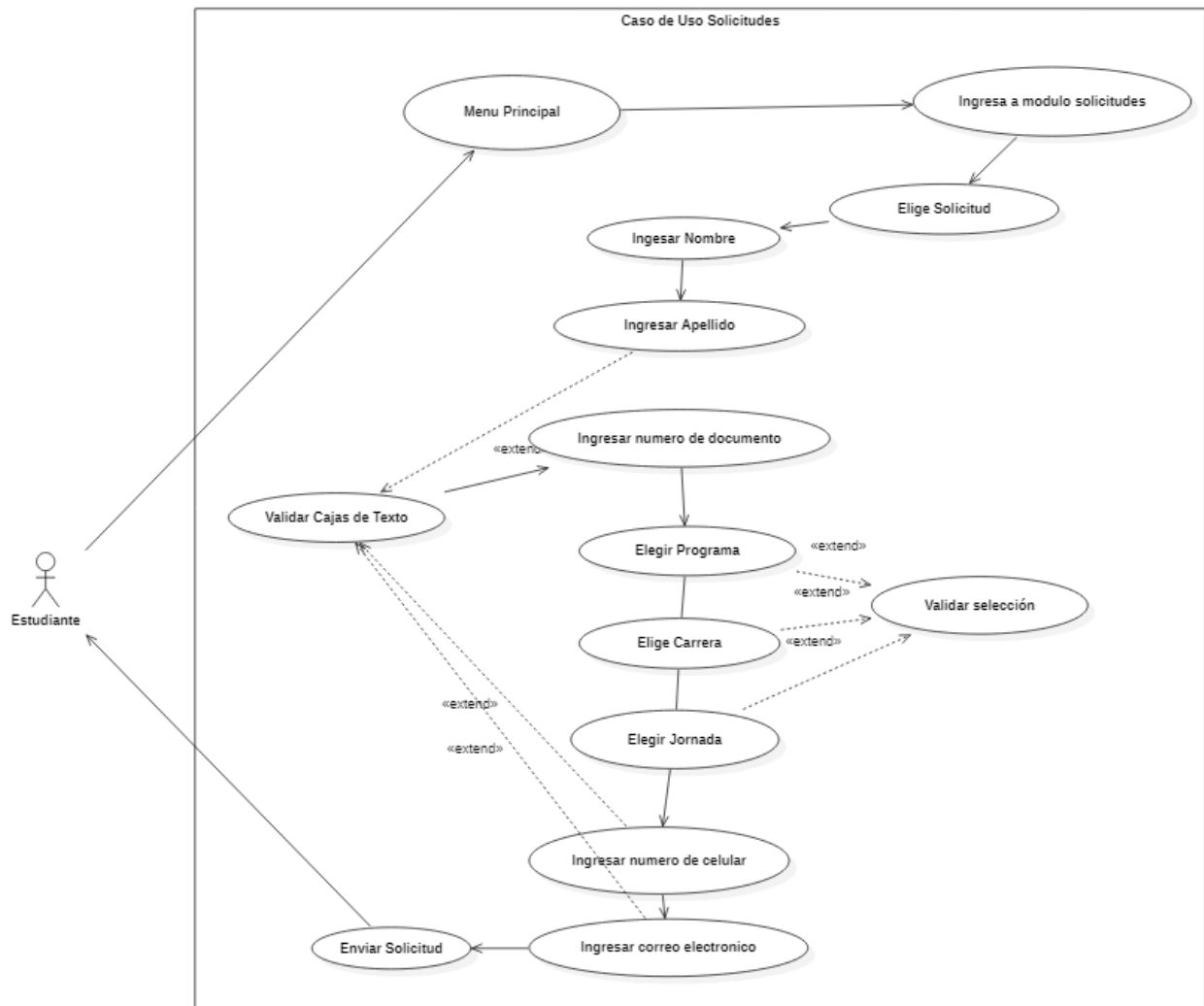
10.2.1 Caso de Uso Extendido Certificado



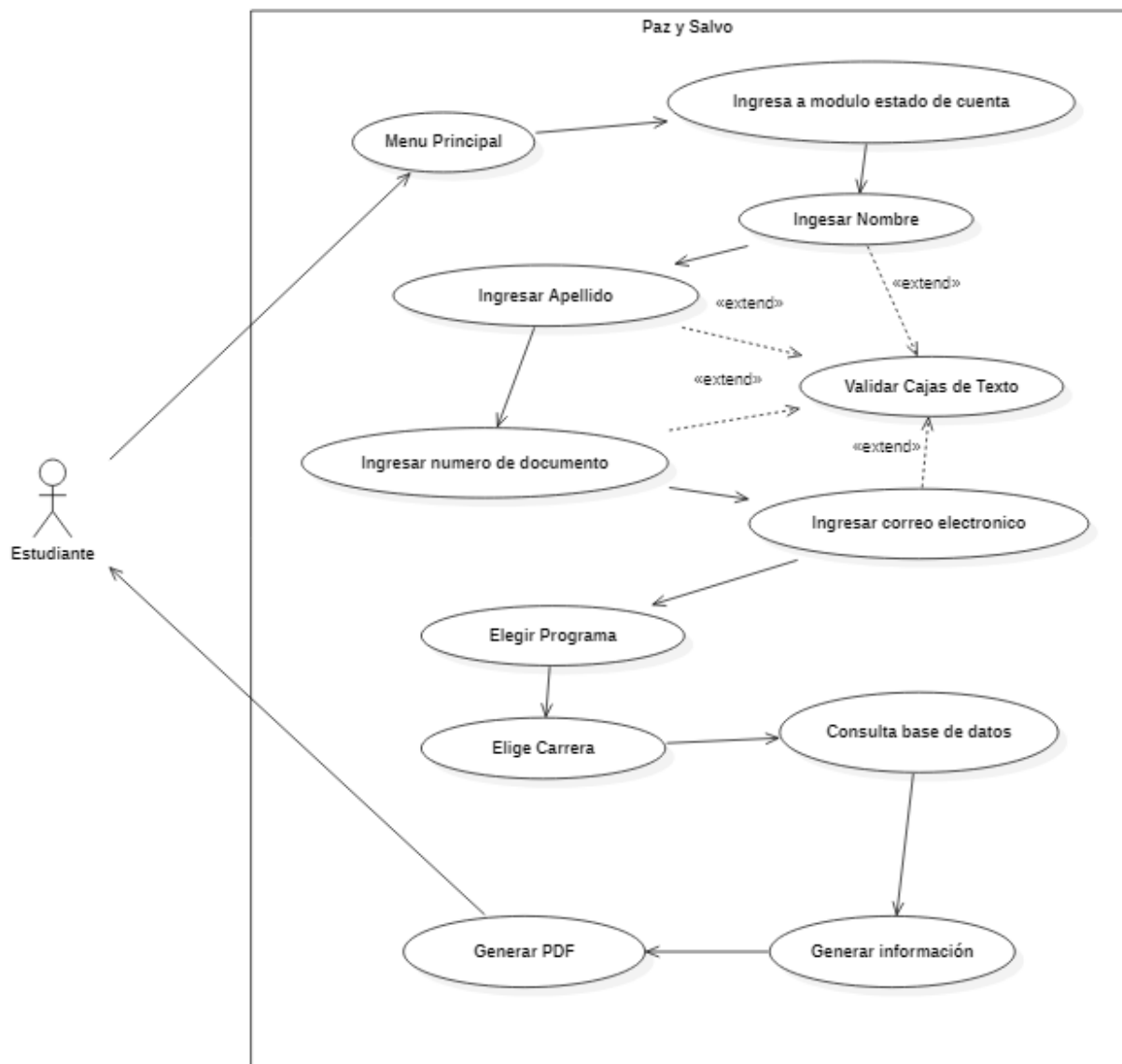
10.2.2 Caso de Uso Extendido Estado de Cuenta



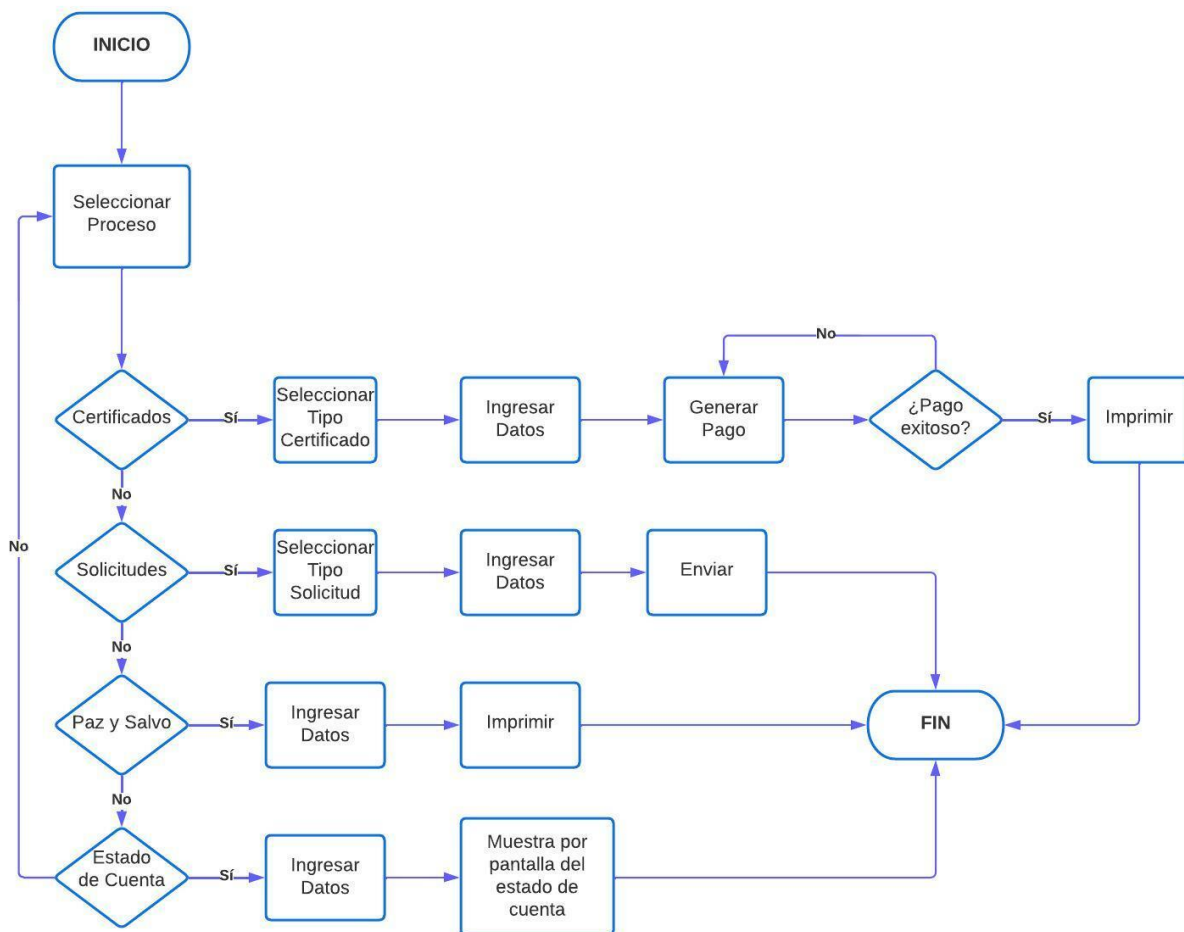
10.2.3 Caso de Uso Solicitudes



10.2.4 Caso de Uso Extendido Paz y Salvo



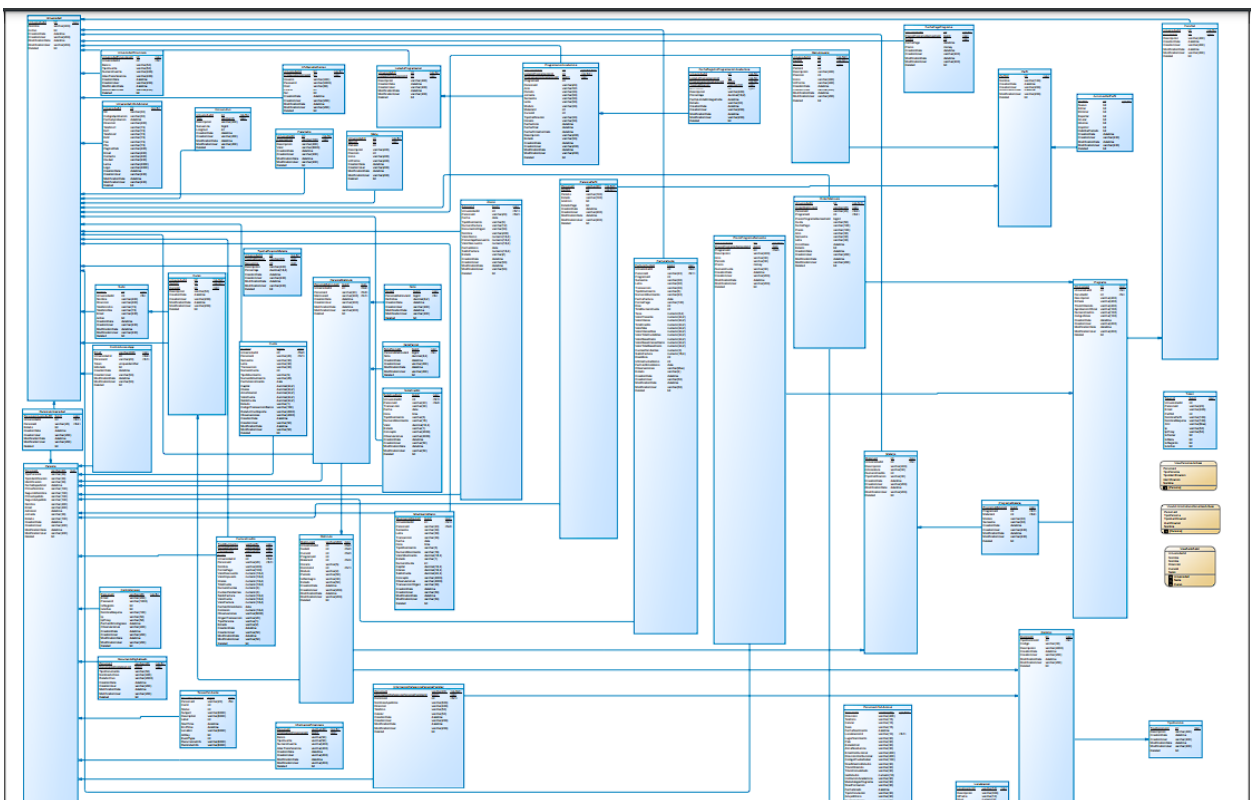
XI Diagrama de Flujo

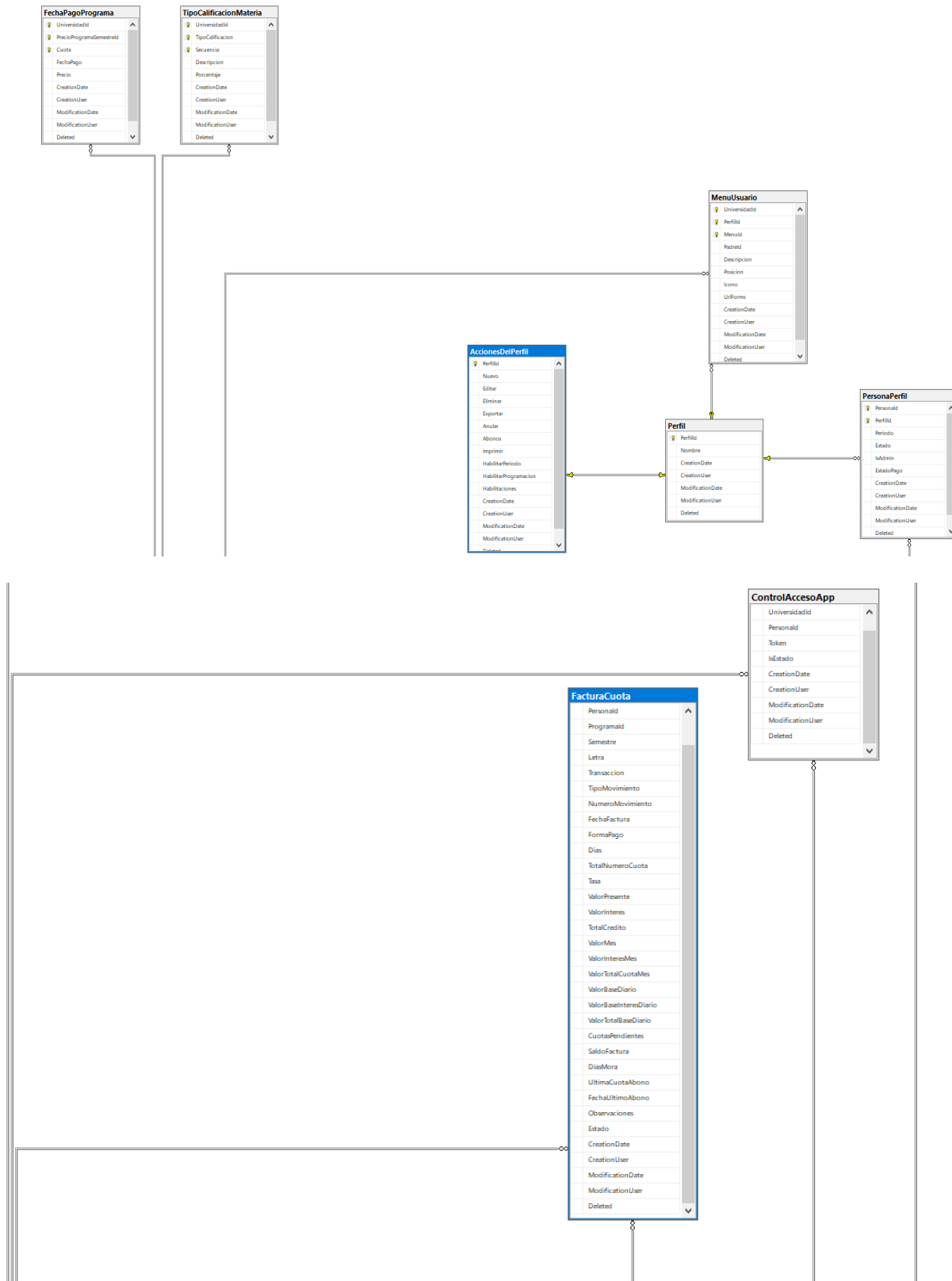


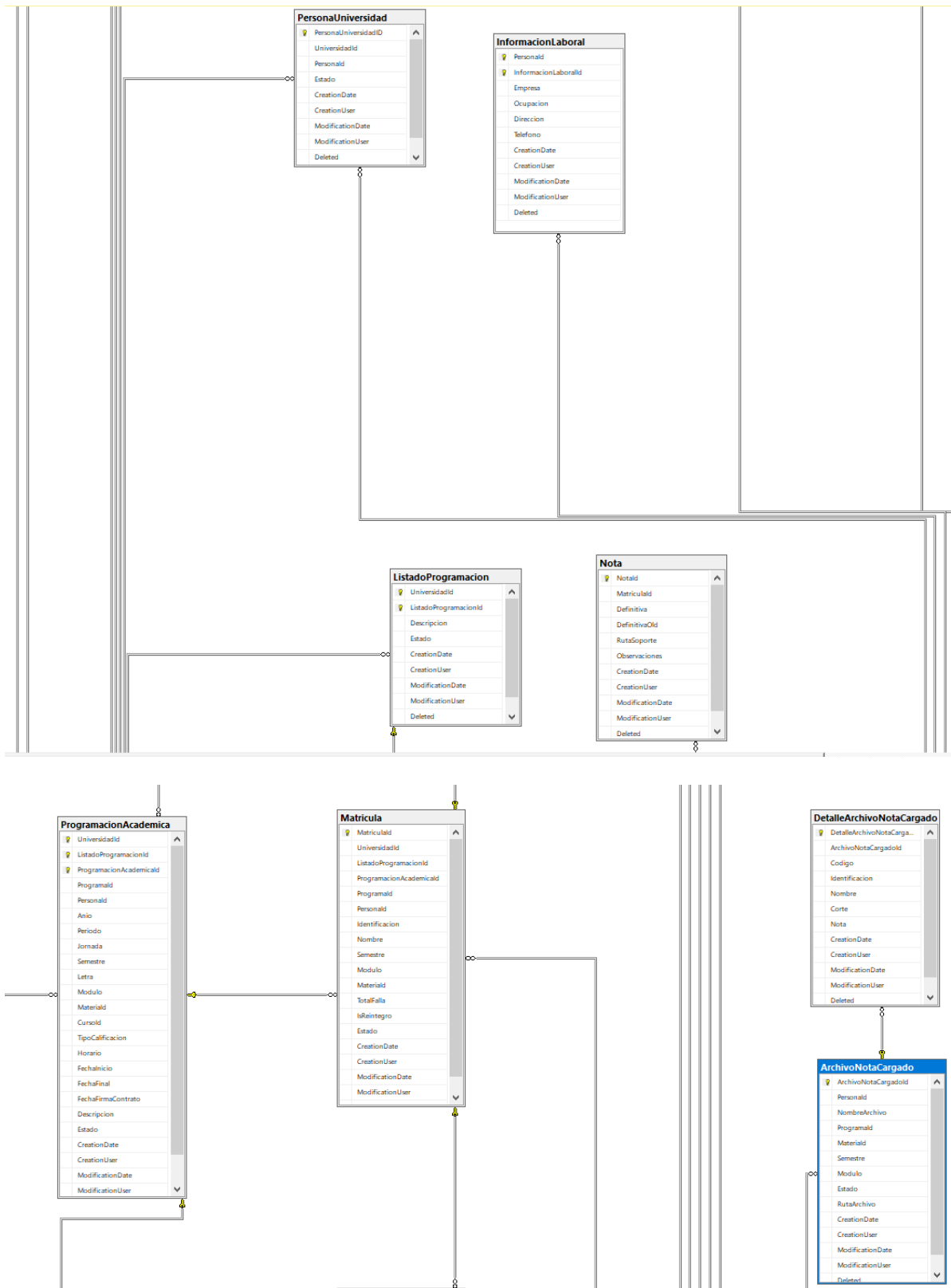
XII Modelamiento de la Base de Datos

Al ser un software que se utiliza específicamente para la Universitaria de Colombia, fue necesario usar la base de datos establecida por la misma, a la cual se le hicieron algunas modificaciones referentes a funcionalidad, confidencialidad y accesibilidad, sin embargo, los cambios fueron sumamente difíciles de desarrollar por la complejidad de la base de datos.

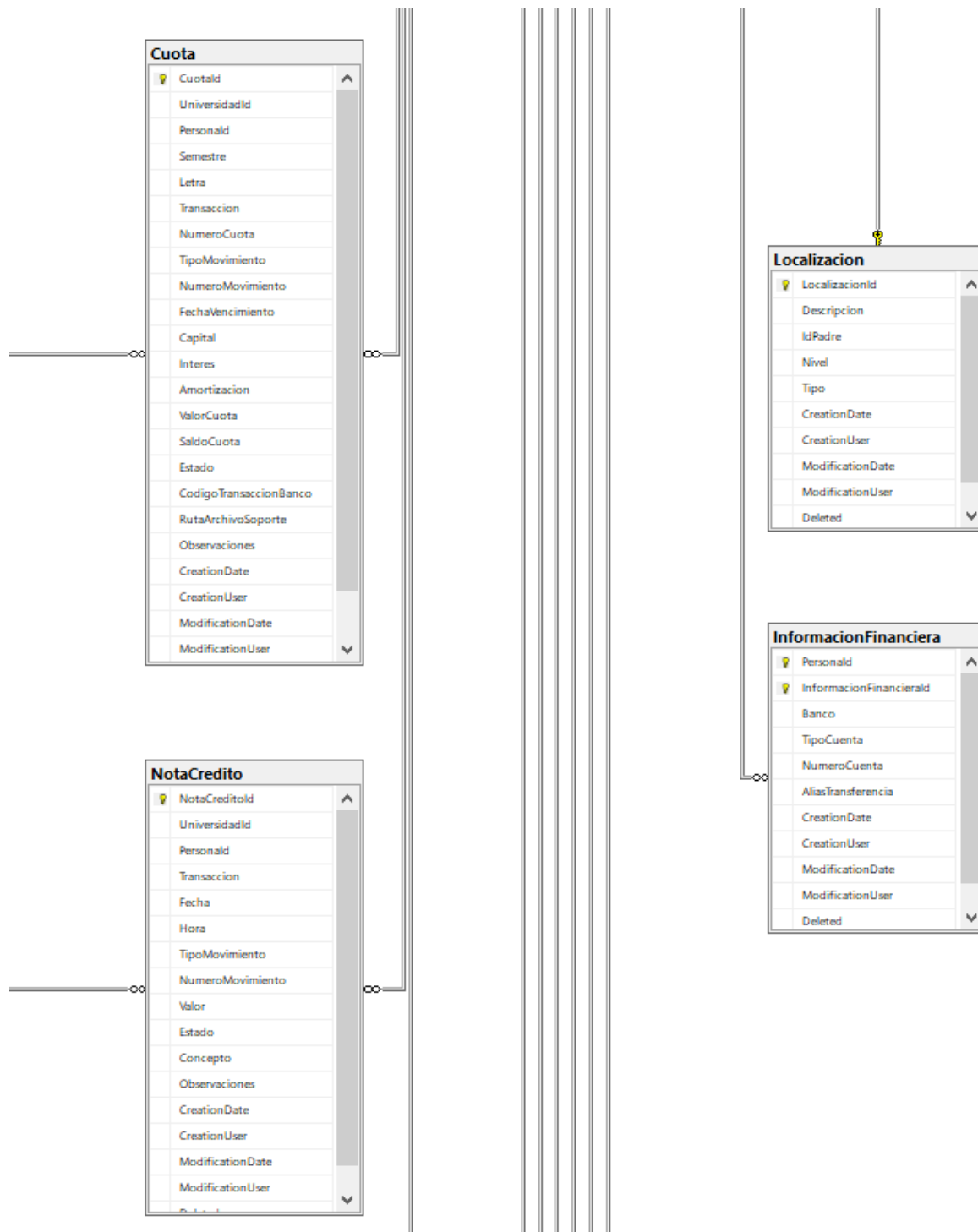
Modelo General:

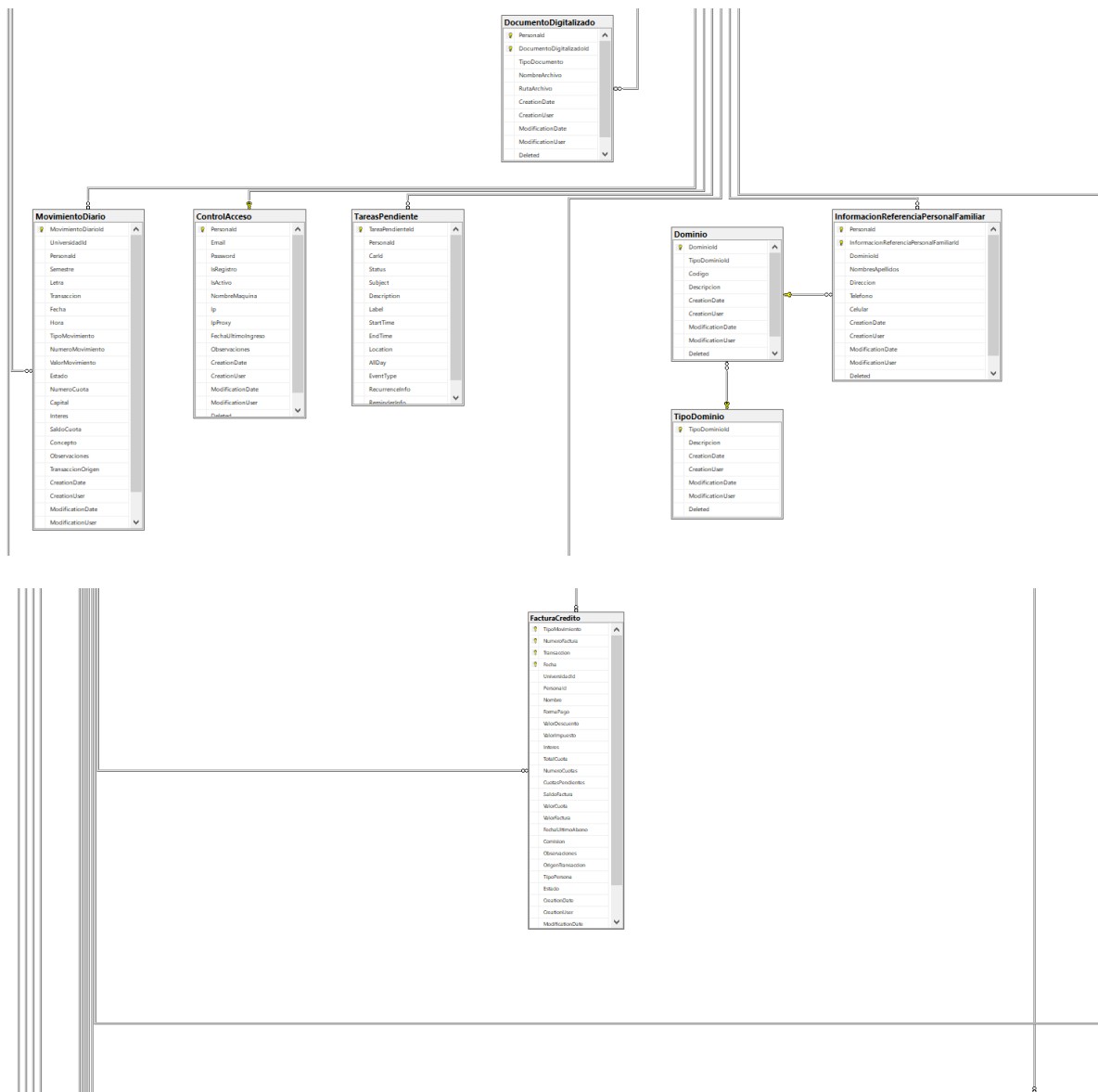


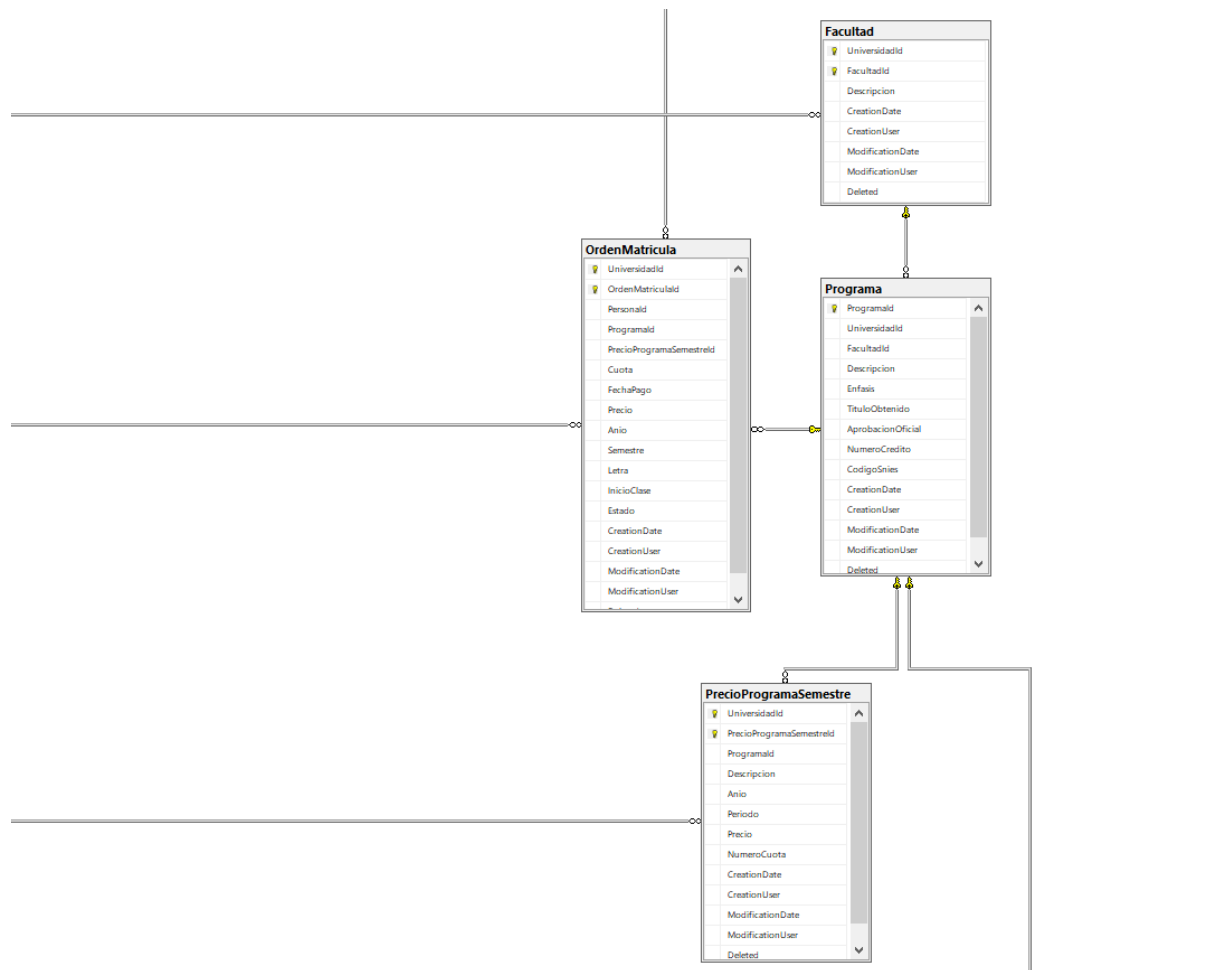


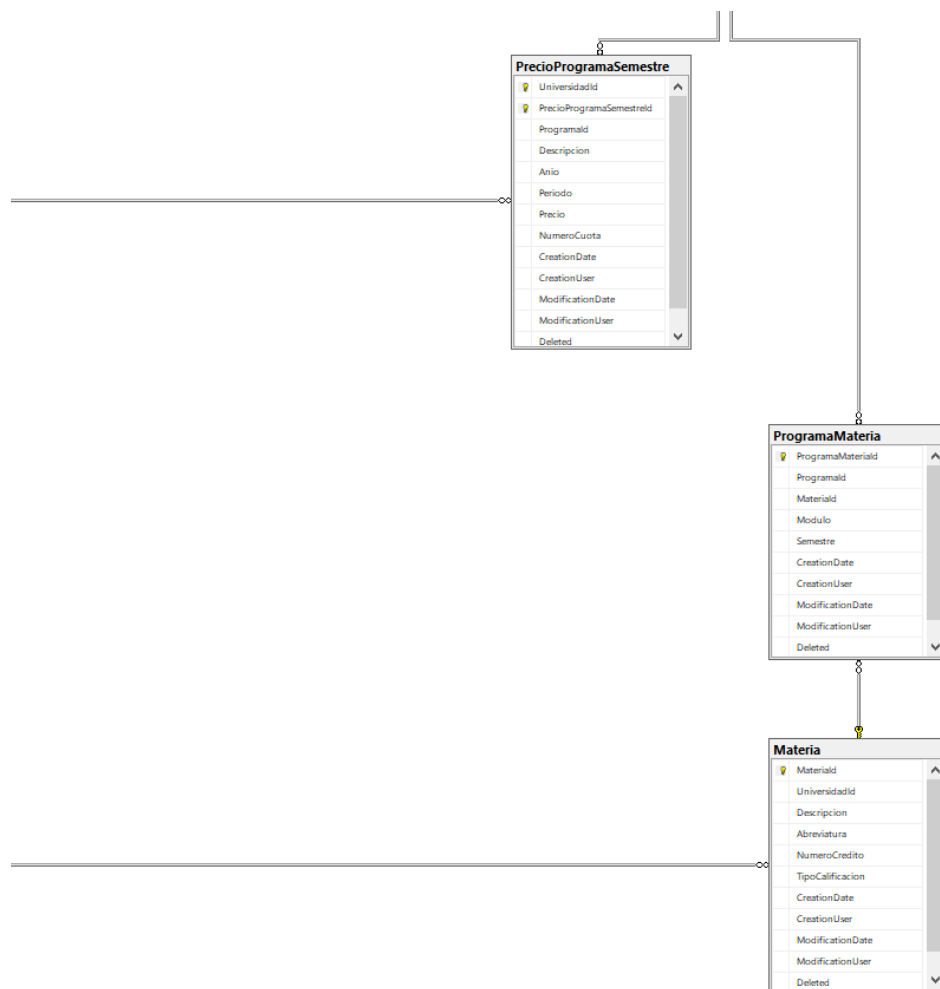


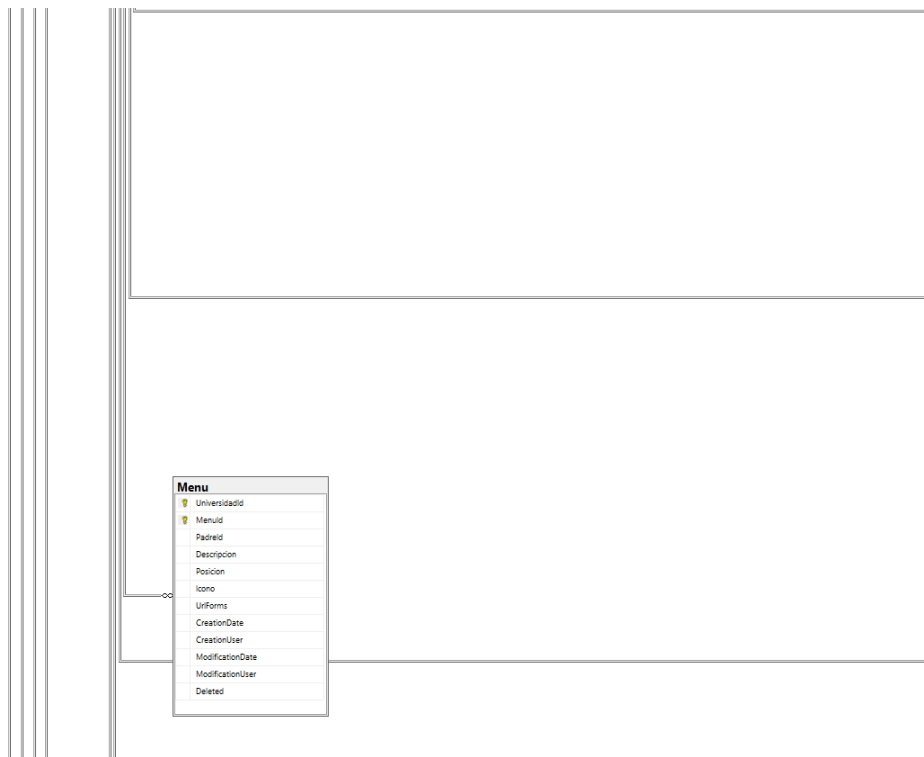
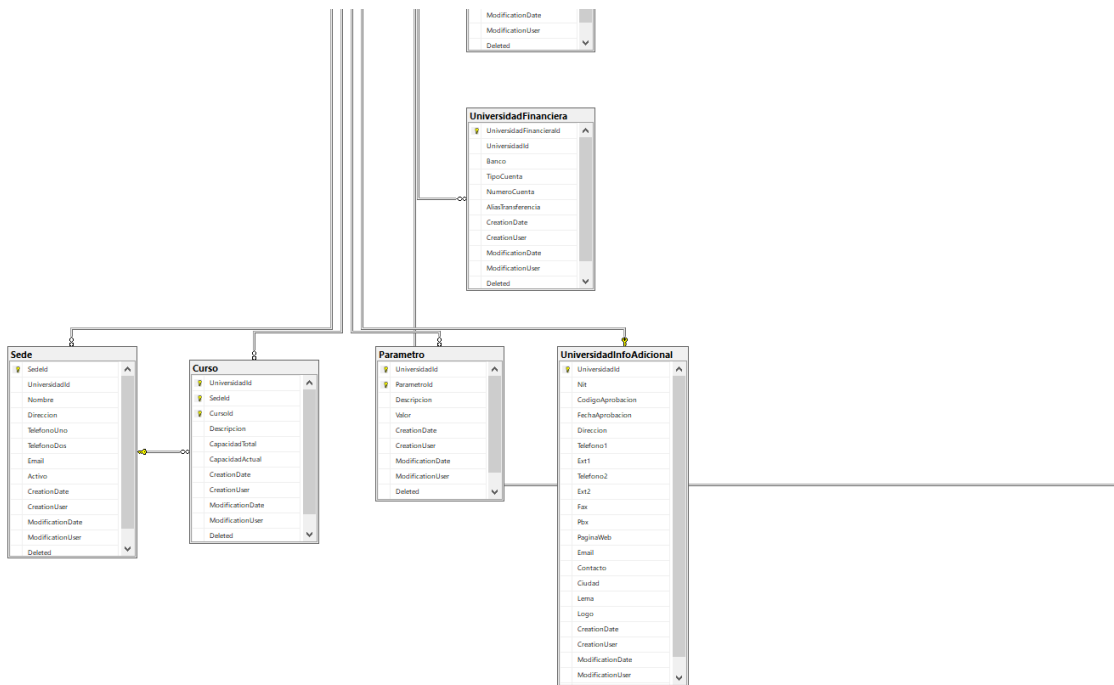


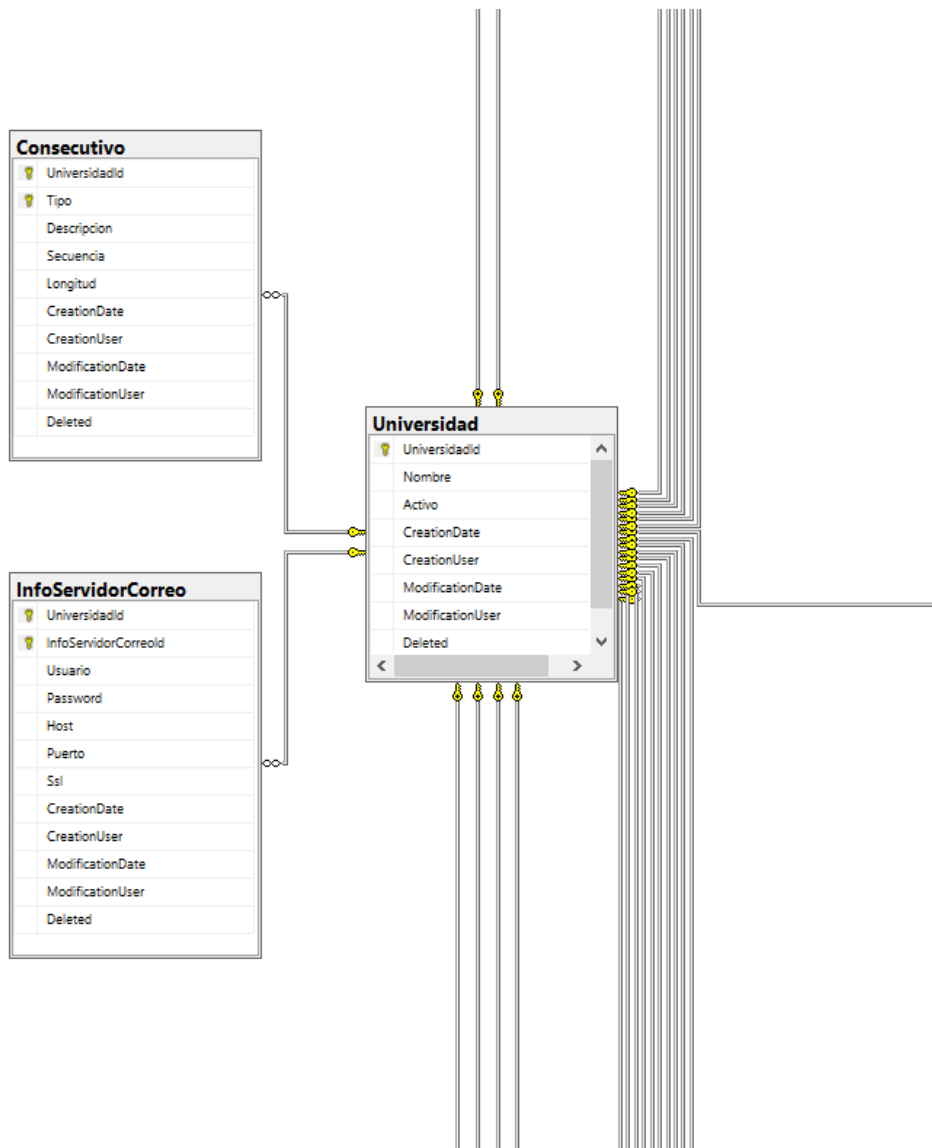




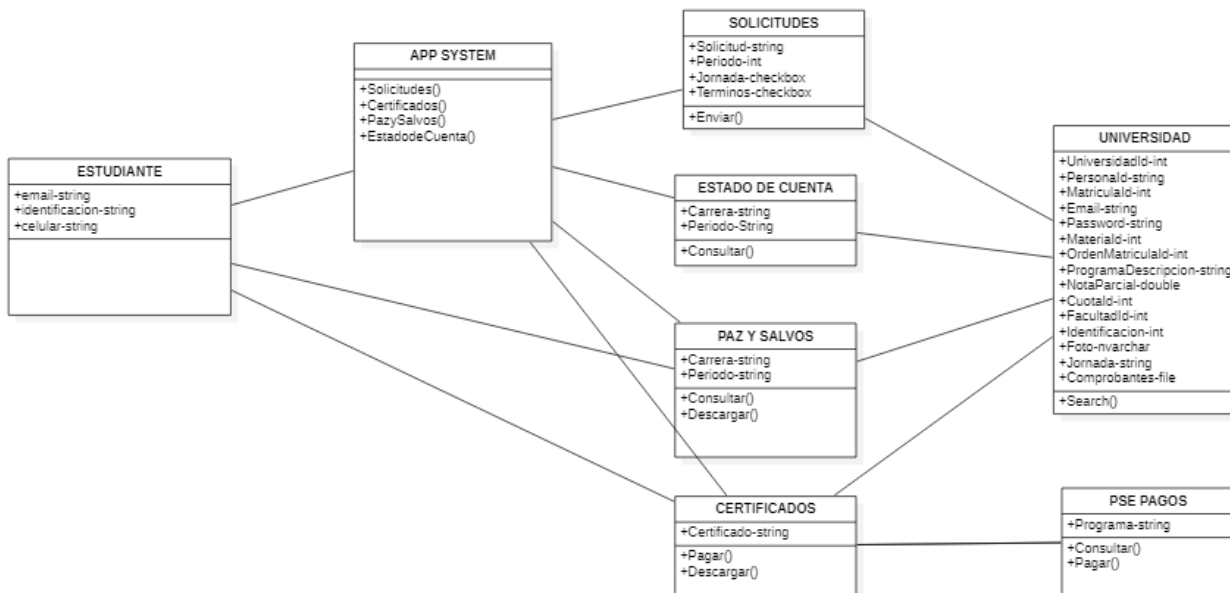




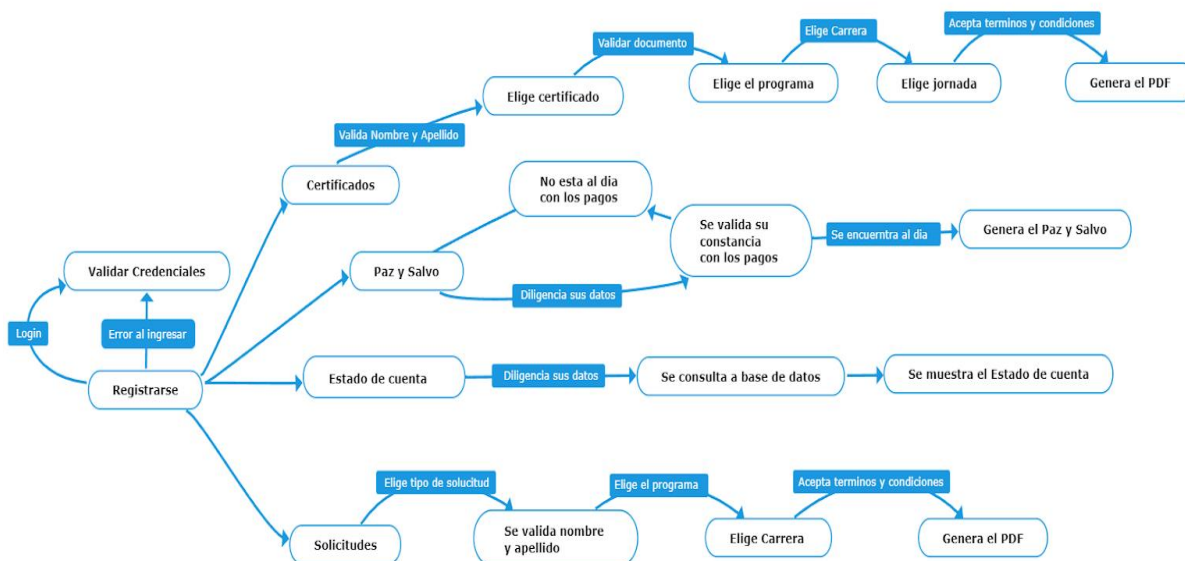




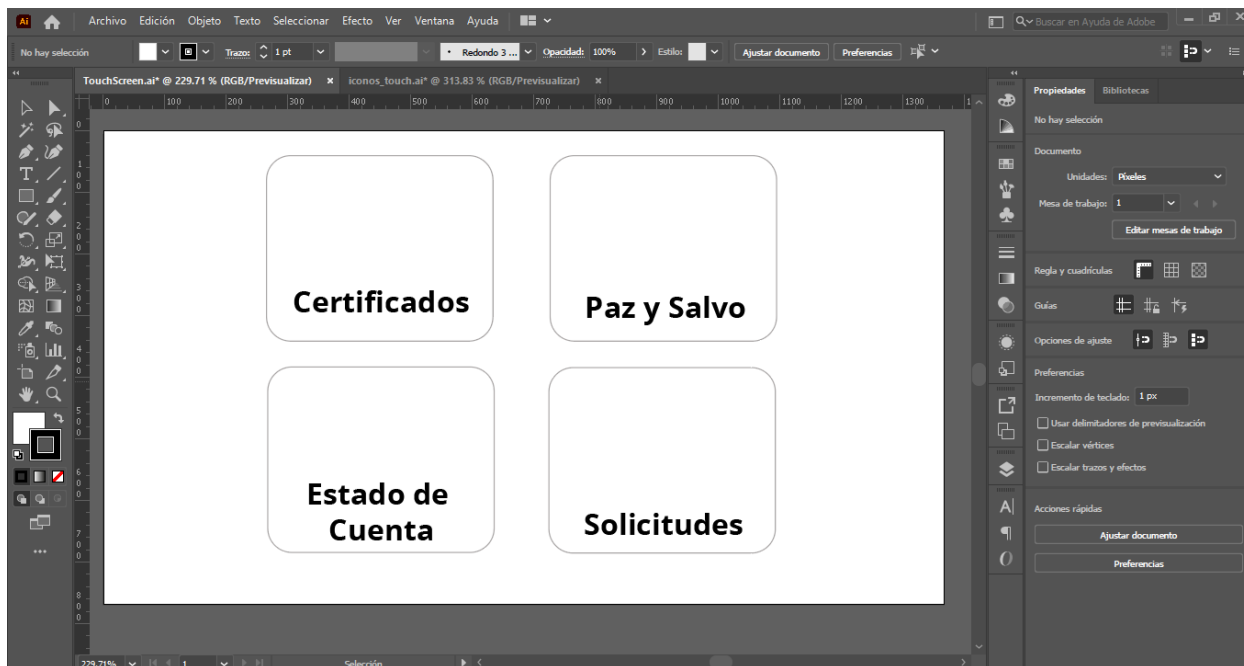
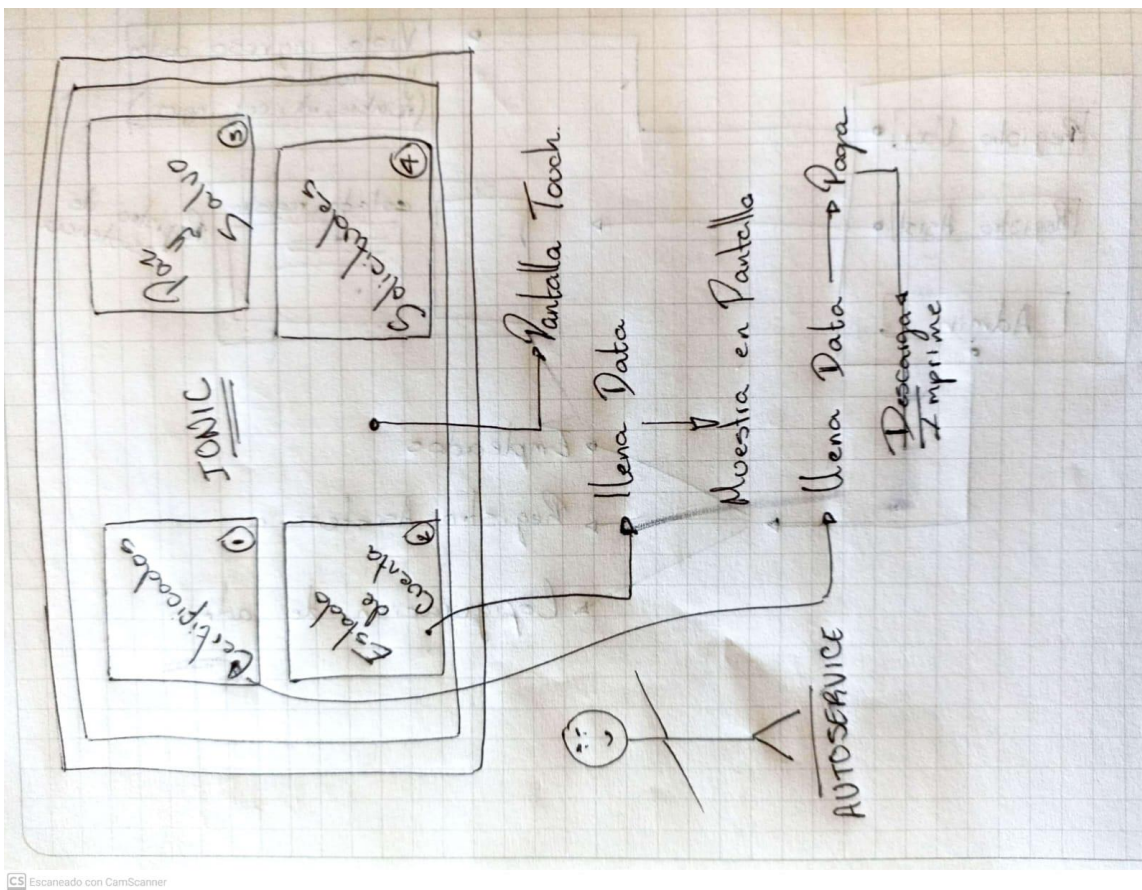
XIII Diagrama de Clases

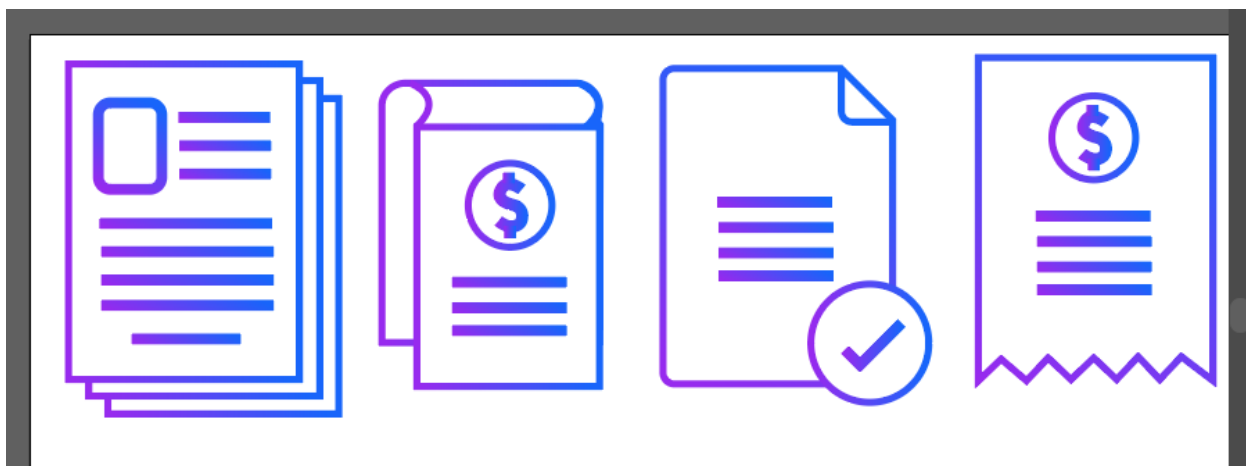
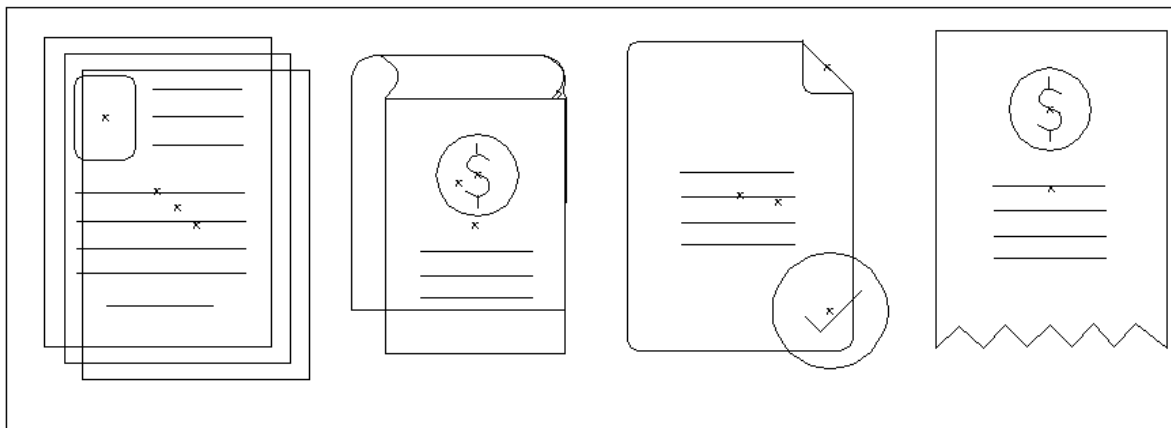


XIV Diagrama de Estado



XV Pre-Diseño





XVI Arquitectura de Software

16.1 Arquitectura de Frontend

16.1.1 Vista Principal



Certificados

Here's a small text description for the card content.
Nothing more, nothing less.



Paz y Salvos

Here's a small text description for the card content.
Nothing more, nothing less.



Estado de Cuenta

Here's a small text description for the card content.
Nothing more, nothing less.



Solicitudes

Here's a small text description for the card content.
Nothing more, nothing less.

16.1.2 Vista Certificados



Certificados

Invalid email

PAGAR

IMPRIMIR



16.1.3 Vista Paz y Salvos



Paz y Salvo

CONSULTAR

IMPRIMIR



16.1.4 Vista Solicitudes



Solicitudes y Requerimientos

A través de solicitudes **NO SE RECIBIRÁ** Derechos de Petición y/o Tutelas, se tendrán que enviar al correo de **notificacionesjudicialesudc@gmail.com**.
Por lo tanto si esta realizando este tipo de solicitudes, **NO SE TENDRAN EN CUENTA**.

Realiza Tu Solicitud aquí

¿Qué tipo de solicitud deseas realizar?

Nombres:

¿No sabes cómo llenar el Formulario? Mira el tutorial... ↓

Tutorial



16.1.5 Vista Estado de Cuenta



Estado de Cuenta

CONSULTAR

Tu Estado de Cuenta es...

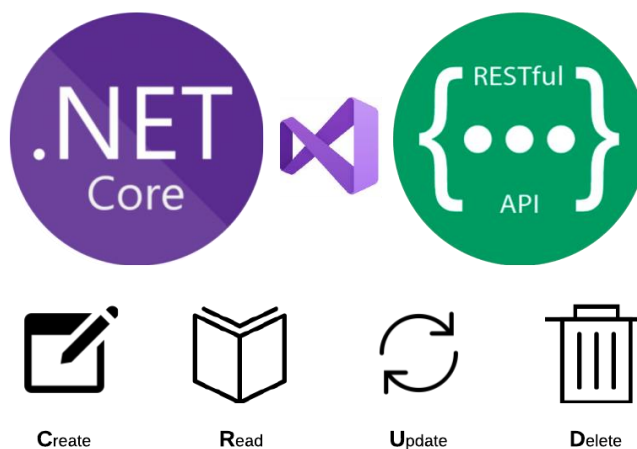
Total del Periodo: \$

Total Pagado: \$

Debo Pagar: \$

16.2 Arquitectura Backend

En este desarrollo se utiliza la arquitectura MVC (Modelo-Vista-Controlador) donde se separa el código según su utilidad, de la misma forma que a su vez para el controlador se utilizó una API en .NET desde la cual se pudiese interactuar sobre el modelo (base de datos) que se encuentra en SQL Server, ya sea para crear, modificar, actualizar o eliminar datos en el mismo.



Nombre	Fecha de modificación	Tipo	Tamaño
.vs	29/10/2022 3:50 p. m.	Carpeta de archivos	
bin	29/10/2022 3:50 p. m.	Carpeta de archivos	
Controllers	7/11/2022 8:41 p. m.	Carpeta de archivos	
Migrations	29/10/2022 3:50 p. m.	Carpeta de archivos	
Models	6/11/2022 4:15 p. m.	Carpeta de archivos	
obj	7/11/2022 6:46 p. m.	Carpeta de archivos	
Properties	29/10/2022 3:50 p. m.	Carpeta de archivos	
appsettings.Development.json	29/10/2022 3:50 p. m.	Archivo de origen ...	1 KB
appsettings.json	29/10/2022 6:19 p. m.	Archivo de origen ...	1 KB
Conexion_SQL.csproj	7/11/2022 6:47 p. m.	C# Project file	2 KB
Conexion_SQL.csproj.user	29/10/2022 3:50 p. m.	Per-User Project O...	1 KB
Conexion_SQL.sln	29/10/2022 3:50 p. m.	Visual Studio Solut...	2 KB
DBUDCContext.cs	29/10/2022 3:50 p. m.	C# Source File	111 KB
Program.cs	29/10/2022 3:50 p. m.	C# Source File	1 KB
Startup.cs	29/10/2022 3:50 p. m.	C# Source File	3 KB

Swagger Support by SMARTBEAR Select a definition Conexion_SQL v1

Conexion_SQL v1 OAS3

/swagger/v1/swagger.json

- Archivo >
- Autenticacion >
- Carnet >
- Connexion >
- ControlAcceso >
- Cuota >
- FacturaCredito >
- Facultad >

Archivo

POST /api/Archivo

Autenticacion

>

Carnet

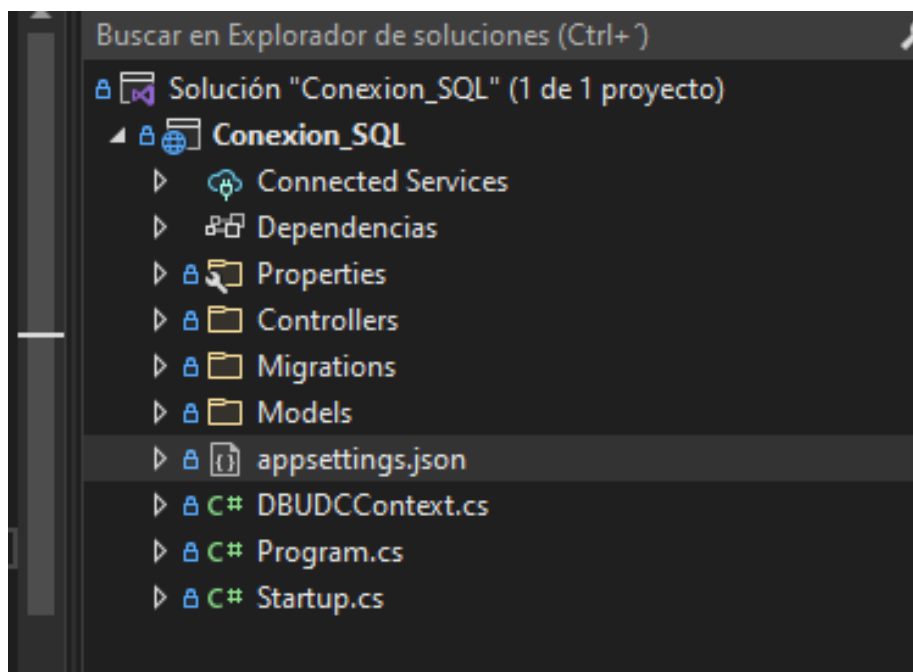
>

- GET /api/Carnet
- POST /api/Carnet
- PUT /api/Carnet
- GET /api/Carnet/{Email}
- DELETE /api/Carnet/{id}

Connexion

>

- PUT /api/Connexion/{id}
- DELETE /api/Connexion/{id}



XVII Librerías y Complementos

- **ComponentFixture**: Las propiedades y métodos de `ComponentFixture` permiten acceso al componente, su representación DOM y algunos aspectos del medio Angular.
- **TestBed**: Es la primera y más importante de las utilidades para poder hacer pruebas en Angular
- **waitForAsync**: Envuelve una función de prueba en una zona de prueba asíncrona. La prueba se completará automáticamente cuando se realicen todas las llamadas asíncronas dentro de esta zona. Se puede utilizar para envolver una inyección.
- **HttpClient**: se utiliza para hacer llamada de información desde la API (API REST)
- **Injectable**: Es un patrón de diseño en el que una clase requiere instancias de una o más clases y en vez de generarlas dentro de su propio constructor, las recibe ya instanciadas por un mecanismo externo.
- **Observable**: Es un objeto con un método `subscribe` que toma como parámetro un observador y nos retorna una suscripción.
- **pdfMake**: una biblioteca para la creación de pdfs desde la parte cliente de Angular.
- **AlertController**: Es el controlador encargado de dar acceso a toda la funcionalidad de los Alerts de manera programática

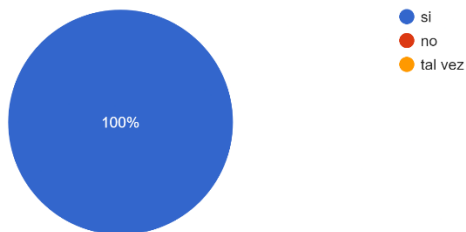
XVIII Resultados

El software se realiza para mejorar y facilitar los procesos dentro de la universidad y cumple con todos los requerimientos funcionales que se establecieron al inicio de su idealización como autoservice, por ende, con este último software se cumple con la satisfacción total del cliente y sus estudiantes, puesto que el sistema ha sido bien desarrollado, bien utilizado y además a ello, se ha establecido que el software ha automatizado procesos para que estos sean atendidos de formas más rápida por los empleados encargados y la autogestión estudiantil.

Por último, adjuntamos una estadística realizada a una pequeña muestra para comprobar los resultados de satisfacción obtenidos con el software:

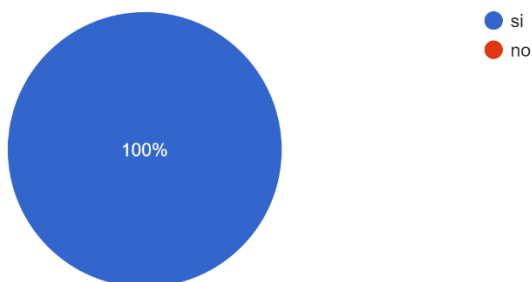
cree que touchscreen logra mejorar los procesos de certificados?

19 respuestas



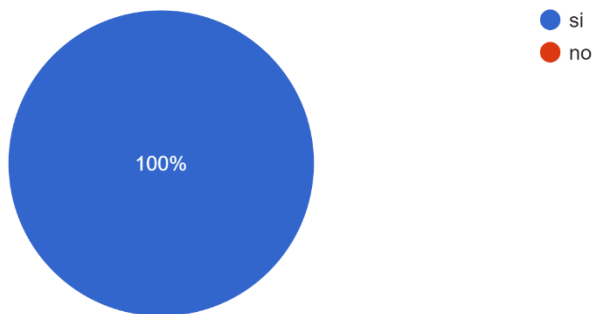
considera que el tiempo del proceso a mejorado considerablemente

19 respuestas



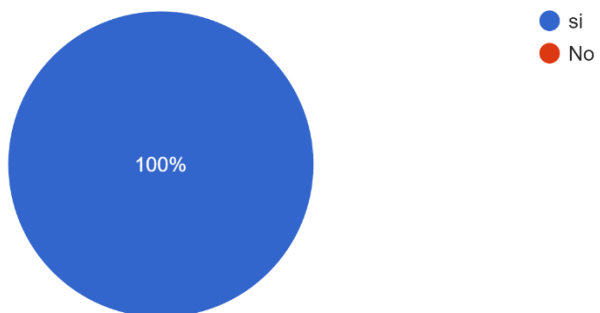
se le facilita usar una tableta para realizar sus solicitudes

19 respuestas



considera efectivos los tiempos de respuesta del touchscreen

19 respuestas



XIX Referencias

- *UniMovil:*

Recuperado de:

http://sedici.unlp.edu.ar/bitstream/handle/10915/23789/Documento_completo.pdf?sequence=1

- *UniNorte*

Recuperado de:

https://play.google.com/store/apps/details?id=co.edu.uninorte.mobileapp&hl=es_CO&gl=US

- *JaveMovil:*

Recuperado de:

https://play.google.com/store/apps/details?id=com.ionicframework.javemovil30721390&hl=es_CO&gl=US