

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

**APLICACIÓN DE SOFTWARE DE GESTIÓN ADMINISTRATIVA  
APLICACIÓN DE ESCRITORIO DE PROYECCIÓN DE CURSOS  
ACADÉMICOS PARA EL SUBDECANATO DE LA FIEE**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO INGENIERO EN  
TECNOLOGÍAS DE LA INFORMACIÓN**

**ANGEL JOEL SOLIS ANDINO**

**angel.solis@epn.edu.ec**

**DIRECTOR: PhD. ANA MARÍA ZAMBRANO VIZUETE**

**ana.zambrano@epn.edu.ec**

**DMQ, agosto 2023**

## **CERTIFICACIONES**

Yo, ANGEL JOEL SOLIS ANDINO, afirmo que soy el autor del trabajo de integración curricular descrito en este documento; este trabajo no ha sido presentado antes para ningún tipo de grado académico o calificación profesional; además, he revisado y utilizado las referencias bibliográficas que están incluidas en este informe.

**ANGEL JOEL SOLIS ANDINO**

Certifico que el presente trabajo de integración curricular fue desarrollado por ANGEL JOEL SOLIS ANDINO, bajo mi supervisión.

**Dra. ANA MARÍA ZAMBRANO VIZUETE**  
**DIRECTOR**

## **DECLARACIÓN DE AUTORÍA**

Mediante esta declaración, dejamos constancia de que el trabajo de integración curricular mencionado, junto con cualquier resultado generado a partir de él, se encuentra en dominio público y estará disponible para la comunidad a través del repositorio oficial de la Escuela Politécnica Nacional. No obstante, queremos aclarar que conservamos los derechos de propiedad intelectual como autores que contribuimos al desarrollo de este trabajo. Esto se ajusta a las regulaciones sobre propiedad intelectual, normativas internas y demás disposiciones pertinentes

Sr. ANGEL JOEL SOLIS ANDINO

Dra. ANA MARÍA ZAMBRANO VIZUETE

## DEDICATORIA

A mis padres Gladys y Angel, quienes siempre dedicaron su tiempo para alentarme a perseguir mis objetivos e inculcaron en mí sabiduría y perseverancia para cumplir mis metas, siendo un ejemplo perfecto de lucha en mi crecimiento como persona.

A mi hermana Mishell, quien ha depositado su confianza en mí para ser cómplice de sus aventuras.

A mi apreciada tía Anita, quien siempre albergó un sincero deseo de presenciar mi crecimiento como un profesional ejemplar. Valoro mucho su influencia y aliento.

A mis queridos amigos, Erika, Daniel y Kevin, quienes me han brindado su apoyo incondicional tanto en mi vida personal como académica, así como felicidad cuando más lo necesito. Juntos hemos compartido experiencias maravillosas creando recuerdos significativos que conservo con afecto, los llevaré siempre en mi corazón futuros colegas.

A mi familia en general, que siempre tuvo palabras adecuadas para aportar al desarrollo de mi vida universitaria, ofreciéndome consuelo en momentos de desahogo.

Quiero expresar mi gratitud a cada persona que, de una forma u otra, ha contribuido para que este logro sea una realidad, desde las grandes acciones hasta los pequeños gestos.

***Angel Joel Solis Andino***

## **AGRADECIMIENTO**

Agradezco mucho a la Dra. Ana María Zambrano, directora de este Trabajo de Integración Curricular, Coordinadora de carrera, y docente de la Escuela Politécnica Nacional, quien con su paciencia supo brindarme las guías adecuadas para poder desarrollar este trabajo, haciendo que terminara en buen recaudo.

Un sincero agradecimiento a todos quienes han creído en mí, a todos quienes han sido un ejemplo de superación, humildad y sacrificio; enseñándome a valorar todo lo que tengo y lo que soy.

Muchas gracias.

***Angel Joel Solis Andino***

# ÍNDICE DE CONTENIDOS

|   |      |
|---|------|
| CERTIFICACIONES.....  | I    |
| DECLARACIÓN DE AUTORÍA.....   | II   |
| DEDICATORIA.....  | III  |
| AGRADECIMIENTO.....   | IV   |
| ÍNDICE DE CONTENIDOS.....   | V    |
| RESUMEN.....  | VII  |
| ABSTRACT.....   | VIII |
| 1. INTRODUCCIÓN.....  | 1    |
| 1.1. OBJETIVO GENERAL.....  | 1    |
| 1.2. OBJETIVOS ESPECÍFICOS.....   | 1    |
| 1.3. ALCANCE.....   | 1    |
| 1.4. MARCO TEÓRICO.....   | 3    |
| 1.4.1. METODOLOGÍA KANBAN [1] [2].....  | 3    |
| 1.4.2. ARQUITECTURA EN CAPAS.....   | 4    |
| 1.4.3. TECNOLOGÍAS, ENTORNOS DE DESARROLLO Y LENGUAJES DE PROGRAMACIÓN EN LA CAPA DE DATOS.....       | 5    |
| 1.4.4. TECNOLOGÍAS, ENTORNOS DE DESARROLLO Y LENGUAJES DE PROGRAMACIÓN EN LA CAPA DE NEGOCIO.....     | 7    |
| 1.4.5. TECNOLOGÍAS, ENTORNOS DE DESARROLLO Y LENGUAJES DE PROGRAMACIÓN PARA CAPA DE PRESENTACIÓN..... | 9    |
| 2. METODOLOGÍA.....   | 11   |
| 2.1. DISEÑO.....  | 11   |
| 2.1.1. PLANTEAMIENTO DEL TABLERO KANBAN.....  | 11   |
| 2.1.2. REQUERIMIENTOS FUNCIONALES.....  | 12   |
| 2.1.3. REQUERIMIENTOS NO FUNCIONALES.....   | 15   |
| 2.1.4. DISEÑO DE LA CAPA DE DATOS.....  | 17   |
| 2.1.5. DISEÑO DE LA CAPA DE NEGOCIO.....  | 19   |
| 2.1.6. DISEÑO DE LA CAPA DE PRESENTACIÓN.....   | 25   |
| 2.2. IMPLEMENTACIÓN.....  | 28   |
| 2.2.1. ACTUALIZACIÓN DEL TABLERO KANBAN.....  | 28   |
| 2.2.2. CODIFICACIÓN DE LA BASE DE DATOS.....  | 29   |
| 2.2.3. CODIFICACIÓN DE LA CAPA DE NEGOCIO.....  | 35   |
| 2.2.4. CODIFICACIÓN DE LA CAPA DE PRESENTACIÓN.....   | 37   |
| 3. RESULTADOS Y DISCUSIÓN.....  | 40   |
| 3.1. ACTUALIZACIÓN DEL TABLERO KANBAN.....  | 40   |

|        |   |    |
|--------|---|----|
| 3.2.   | DEFINICIÓN DEL ENTORNO DE PRUEBAS.....                    | 41 |
| 3.3.   | VALIDACIÓN DE REQUERIMIENTOS .....                        | 42 |
| 3.3.1  | VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES .....            | 43 |
| 3.3.2  | VALIDACIÓN DE REQUERIMIENTOS NO FUNCIONALES .....         | 44 |
| 3.4.   | CIERRE DEL TABLERO KANBAN.....                            | 46 |
| 3.5.   | CONCLUSIONES Y RECOMENDACIONES.....                       | 47 |
| 3.5.1. | CONCLUSIONES .....  | 47 |
| 3.5.2. | RECOMENDACIONES.....                                      | 48 |
| 4.     | REFERENCIAS BIBLIOGRÁFICAS .....                          | 49 |
| 5.     | ANEXOS .....  | 51 |
| 5.1.   | ANEXO I: CÓDIGO DE LA APLICACIÓN.....                     | 51 |
| 5.2.   | ANEXO II: SCRIPT DE LA BASE DE DATOS .....                | 51 |
| 5.3.   | ANEXO III: MANUAL DE USUARIO .....                        | 51 |
| 5.4.   | ANEXO IV: PLANTILLA DE INGRESO DE ÍNDICES ACADÉMICOS..... | 51 |

## RESUMEN

El presente Trabajo de Integración Curricular trata sobre un prototipo de Software de Gestión Administrativa; este prototipo da como resultado la generación de las respectivas Proyecciones Académicas utilizados en los diferentes periodos académicos de la Escuela Politécnica Nacional, especialmente en la Facultad de Ingeniería Eléctrica y Electrónica. Mediante la implementación de este prototipo, se agilizará el trabajo manual del personal administrativo encargado, dando como resultado la simplificación de acciones en el proceso establecido, convirtiéndose en una tarea mucho más eficiente.

En el primer capítulo se establecen los fundamentos teóricos, detallando los aspectos importantes que se servirán como base para el desarrollo del prototipo. Estos aspectos incluyen un análisis y estudio de la metodología Kanban la cual permite llevar a cabo el proyecto de manera ordenada y mediante un cronograma establecido. Además, se aborda el estudio de la arquitectura en capas. Una vez abarcado dichos apartados, se procede a conocer las tecnologías pertinentes, y por último se explicará los diferentes entornos y lenguajes de programación utilizados para la implementación de la aplicación.

A lo largo del segundo capítulo, se exponen características distintivas de la creación de la aplicación de escritorio, las cuales se dividen en dos aspectos: Diseño e Implementación. En la sección de Diseño, se establecen los criterios fundamentales para el desarrollo del prototipo. Por otro lado, en la sección de Implementación, se consideran las diversas decisiones necesarias para la creación del aplicativo.

Por último, se detalla la definición y el entorno de pruebas que se han llevado a cabo en la aplicación, así como los resultados obtenidos tras la finalización de la fase de implementación. Se verifica el cumplimiento de los requerimientos funcionales y no funcionales previamente establecidos, lo que culmina en el cierre del desarrollo de la aplicación en cuestión.

**PALABRAS CLAVE:** Proyección Académica, Kanban, Requerimientos funcionales, Requerimientos no funcionales.



## **ABSTRACT**

This Curricular Integration Work deals with a prototype of Administrative Management Software; This prototype results in the generation of the respective Academic Projections used in the different academic periods of the National Polytechnic School, especially in the Faculty of Electrical and Electronic Engineering. Through the implementation of this prototype, the manual work of the administrative staff in charge will be streamlined, resulting in the simplification of actions in the established process, becoming a much more efficient task.

In the first chapter the theoretical foundations are established, detailing the important aspects that will be used as a basis for the development of the prototype. These aspects include an analysis and study of the Kanban methodology which allows carrying out the project in an orderly manner and through an established schedule. In addition, the study of layered architecture is addressed. Once these sections are covered, we proceed to learn about the relevant technologies, and finally, the different environments and programming languages used for the implementation of the application will be explained.

Throughout the second chapter, distinctive features of the creation of the desktop application are exposed, which are divided into two aspects: Design and Implementation. In the Design section, the fundamental criteria for the development of the prototype are established. On the other hand, in the Implementation section, the various decisions necessary for the creation of the application are considered.

Finally, the definition and the test environment that have been carried out in the application are detailed, as well as the results obtained after the completion of the implementation phase. Compliance with previously established functional and non-functional requirements is verified, which culminates in the closure of the development of the application in question.

**KEY WORDS:** Academic projection, Kanban, Functional requirements, Non-functional requirements.

# **1. INTRODUCCIÓN**

Actualmente el Subdecanato de la Facultad de Ingeniería Eléctrica y Electrónica (FIEE) de la Escuela Politécnica Nacional requiere de la información sobre la Proyección de Cursos Académicos; esta información permite gestionar la numeración de estudiantes y la creación de cursos para el próximo periodo estudiantil. Sin embargo, esta información se la obtiene de forma manual en un archivo xlsx (Excel), ocasionando que se tenga un gran porcentaje de incertidumbre en los resultados finales de los datos; y por tanto, obtener factores poco eficientes como pérdida de tiempo útil, toma de decisiones no acertadas, y/o fallos manuales involuntarios. Por tal razón el presente Trabajo de Integración Curricular conlleva a desarrollar una aplicación la cual facilite la realización de la actividad de forma automatizada, y así evitar los factores antes mencionados.

## **1.1. OBJETIVO GENERAL**

Desarrollar una aplicación de escritorio de tal manera que se obtenga la proyección de cursos académicos para el Subdecanato de la FIEE.

## **1.2. OBJETIVOS ESPECÍFICOS**

1. Analizar herramientas y tecnologías para el desarrollo de la aplicación.
2. Diseñar una metodología que establezca los datos esperados en la proyección de cursos académicos.
3. Implementar la aplicación de escritorio en base al diseño planteado.
4. Realizar un proceso de pruebas comparando los resultados obtenidos con los esperados.

## **1.3. ALCANCE**

El componente práctico que se realizará como parte del Trabajo de Integración Curricular se llevará a cabo mediante varios parámetros que se los ha obtenido a partir de los objetivos específicos, lo cual va a formar parte de una serie de fases que se describen a continuación.

### **A. Fase de Análisis**

En esta fase se procederá con el estudio de recursos necesarios para llevar a cabo el Trabajo de Integración Curricular, el cual empieza con el manejo adecuado de la arquitectura en capas; se incluirá el estudio de los lenguajes de programación como C# y Visual Studio, junto

con un sistema gestor de base de datos como SQL Server, ya que han sido escogidos como los que mejor se acoplan a esta aplicación.

Adicionalmente se estudiará la metodología Kanban que ayudará a tener una correcta planificación de las tareas que se mantendrán el adecuado orden en la realización del proyecto.

## **B. Fase de Diseño**

En el comienzo de esta fase se hará la implementación del tablero Kanban, adecuando las tareas y actividades necesarias con los tiempos requeridos para la resolución de estas; por otra parte, se establece los requerimientos funcionales y no funcionales que va a requerir el sistema. Se realizarán bosquejos en donde se puedan apreciar los módulos requeridos del proyecto; y se definirán las capas que distribuirán diferentes roles en la aplicación con la finalidad de que cada capa cumpla con su responsabilidad. También se procederá con la realización de diagramas que serán de utilidad para documentar y planificar mejor las tareas, de preferencia se tomará en cuenta los siguientes diagramas:

- **Diagrama Relacional:** Representa los datos mediante tablas para administrarlos dinámicamente.
- **Diagrama de Clases:** Trazan la estructura de un sistema tomando en cuenta las clases, atributos, operaciones y objetos.
- **Diagrama de Actividades:** Indica el proceso de ejecución de las acciones en el sistema.

## **C. Fase de Implementación**

Cada actividad que se va realizando deberá ser actualizada en el tablero Kanban hasta llegar a la fecha en la que la aplicación ya debe estar totalmente terminada.

Mediante el uso de Lenguaje C# a través del IDE de Visual Studio, se procede a codificar las capas y clases necesarias establecidas en la fase de diseño.

Mediante el uso del gestor de base de datos de SQL Server se realizará la base de datos acorde al diagrama relacional.

Para implementar los bosquejos en la capa de presentación, se utilizará la herramienta de Windows Forms.

Al finalizar todas estas actividades se tendrá integrado el proyecto que dará paso a la realización de una fase de pruebas.

## **D. Fase de Análisis de Pruebas**

Una vez actualizada la fase de implementación en el tablero Kanban, se establecerá un escenario simulado verificando el cumplimiento de los requerimientos funcionales y no funcionales, validando que el prototipo final cumpla con el correcto funcionamiento. Si es necesario se corregirá los problemas encontrados, caso contrario se procede a redactar conclusiones, recomendaciones, y se dará por terminado el proyecto.

Estas actividades también deberán ser implementadas y actualizadas en el tablero Kanban.

## **1.4. MARCO TEÓRICO**

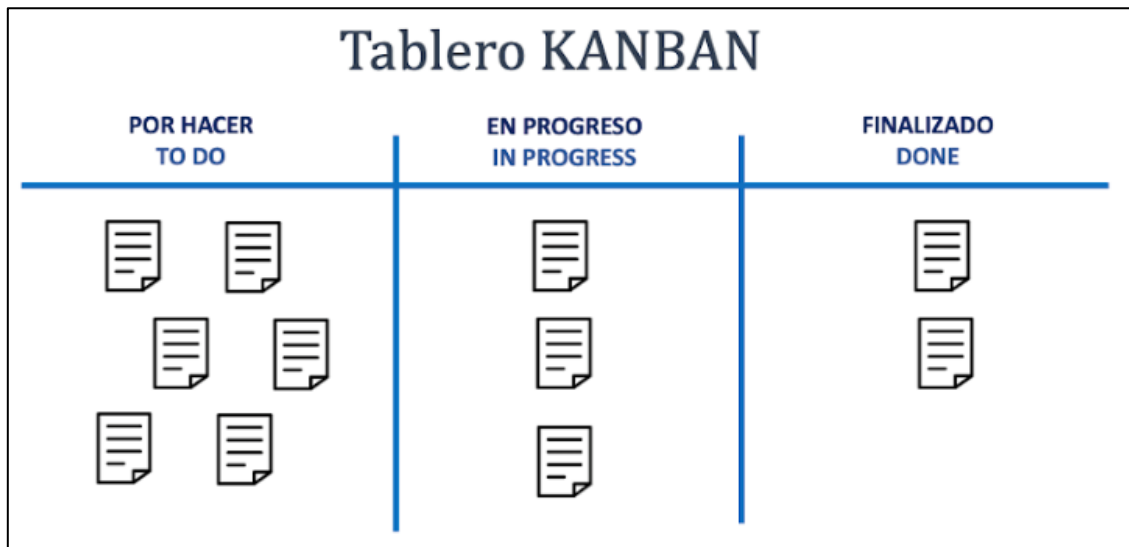
En el **Apartado 1.4.1** se abordará el estudio de la metodología Kanban, mientras que el **Apartado 1.4.2** corresponde al estudio de la arquitectura en capas, una vez abarcado dichos apartados, se procede a conocer las tecnologías, entornos y lenguajes de programación usados para la implementación de la aplicación.

### **1.4.1. METODOLOGÍA KANBAN [1] [2]**

La metodología Kanban es un sistema compuesto de principios y prácticas, combinadas de varias herramientas que permiten la visualización de las tareas que se realizan en un trabajo o proyecto; se basa en gestionar flujos de trabajo en donde se define, y se mejora los avances que proporciona el trabajo con la finalidad de maximizar la eficiencia y mejorar continuamente.

El origen de esta metodología viene del ingeniero Taiichi Ono, la cual lo diseñó como un sistema de planificación para la empresa automovilística Toyota que realizaba sus procedimientos en base a la técnica JIT (Just in Time) que se refiere a realizar solo lo que es necesario y justo a tiempo [3]. Debido a los avances en la tecnología, esta metodología tuvo éxito y crecimiento, y por tal razón, diferentes sectores comerciales como el desarrollo de software y la informática han optado por adoptar esta metodología en la ejecución de sus proyectos. Cuenta con varias ventajas, una de ellas es que la metodología Kanban es transparente, es decir que indica las respectivas tareas asignadas y cada persona del equipo tiene claro lo que debe realizar. Otra ventaja importante es que evita tareas ineficientes, de tal manera que se elimina la sobreproducción y limitación de recursos. También permite controlar las tareas haciendo que el tiempo de producción se reduzca y que la planificación mejore; por otro lado, es totalmente flexible ya que permite realizar un buen seguimiento de las tareas asignadas y realizadas con el fin de poder llevar un buen seguimiento de las mismas.

La metodología Kanban es flexible y se puede adaptar a diferentes tipos de equipos y procesos de trabajo siendo su enfoque principal el mejorar el flujo de trabajo y aumentar la colaboración y la transparencia dentro del equipo.



**Figura 1.1.** Esquema del Tablero Kanban.

#### **1.4.2. ARQUITECTURA EN CAPAS**

La arquitectura de 3 capas es una estructura de software de aplicación, la cual separa una aplicación en tres niveles los cuales componen la Capa de Datos, Capa Lógica y la Capa de Presentación, teniendo así la ventaja de que cada nivel se ejecute en su propia infraestructura y de esta manera permita el escalamiento sin afectar a las demás capas [4].

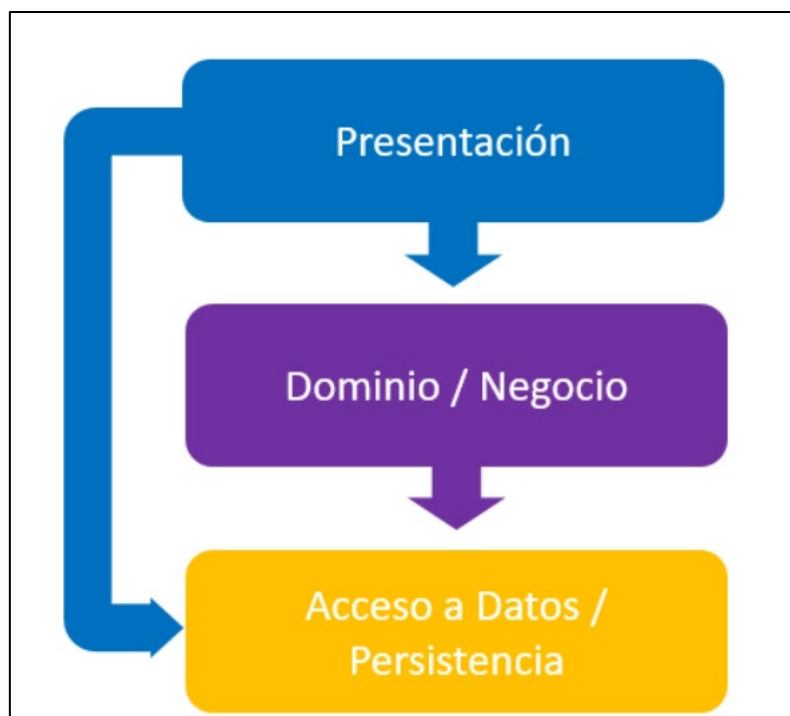
La Capa de Acceso a Datos, también conocido como DAL por su acrónimo en inglés (Data Acces Layer), es una capa perteneciente a la arquitectura de una aplicación informática la cual permite el acceso a los datos mediante solicitudes realizadas a un gestor de base de datos.

La DAL se comunica con la Capa de Negocio, la cual asigna métodos adecuados para recuperar datos de manera abstracta debido a que se vinculan objetos de la base de datos relacionales, con objetos de lenguajes de programación.

La Capa de Negocio realiza funciones principales en la aplicación, como por ejemplo la recopilación y procesamiento de la información generada por la Capa de Presentación, la implementación de funciones, y añadir, eliminar o editar datos en la Capa de Datos.

La Capa de Presentación o Interfaz de Usuario tiene como función especial la interacción con el usuario final; de esta manera se puede establecer parámetros para que el usuario maniobre la aplicación desarrollada con el fin de comunicarse con el lenguaje de máquina de manera sencilla. El objetivo de esta capa es mostrar la información al usuario, y también recopilar datos de este.

Al utilizar una arquitectura en capas en un sistema, se puede lograr una mayor cohesión y una baja dependencia entre los componentes, por lo cual facilita el mantenimiento, la flexibilidad y la evolución del sistema a largo plazo.



**Figura 1.2.** Representación de la arquitectura en capas.

### **1.4.3. TECNOLOGÍAS, ENTORNOS DE DESARROLLO Y LENGUAJES DE PROGRAMACIÓN EN LA CAPA DE DATOS.**

#### **1.4.3.1. Lenguaje SQL**

El Lenguaje de Consulta Estructurado o Structured Query Language (SQL), es un lenguaje de programación utilizado para procesar y almacenar información de una base de datos [5]. Con el uso de este lenguaje se puede realizar varias operaciones como consultas, inserciones, actualizaciones y eliminaciones de datos.

Este lenguaje brinda comandos válidos en los sistemas administrables de bases de datos, consta con varios elementos y componentes como identificadores, variables y condiciones las cuales permiten crear instrucciones de manera correcta [6].

Algunos de los comandos más importantes de SQL son los que se muestran en la Tabla 1.1:

**Tabla 1.1** Principales comandos del Lenguaje SQL.

| <b>COMANDO</b>      | <b>DESCRIPCIÓN</b>   |
|---------------------|--|
| <b>SELECT</b>       | Este comando permite obtener consultas desde una base de datos, a partir de esta función se pueden especificar condiciones como ordenar resultados, y limitar número de registros. |
| <b>INSERT</b>       | Permite agregar nuevos registros en una tabla, esto se lo puede hacer de una tabla a otra o especificando los valores para cada columna de la tabla.                               |
| <b>UPDATE</b>       | Facilita la modificación de datos existentes en una tabla mediante la especificación de una columna y proporcionando un nuevo valor.   |
| <b>DELETE</b>       | Elimina los registros de una tabla existente en una base de datos.   |
| <b>CREATE TABLE</b> | Crea las tablas donde serán almacenados los registros de la base de datos mediante la especificación de columnas, tipos de datos, restricciones, claves primarias y foráneas.      |
| <b>ALTER TABLE</b>  | Realiza cambios en la estructura de una tabla existente, permite agregar o eliminar columnas, cambiar los tipos de datos y modificar restricciones.                                |
| <b>DROP TABLE</b>   | Elimina las tablas creadas y todos los registros almacenados en dicha tabla.   |

#### 1.4.3.2. Microsoft SQL Server

Microsoft SQL Server es un sistema gestor de bases de datos relacional (RDBMS) el cual brinda servicio a aplicaciones de software [7]. Se basa en un lenguaje Transact-SQL lo que permite aplicar comandos necesarios para construir estructuras de bases de datos, implementar protecciones y manipular datos, crear procedimientos almacenados, realizar copias de seguridad, y administrar el flujo de los programas.

Un RDBMS es muy utilizado por organizaciones que administran grandes cantidades de datos, lo cual resulta beneficioso ya que permite representar y tener acceso fácil a datos relacionados [8].

Entre algunas características importantes de SQL Server se presentan en la Tabla 1.2:

**Tabla 1.2** Características de SQL Server.

| CARACTERÍSTICA                       | DESCRIPCIÓN  |
|--------------------------------------|--|
| <b>Escalabilidad</b>                 | SQL Server es altamente escalable y puede manejar una gran cantidad de volúmenes de datos de manera efectiva.  |
| <b>Seguridad</b>                     | Proporciona herramientas que garantizan la seguridad de datos, como por ejemplo la autenticación de usuarios, control de acceso, y auditorías de bases de datos. |
| <b>Inteligencia Empresarial (BI)</b> | Facilita herramientas de inteligencia empresarial destinadas a la administración y análisis de datos las cuales son apoyadas con <i>machine learning</i> .       |
| <b>Integración con Visual Studio</b> | Cuenta con una integración a Visual Studio, lo que permite a programadores crear y depurar aplicaciones compuestas con bases de datos.                           |

#### 1.4.4. TECNOLOGÍAS, ENTORNOS DE DESARROLLO Y LENGUAJES DE PROGRAMACIÓN EN LA CAPA DE NEGOCIO.

##### 1.4.4.1. Microsoft Visual Studio [9]

Microsoft Visual Studio es un Entorno de Desarrollo Integrado (IDE), es decir es un programa que proporciona diversas herramientas necesarias para el desarrollo de software; con la ayuda de este programa se puede realizar diversas acciones como editar, depurar y compilar código con el fin de crear sitios y aplicaciones tanto web como de escritorio, aplicaciones móviles, entre otras.

Este IDE es compatible con varios lenguajes de programación como C#, C++, Visual Basic, Python, PHP, entre otras, y brinda asistencia necesaria sin importar el tipo de lenguaje que se esté utilizando.

Algunas de las principales características se presentan en la Tabla 1.3:



**Tabla 1.3** Características de Microsoft Visual Studio.

| <b>CARACTERÍSTICA</b>               | <b>DESCRIPCIÓN</b>   |
|-------------------------------------|--|
| <b>Interfaz de usuario amigable</b> | Visual Studio cuenta con una interfaz de usuario muy intuitiva y fácil de utilizar de modo que permite a los desarrolladores trabajar de forma más rápida y eficiente. |
| <b>Integración con Git</b>          | Tiene una gran integración con Git, lo que permite controlar las versiones y el flujo de trabajo del equipo.   |
| <b>Soporte en la nube</b>           | Está integrado con varios servicios en la nube de Microsoft, lo que permite trabajar con servicios en la nube, y construir aplicaciones de nube escalares.             |
| <b>Depuración avanzada</b>          | Ofrece herramientas de depuración avanzadas, lo que permite a desarrolladores encontrar y solucionar problemas de código.  |

#### **1.4.4.2. Lenguaje C#**

C# es un lenguaje de Programación Orientado a Objetos desarrollado por la empresa Microsoft, el cual permite crear varios tipos de aplicaciones robustas que se ejecutan en el framework de .NET [10].

Este lenguaje es una evolución de los lenguajes C y C++, sin embargo, se diferencia uno del otro debido a que fueron creados con diferentes propósitos, por ejemplo, C# está especialmente diseñado para el desarrollo de programas y aplicaciones gráficas con un alto nivel de abstracción [11], por otra parte, C++ es diseñado como un propósito general, lo que significa que está destinado a ser utilizado en una amplia variedad de aplicaciones y escenarios de desarrollo como por ejemplo: sistemas embebidos de bajo nivel hasta aplicaciones de software de gran escala [11].

El objetivo principal de este lenguaje es proporcionar un entorno de desarrollo sencillo y eficiente de interfaces de usuario, además los desarrolladores pueden enfocarse en la lógica y funcionalidad de las aplicaciones, aprovechando el manejo de los aspectos gráficos y la interacción con el usuario [12].

Algunas de las ventajas y características del lenguaje C# se las puede apreciar en la Tabla 1.4:

**Tabla 1.4** Características y ventajas de C#.

| <b>CARACTERÍSTICA</b>        | <b>DESCRIPCIÓN</b>  |
|------------------------------|---|
| <b>Orientación a objetos</b> | Se basa en la programación con objetos y conceptos como encapsulación, herencia y polimorfismo. Permite la creación de clases, interfaces y abstracción de datos. |
| <b>Seguridad</b>             | Ofrece seguridad de tipos, es decir, que incluye un mecanismo para garantizar que los accesos a los tipos de datos se realicen de manera precisa y segura.        |
| <b>Multiplataforma</b>       | Originalmente creado para plataformas Windows, ahora se usa en otros sistemas operativos como MacOS, Linux, Android y iOS.  |

#### **1.4.5. TECNOLOGÍAS, ENTORNOS DE DESARROLLO Y LENGUAJES DE PROGRAMACIÓN PARA CAPA DE PRESENTACIÓN.**

##### **1.4.5.1. Windows Forms**

Windows Forms es una tecnología de desarrollo de aplicaciones de interfaz gráfica de usuario (GUI) para el sistema operativo Windows [13]. Esta tecnología ofrece una de las formas más productivas para desarrollar aplicaciones de escritorio gracias al diseñador visual integrado en Visual Studio.

Contiene un conjunto de bibliotecas administradas que simplifican las tareas comunes en el desarrollo de aplicaciones, como la manipulación de archivos y la interacción con el usuario [14]. Mediante el entorno de desarrollo de Visual Studio se puede crear aplicaciones que permitan la proyección de la información de manera intuitiva, solicitar datos a usuarios y comunicación con servidores mediante red.

Windows Forms ofrece una amplia gama de controles que se pueden añadir a los formularios, incluyendo controles para mostrar cuadros de texto, botones, cuadros desplegables, botones de opción e incluso integrar páginas web. Si alguno de los controles predefinidos no satisface las necesidades requeridas, Windows Forms también permite la creación de controles personalizados que se ajustan a las necesidades del proyecto [14]. También persigue un patrón de diseño basado en eventos, lo que significa que gran parte de la lógica de la aplicación se implementa en métodos de manejo de eventos. Esto permite una separación clara entre la interfaz de usuario y la lógica de negocio de la aplicación.

Algunas de las características más mencionadas de Windows Forms se encuentran en la Tabla 1.5:

**Tabla 1.5** Características de Windows Forms.

| <b>CARACTERÍSTICA</b>          | <b>DESCRIPCIÓN</b>  |
|--------------------------------|---|
| <b>Diseño visual</b>           | Permite diseñar la interfaz de usuario de manera visual mediante un editor de formularios facilitando la personalización y apariencia de la aplicación.   |
| <b>Controles y componentes</b> | Contiene una amplia gama de controles como botones, cajas de texto, listas desplegables, cuadros de dialogo, entre otros, que se pueden configurar mediante propiedades y eventos para adaptarse a necesidades. |
| <b>Manejo de eventos</b>       | Cada control genera una acción como respuesta a alguna interacción del usuario como hacer clic un botón o cambiar un valor.   |

## 2. METODOLOGÍA

En este capítulo se presentarán algunas características distintivas de la creación de la aplicación de escritorio sobre la Proyección de Cursos Académicos para el Subdecanato de la FIEE, consideraciones de diseño, y algunos de las particularidades implementados en el proceso de desarrollo.

En el **Apartado 2.1 DISEÑO**, se establecerán los criterios fundamentales para el desarrollo del prototipo, mientras que en el **Apartado 2.2 IMPLEMENTACIÓN**, se analizarán las diversas decisiones de desarrollo implementadas.

### 2.1. DISEÑO

En esta sección se mostrarán las directrices seguidas para el desarrollo de la aplicación de escritorio, así como los requerimientos funcionales y no funcionales del prototipo, y los diversos diseños de la arquitectura en capas. Por último, se presentará la propuesta de Diseño para la aplicación de Proyección de Cursos Académicos.

#### 2.1.1. PLANTEAMIENTO DEL TABLERO KANBAN.

Para establecer el tablero de Kanban, se utiliza una herramienta en línea denominada *Monday Work Management* [15], en la cual se puede especificar las distintas actividades a realizar y de igual manera actualizar los resultados a medida que se cumplan los objetivos y tareas establecidas.

El tablero Kanban está dividido en tres etapas (detenido, en curso, y listo), siguiendo el cronograma, las actividades que se encuentran ya realizadas para esta etapa del proyecto consisten las siguientes: investigación de la metodología Kanban y de la arquitectura en capas, consulta de parámetros importantes de las plataformas de visual Studio y Microsoft SQL Server, y por último, la comprensión del proceso para obtener resultados de la proyección académica.

Como se observa en la Figura 2.1, para la fase de Diseño se deben llevar a cabo 6 actividades, las cuales pertenecen a la etapa “En curso” de tablero Kanban, una vez culminadas dichas actividades se podrá seguir a la fase de implementación de la aplicación de escritorio en donde dichas actividades se encontrarán asignadas en la etapa “Detenido”, del tablero.

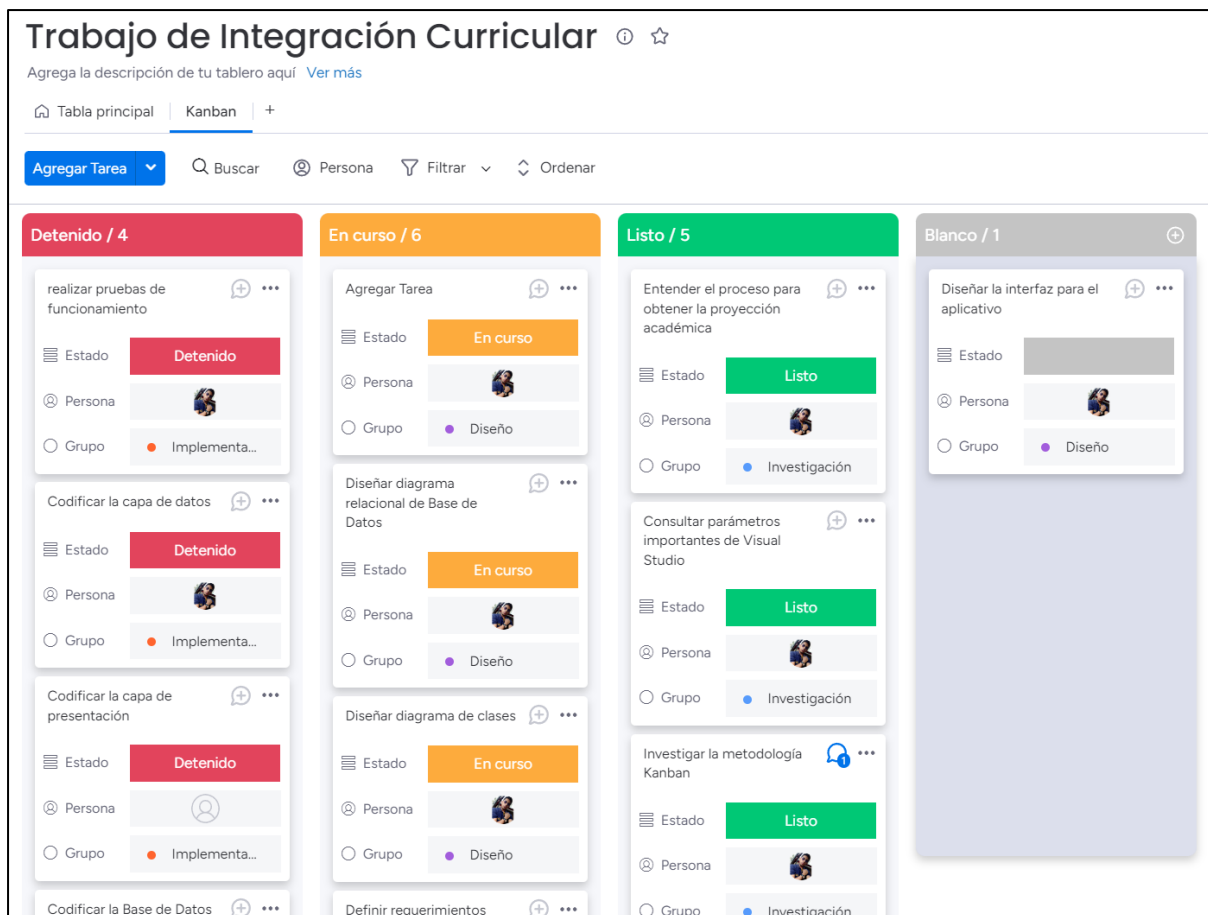


Figura 2.1. Esquema del Tablero Kanban en Etapa de Diseño.

### 2.1.2. REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales son enunciados que describen los servicios que el sistema debe proporcionar, cómo debe responder a entradas específicas y cómo debe comportarse en situaciones particulares. Estas características y funcionalidades específicas son necesarias para cumplir con las necesidades y expectativas de los usuarios. Estos requerimientos definen las acciones que el sistema debe realizar y los resultados que se esperan obtener [16].

Para la obtención de los requerimientos funcionales se ha tomado en cuenta el actor de interés denominado **“Analista”**, quien será la persona encargada de manipular la aplicación de escritorio, por tal razón los servicios que debe presentar la aplicación se definen en las siguientes Historias de Usuario.

**Tabla 2.1** Requerimiento Funcional de CRUD de datos.

| <b>Historia de Usuario #1</b>   |               |                   |      |
|---|---------------|-------------------|------|
| <b>Nombre:</b>  | CRUD de Datos |                   |      |
| <b>Código:</b>  | RF01          | <b>Prioridad:</b> | Alta |
| <b>Usuario:</b>   | Analista      |                   |      |
| <b>Descripción</b>  |               |                   |      |
| Se requiere que la aplicación cuente con una interfaz la cual permita crear, leer, actualizar, y eliminar registros, ya sea materias, carreras o índices académicos.  |               |                   |      |
| <b>Criterios de Aceptación</b>  |               |                   |      |
| <ul style="list-style-type: none"> <li>• El prototipo contará con un módulo denominado configuración de datos, el cual permite realizar el respectivo CRUD de los registros.</li> <li>• Mediante la aplicación se podrá realizar CRUD de carreras, materias e índices académicos.</li> <li>• Se podrá validar este requerimiento al observar que los cambios se mantienen al realizar una nueva consulta.</li> <li>• La interfaz contendrá los filtros necesarios para facilitar la búsqueda de los datos que se deseen manipular.</li> </ul> |               |                   |      |

**Tabla 2.2** Requerimiento Funcional de servicio de reportería.

| <b>Historia de Usuario #2</b>   |                        |                   |      |
|---|------------------------|-------------------|------|
| <b>Nombre:</b>  | Servicio de reportería |                   |      |
| <b>Código:</b>  | RF02                   | <b>Prioridad:</b> | Alta |
| <b>Usuario:</b>   | Analista               |                   |      |
| <b>Descripción</b>  |                        |                   |      |
| Se requiere que la aplicación produzca documentos, los cuales permitan observar las proyecciones calculadas a partir de los índices académicos en diferentes formatos.  |                        |                   |      |
| <b>Criterios de Aceptación</b>  |                        |                   |      |
| <ul style="list-style-type: none"> <li>• El prototipo contará con un módulo denominado Reportería, el cual indicará la proyección de los índices seleccionados en un documento PDF.</li> <li>• El mismo módulo permitirá que el archivo PDF se pueda exportar e imprimir.</li> <li>• También la aplicación permitirá exportar las proyecciones como un archivo xlsx.</li> <li>• Los documentos creados no solo arrojaran datos, este constará de un formato el cual resulte amigable con el usuario.</li> </ul> |                        |                   |      |

**Tabla 2.3** Requerimiento Funcional del cálculo de proyecciones.

| <b>Historia de Usuario #3</b>  |                         |                   |      |
|--|-------------------------|-------------------|------|
| <b>Nombre:</b>   | Cálculo de proyecciones |                   |      |
| <b>Código:</b>   | RF03                    | <b>Prioridad:</b> | Alta |
| <b>Usuario:</b>  | Analista                |                   |      |
| <b>Descripción</b>   |                         |                   |      |
| Se requiere que la aplicación realice el cálculo de las proyecciones para el siguiente periodo académico.  |                         |                   |      |
| <b>Criterios de Aceptación</b>   |                         |                   |      |
| <ul style="list-style-type: none"> <li>• El prototipo contará con un módulo denominado Generar Proyecciones, el cual permitirá realizar el cálculo de las proyecciones de acuerdo a los índices almacenados en la base de datos.</li> <li>• El cálculo de la proyección indicará el número de estudiantes nuevos, repetidos y totales, además indicará el número de estudiantes que deben ir en cada curso.</li> <li>• A partir de este punto también se puede obtener el documento de reportería, con la facilidad de mejorar la visualización de los resultados.</li> <li>• Una vez concebido el documento se podrá verificar los resultados.</li> </ul> |                         |                   |      |

**Tabla 2.4** Requerimiento Funcional de ingreso de índice de error.

| <b>Historia de Usuario #4</b>   |                            |                   |      |
|---|----------------------------|-------------------|------|
| <b>Nombre:</b>  | Ingreso de índice de error |                   |      |
| <b>Código:</b>  | RF04                       | <b>Prioridad:</b> | Alta |
| <b>Usuario:</b>   | Analista                   |                   |      |
| <b>Descripción</b>  |                            |                   |      |
| Se requiere que la aplicación permita ingresar un índice de error en las estimaciones calculadas.   |                            |                   |      |
| <b>Criterios de Aceptación</b>  |                            |                   |      |
| <ul style="list-style-type: none"> <li>• El prototipo contará con una pantalla flotante el cual indicará la estimación calculada de la materia seleccionada, para poder ingresar los índices de error.</li> <li>• Estos índices serán establecidos por el criterio del usuario en base a resultados anteriores.</li> <li>• Al momento de guardar el índice de error, el sistema realizará el cambio necesario que modificará la estimación de la proyección.</li> </ul> |                            |                   |      |

- Se podrá observar el índice que ha sido asignado a cada materia, para tomar en cuenta en los resultados de las proyecciones.
- El ingreso del índice se lo ingresará antes de realizar la proyección académica.
- Con el índice de error se tendrá mejores resultados en la proyección.

### 2.1.3. REQUERIMIENTOS NO FUNCIONALES

Se denomina a los requerimientos no funcionales como las características y criterios que definen la calidad, eficiencia, y limitaciones de un sistema. Por lo general se centran en aspectos como la seguridad, usabilidad, rendimiento y escalabilidad que el sistema presente y no tienen ninguna relación con la funcionalidad del sistema. También se consideran aspectos como disponibilidad, interoperabilidad, capacidad de mantenimiento y adaptabilidad. Estos requerimientos aseguran que el sistema cumpla con las expectativas de los usuarios y pueda evolucionar en un entorno dinámico [17].

Mediante estos requerimientos se busca asegurar la conformidad del usuario a través de un software proporcionado de calidad. Algunos de los requerimientos no funcionales utilizados para el desarrollo de esta aplicación se detallan a continuación:

**Tabla 2.6** Requerimiento no Funcional de usabilidad de la aplicación.

| <b>Historia de Usuario #6</b>  |                             |                   |      |
|--|-----------------------------|-------------------|------|
| <b>Nombre:</b>   | Usabilidad de la aplicación |                   |      |
| <b>Código:</b>   | RNF01                       | <b>Prioridad:</b> | Alta |
| <b>Usuario:</b>  | Analista                    |                   |      |
| <b>Descripción</b>   |                             |                   |      |
| Es necesario contar con una interfaz gráfica que sea amigable, de fácil uso e intuitiva, de manera que el usuario pueda emplearla sin enfrentar obstáculos significativos.   |                             |                   |      |
| <b>Criterios de Aceptación</b>   |                             |                   |      |
| <ul style="list-style-type: none"> <li>• El prototipo contará con mensajes de ayuda que permitirá manejar adecuadamente la aplicación.</li> <li>• Las interfaces de las pantallas de la aplicación serán adaptadas a un diseño “<i>Responsive</i>”, para que los usuarios accedan de manera efectiva sin importar el tipo de resolución de su dispositivo.</li> <li>• Al usuario no le debe ser difícil manipular las distintas opciones del sistema.</li> </ul> |                             |                   |      |



**Tabla 2.7** Requerimiento no Funcional de conexión a base de datos.

| <b>Historia de Usuario #7</b>  |                          |                   |      |
|--|--------------------------|-------------------|------|
| <b>Nombre:</b>   | Conexión a Base de Datos |                   |      |
| <b>Código:</b>   | RNF02                    | <b>Prioridad:</b> | Alta |
| <b>Usuario:</b>  | Analista                 |                   |      |
| <b>Descripción</b>   |                          |                   |      |
| El sistema debe ser capaz de tener una conexión con una base de datos externa, esta conexión debe incluir el acceso a los datos almacenados y por ende también la consulta, inserción, actualización y eliminación de registros.   |                          |                   |      |
| <b>Criterios de Aceptación</b>   |                          |                   |      |
| <ul style="list-style-type: none"> <li>• El sistema debe ser capaz de establecer una conexión con la base de datos.</li> <li>• Al momento de realizar las consultas, los resultados deben ser puntuales de acuerdo a los datos almacenados, de esta forma se validará que existe una conexión con una base de datos.</li> <li>• El sistema debe permitir el CRUD de datos</li> <li>• La base de datos será implementada a través del sistema de SQL Server.</li> </ul> |                          |                   |      |

**Tabla 2.8** Requerimiento no Funcional de rendimiento de la aplicación.

| <b>Historia de Usuario #8</b>   |                              |                   |      |
|---|------------------------------|-------------------|------|
| <b>Nombre:</b>  | Rendimiento de la aplicación |                   |      |
| <b>Código:</b>  | RNF03                        | <b>Prioridad:</b> | Alta |
| <b>Usuario:</b>   | Analista                     |                   |      |
| <b>Descripción</b>  |                              |                   |      |
| La aplicación debe ser rápida y responder de manera eficiente. Se pueden establecer requisitos de tiempo de respuesta máximo para asegurar un rendimiento óptimo.   |                              |                   |      |
| <b>Criterios de Aceptación</b>  |                              |                   |      |
| <ul style="list-style-type: none"> <li>• El cálculo de las proyecciones académicas no pasará un máximo de 5 segundos.</li> <li>• El tiempo de respuesta a las solicitudes del usuario no sobrepasará los dos segundos en al menos el 95% de solicitudes.</li> </ul> |                              |                   |      |

**Tabla 2.9** Requerimiento no Funcional de compatibilidad.

| <b>Historia de Usuario #9</b>   |                                 |                   |      |
|---|---------------------------------|-------------------|------|
| <b>Nombre:</b>  | Compatibilidad de la aplicación |                   |      |
| <b>Código:</b>  | RNF04                           | <b>Prioridad:</b> | Alta |
| <b>Usuario:</b>   | Analista                        |                   |      |
| <b>Descripción</b>  |                                 |                   |      |
| Se requiere que la aplicación sea compatible con el Sistema Operativo Windows debido a la preferencia del usuario que manipulará la aplicación. |                                 |                   |      |
| <b>Criterios de Aceptación</b>  |                                 |                   |      |
| <ul style="list-style-type: none"> <li>• El prototipo se codificará como una aplicación de escritorio.</li> </ul>                               |                                 |                   |      |

**Tabla 2.1** Requerimiento Funcional de validación de datos.

| <b>Historia de Usuario #1</b>  |                     |                   |      |
|--|---------------------|-------------------|------|
| <b>Nombre:</b>   | Validación de datos |                   |      |
| <b>Código:</b>   | RF01                | <b>Prioridad:</b> | Alta |
| <b>Usuario:</b>  | Analista            |                   |      |
| <b>Descripción</b>   |                     |                   |      |
| El sistema realizará una validación de datos para evitar errores y mantener un formato adecuado de los datos.  |                     |                   |      |
| <b>Criterios de Aceptación</b>   |                     |                   |      |
| <ul style="list-style-type: none"> <li>• El prototipo contará con una validación previa al ingresar los datos de los índices académicos, para evitar datos erróneos, vacíos, o diferentes al tipo de dato requerido, antes de ser registrados en la base de datos.</li> <li>• El formulario contará con formatos correspondientes al tipo de datos, ya sea numérico o texto para que no se ingrese cualquier tipo de datos.</li> <li>• Antes de guardar los datos, el sistema verificará datos faltantes o erróneos, y en caso de error se notificará mediante mensajes emergentes.</li> </ul> |                     |                   |      |

#### 2.1.4. DISEÑO DE LA CAPA DE DATOS

El diseño de la Capa de Datos es muy importante para garantizar una eficiente gestión de los datos en un sistema o aplicación. Al momento de diseñar correctamente se puede lograr un diseño robusto y escalable que satisfaga las necesidades de la aplicación a largo plazo.

### 2.1.4.1. Diagrama Relacional de Base de Datos

Para empezar con el diseño del diagrama relacional del proyecto, se opta por crear inicialmente 5 tablas principales las cuales toman los nombres de: **Materia**, **Malla**, **Prerrequisito**, **Correquisito**, e **Indice\_Academico**; todas estas tablas cuentan con sus respectivas claves primarias necesarias para identificar de forma exclusiva la fila de una tabla; también cuentan con sus respectivas columnas que contendrán los campos de las entidades a registrar [18].

Las tablas contarán con una multiplicidad de uno a varios entre sí, siendo la tabla **Materia** la encargada de recibir las claves foráneas de las demás tablas, a excepción de la tabla **Malla**, ya que esta tendrá una multiplicidad diferente.

Dado que se menciona la excepción de la multiplicidad uno a varios entre las tablas **Materia** y **Malla**, se asigna una relación de varios a varios entre estas tablas. Por lo tanto, se crea una tabla intermedia o de vinculación, llamada **Materia\_Malla**, que contendrá claves foráneas y un índice principal. Además, se añade una nueva columna que indicará al campo **nivel** en la nueva entidad.

Tomando en cuenta las consideraciones necesarias, el diseño de la Capa de Datos se obtiene como se muestra en la Figura 2.2:

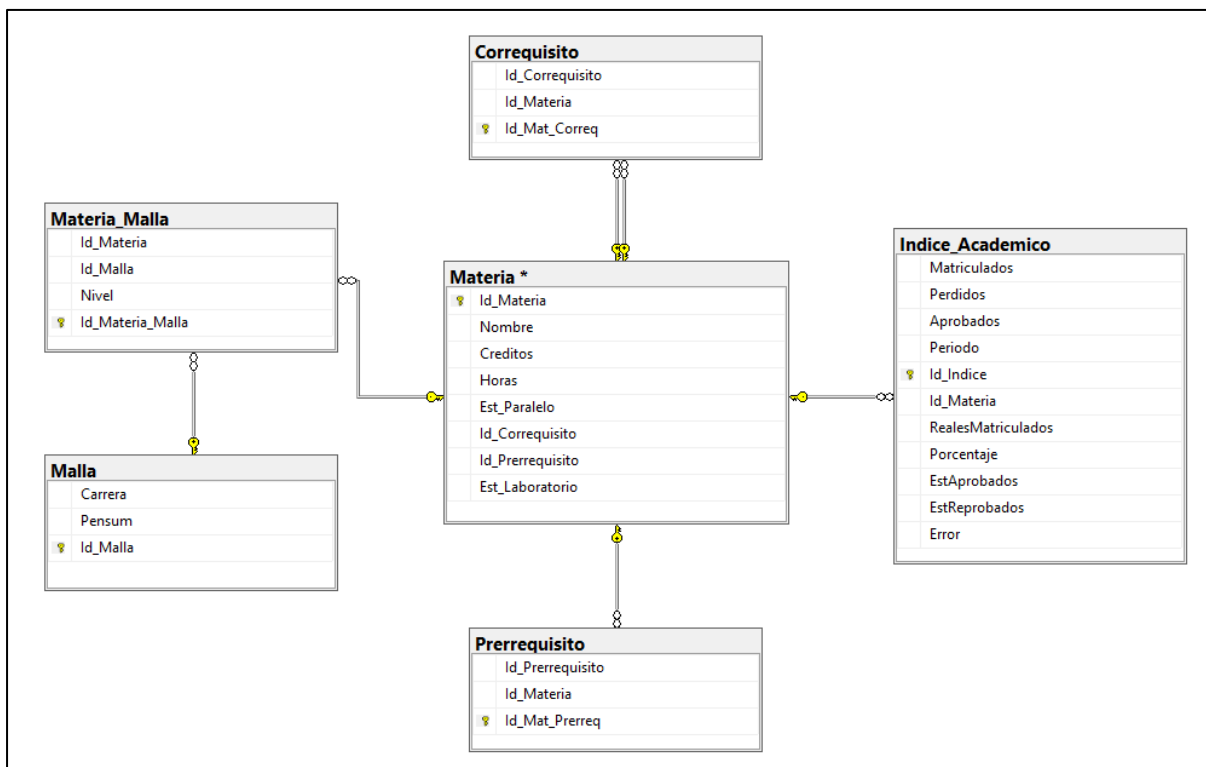


Figura 2.2. Diseño del diagrama Relacional de la Base de Datos.

### 2.1.5. DISEÑO DE LA CAPA DE NEGOCIO

El diseño de la Capa de Negocio es una parte clave para el desarrollo de la aplicación ya que se encarga de la lógica del negocio y además define las reglas necesarias para el funcionamiento de la aplicación.

#### 2.1.5.1. Diseño del Diagrama de Clases

El diagrama de clases representa una visualización general de las entidades y objetos implementados en el sistema; muestra sus atributos, métodos y conexiones entre sí permitiendo que se pueda modelar e interpretar el comportamiento del sistema de una manera gráfica y estructurada [19].

En la **Figura 2.3**, se puede observar el diseño del diagrama de clases de la Capa de Negocio, el cual constituye las clases **Malla**, **Materia**, **Índice** y **Proyección**, con sus respectivas relaciones, además cuenta con las clases **Correquisito** y **Prerrequisito**, que son fundamentales para poder manejar el proceso de proyección en la parte del negocio del prototipo.

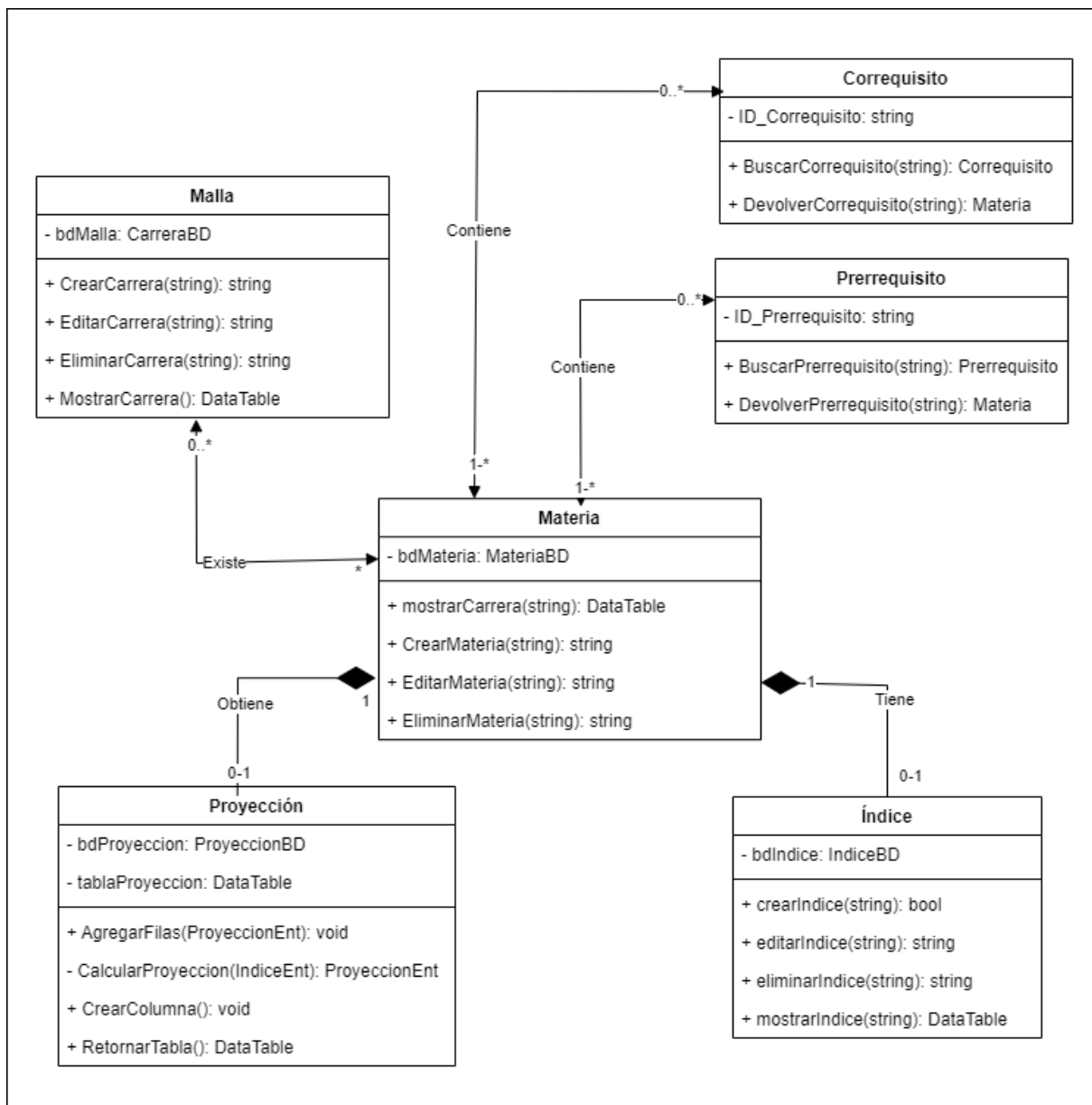
De forma detallada la clase **Materia** se relaciona mediante composición [20] con la clase **Proyección** y la clase **Índice**, es decir esta relación indica que la clase **Materia** se comporta como un “todo”, y las clases **Proyección** e **Índice** son partes constituyentes de la clase **Materia**; esta relación de composición implica una fuerte dependencia y acoplamiento entre las clases involucradas, ya que las partes (Proyección e Índice) no pueden existir sin la clase **Materia**.

Por otra parte, la clase **Malla** y **Materia** cuentan con una asociación estructural [20], es decir una conexión entre objetos; esta asociación es bidireccional y tiene una multiplicidad de cero a varios en sentido Materia-Carrera, y de varios en sentido Carrera-Materia.

Como ya se mencionó, las clases **Prerrequisito** y **Correquisito** son necesarias para poder realizar el cálculo de la proyección académica, por tal razón, estas clases se encuentran relacionadas con la clase **Materia**, manteniendo una multiplicidad de una a varias en sentido Materia-Correquisito y de cero a varias en sentido Correquisito-Materia. Esta misma multiplicidad se mantiene con la clase Prerrequisito.

De esta manera es como ha sido detallado el diseño del diagrama de clases, de tal forma que permita obtener el resultado esperado para la aplicación a realizar.

A continuación, se muestra el diagrama de clases en la Figura 2.3:

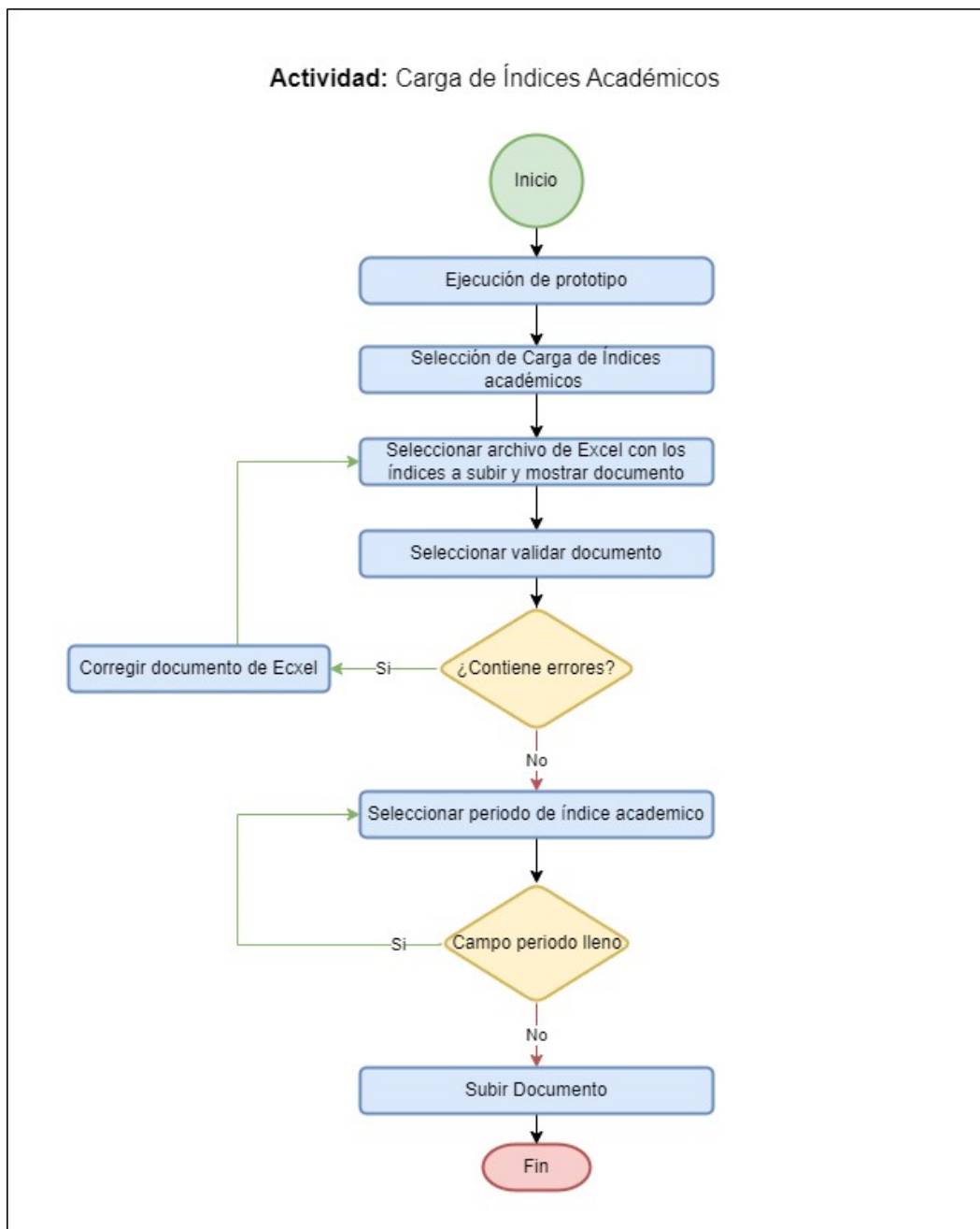


**Figura 2.3.** Diseño del diagrama de clases de la Capa de Negocio.

### 2.1.5.2. Diagrama de Actividades [21]

Un diagrama de actividades es una representación gráfica que se utiliza para modelar el comportamiento o flujo de trabajo de un proceso o sistema. Es una herramienta efectiva la cual comunica los procesos, procedimientos y algoritmos complejos de una manera clara y comprensible. Esta herramienta es utilizada por los desarrolladores de software y por profesionales de negocios, ya que permite la visualización y documentación del comportamiento del sistema, permitiendo obtener mejores resultados en la toma de decisiones.

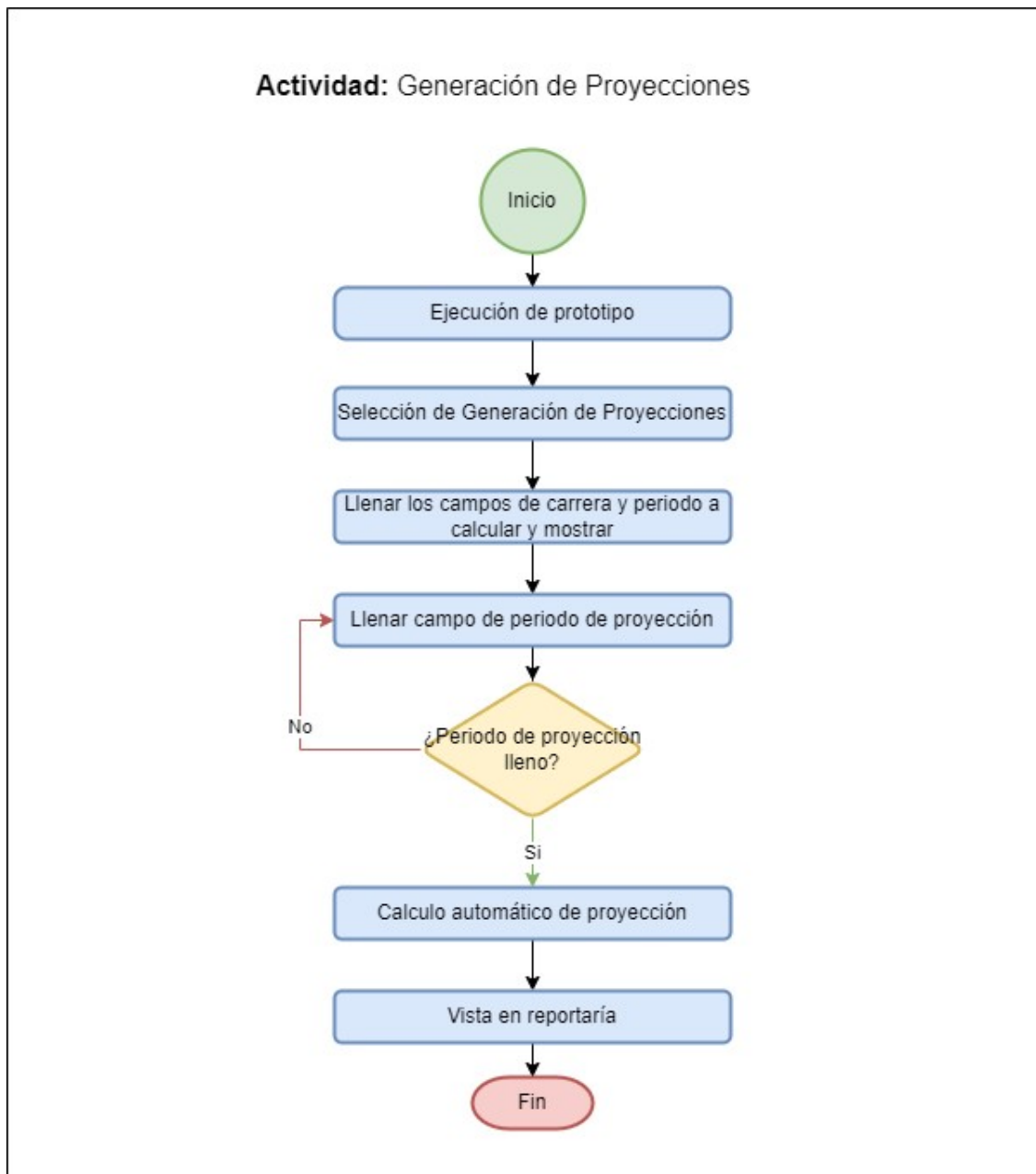
Para este proyecto se han especificado 4 diagramas de actividades correspondientes a las principales tareas o secciones que se pueden realizar al momento de que la aplicación se encuentre en ejecución.



**Figura 2.4.** Diagrama de carga de Índices Académicos.

En la Figura 2.4, se encuentra representada la actividad de “Carga de Índices Académicos”, y es la encargada de subir la información necesaria para poder almacenar la información en la base de datos y a partir de ahí poder realizar el cálculo de las proyecciones.

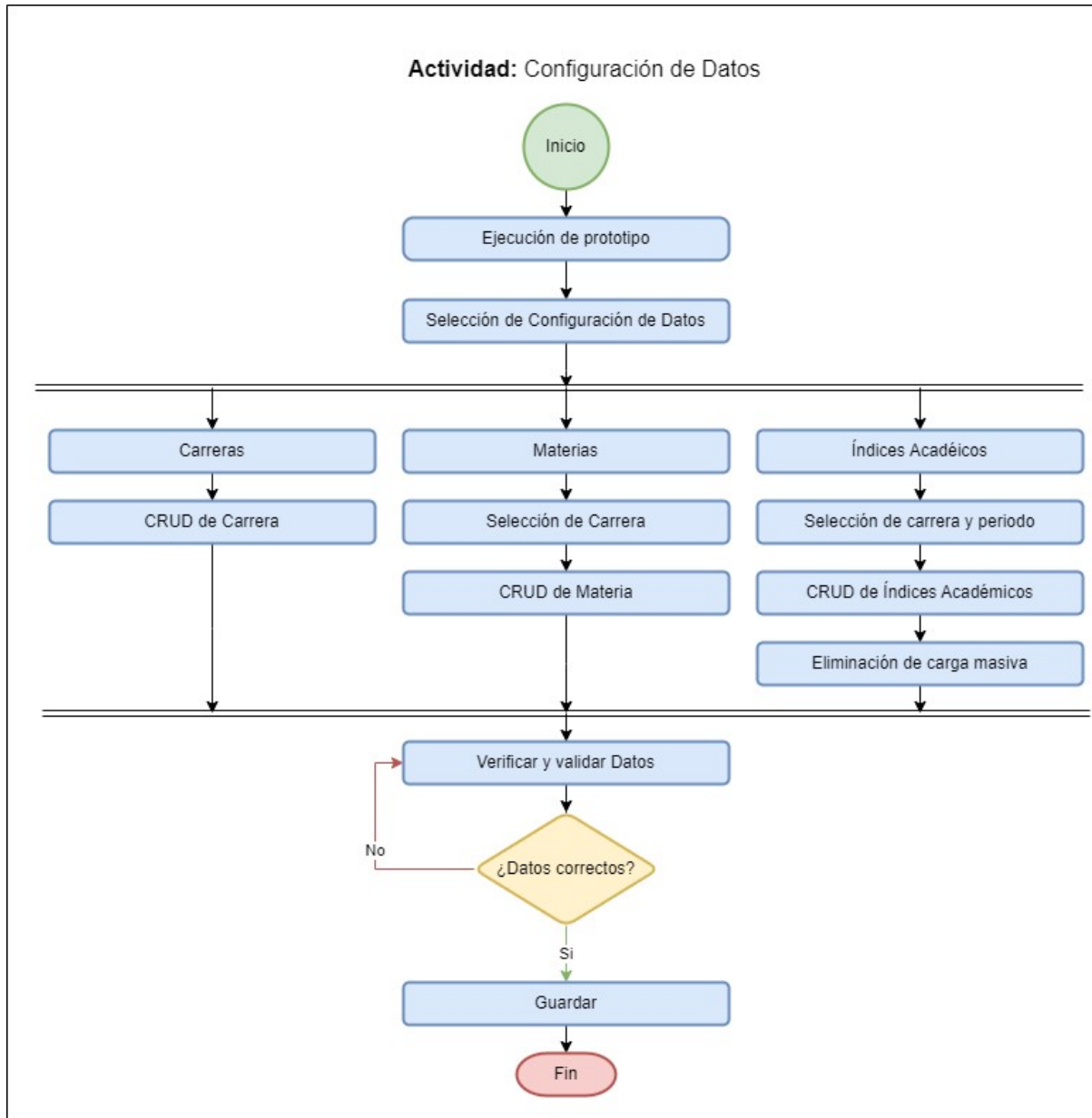
La siguiente actividad corresponde a la sección Generación de Proyecciones, esta se la representa en la Figura 2.5:



**Figura 2.5.** Diagrama de generación de proyecciones

La actividad representada en el diagrama de la Figura 2.5 es de vital importancia, ya que desempeña un papel fundamental en la obtención del resultado esperado. Sin esta actividad específica, el prototipo no lograría cumplir con el resultado final que se requiere. Puede ser una etapa crítica del proceso que determina la funcionalidad, eficiencia o rendimiento clave

del prototipo. Su correcta ejecución y atención adecuada son fundamentales para el éxito general del proyecto.

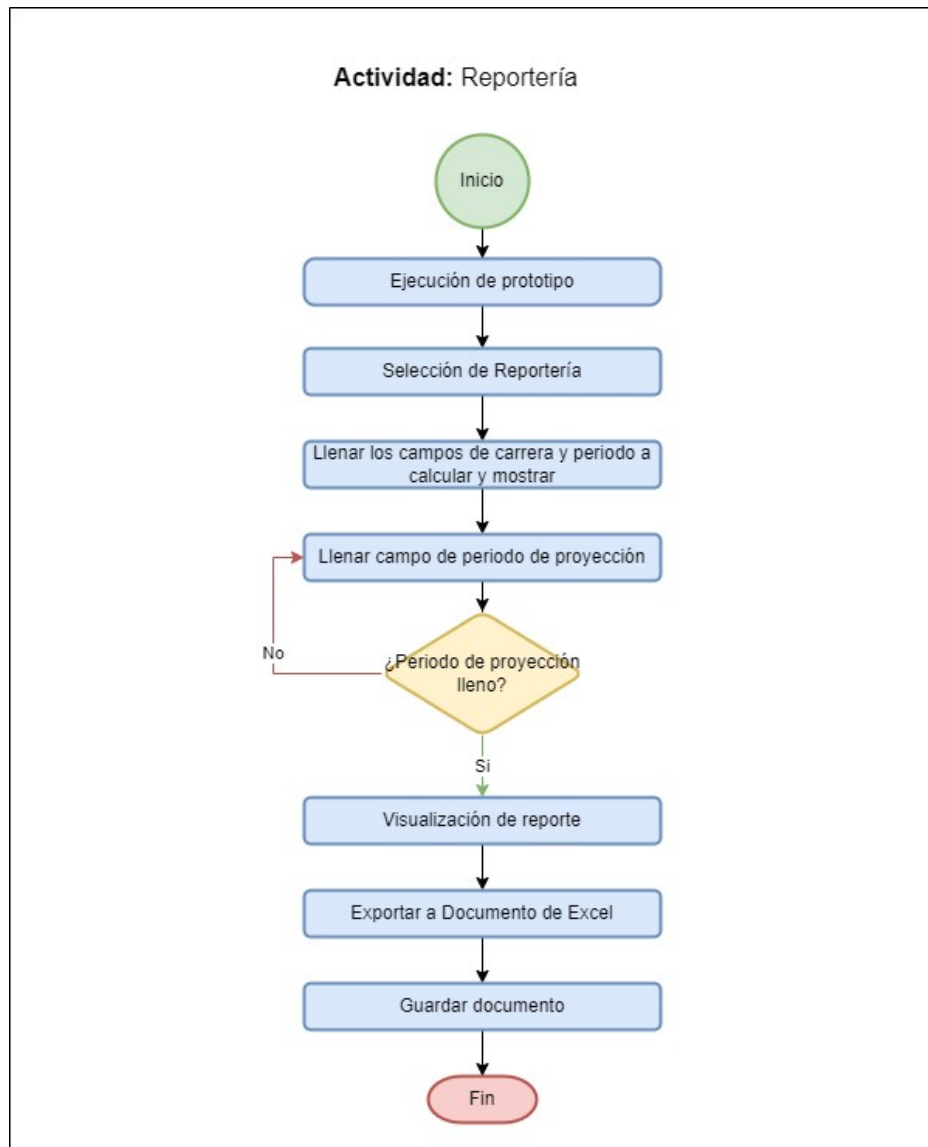


**Figura 2.6.** Diagrama de Configuración de Datos

El diagrama representado en la Figura 2.6 se centra principalmente en las operaciones de creación, actualización y eliminación de datos. Esta actividad es crucial para establecer la comunicación con la base de datos, lo que a su vez permite al usuario interactuar con dicha base y alimentarla con la información necesaria para el funcionamiento del sistema.



La actividad descrita desempeña un papel esencial al permitir la comunicación bidireccional entre el sistema y la base de datos, lo que garantiza la entrada y actualización de información necesaria para el correcto funcionamiento del sistema.



**Figura 2.7.** Diagrama de Reportería

El diagrama presentado en la Figura 2.7 ilustra la actividad de generación de informes, la cual tiene como objetivo obtener los resultados de la proyección de una manera amigable y accesible. Esta actividad ofrece la posibilidad de obtener los resultados en formato PDF o exportarlos a un documento de Excel.

En otras palabras, el diagrama describe el proceso de generar informes a partir de los datos obtenidos, con el fin de presentarlos de una manera visualmente atractiva y fácil de entender.

Los resultados pueden ser exportados en formato PDF, lo que permite su visualización e impresión en un formato estándar y portátil. Además, se ofrece la opción de exportar los informes a un documento de Excel, lo que facilita su manipulación y análisis posterior.

### 2.1.6. DISEÑO DE LA CAPA DE PRESENTACIÓN

La Capa de Presentación se refiere a la interfaz gráfica que permite al usuario ejecutar acciones para automatizar tareas en el sistema realizado. La función principal de esta capa es interactuar como un punto de interacción entre el usuario y el sistema [22].

Esta capa se comunica únicamente con la Capa de Negocio, lo cual significa que su función principal es transmitir instrucciones desde el usuario hacia la Capa de Negocio para su procesamiento, asegurando que las acciones realizadas se ejecuten adecuadamente y se reflejen en la funcionalidad del sistema.

#### 2.1.6.1. Interfaz Gráfica

La interfaz gráfica del prototipo estará conformada por un formulario principal, el cual trabajará como un formulario MDI (*Multiple Document Interface*), es decir, como un contenedor de otros formularios. Este formulario será la pantalla principal al iniciar la aplicación y a partir de ello, se crearán los formularios hijos de manera que se pueda interactuar fácilmente con el usuario y la aplicación.



**Figura 2.8.** Diseño de interfaz de pantalla principal.

Como se puede observar, la Figura 2.8, representa la pantalla principal de la aplicación, en la parte superior se encuentra una barra de título, la cual permite contener el nombre de cada formulario que se va a abrir, y adicionalmente, los botones de control de pantalla (minimizar, maximizar y cerrar). En la parte izquierda, se encuentra el menú de opciones lo cual permitirá abrir los formularios hijos, que se indicarán en la parte central del formulario. Este ejemplo de diseño será el formulario padre o MDI.

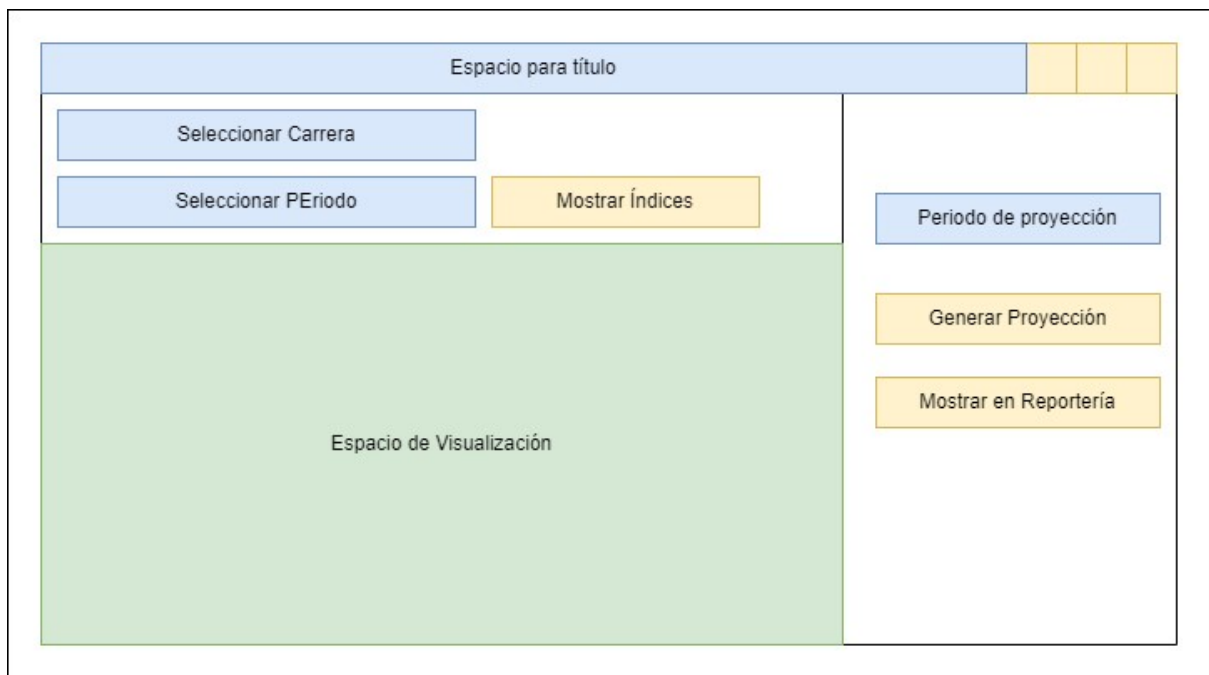
El primer formulario hijo que representará a la Carga de Índices Académicos se lo muestra representado en la Figura 2.9:

El diagrama muestra un formulario con una barra de título azul que contiene el texto "Espacio para título" y tres botones de control de ventana (minimizar, maximizar, cerrar) amarillos. Debajo de la barra de título, hay un campo de texto azul "Ruta de archivo" y un botón amarillo "Mostrar Documento". A la izquierda de "Ruta de archivo" hay un campo "Hojas encontradas" y a la derecha un campo "Periodo". Debajo de "Hojas encontradas" y "Periodo" hay un botón amarillo "Validar Documento". El centro del formulario es un gran espacio verde etiquetado como "Espacio de Visualización". En la parte inferior derecha hay un botón amarillo "Subir Documento".

**Figura 2.9.** Diseño del formulario Carga de Índices Académicos.

El formulario de la Figura 2.9, está compuesto por una barra de título, adicionalmente se encuentra en la parte superior izquierda el espacio para escoger y manipular el archivo que contiene los índices académicos, en este caso se indicará la ruta del archivo, y se permitirá seleccionar las hojas del documento de Excel, además se elegirá también el periodo al cual representa los índices del documento. Para la parte superior derecha se tiene los botones que llevaran acciones de manipulación del sistema, como por ejemplo la validación, la vista y la carga del documento. Por último, en la parte central se proyectará el documento que se ha seleccionado para realizar la carga de los índices.

El segundo formulario hijo, representará la sección de Generar Proyecciones, y se lo puede observar en la Figura 2.10:

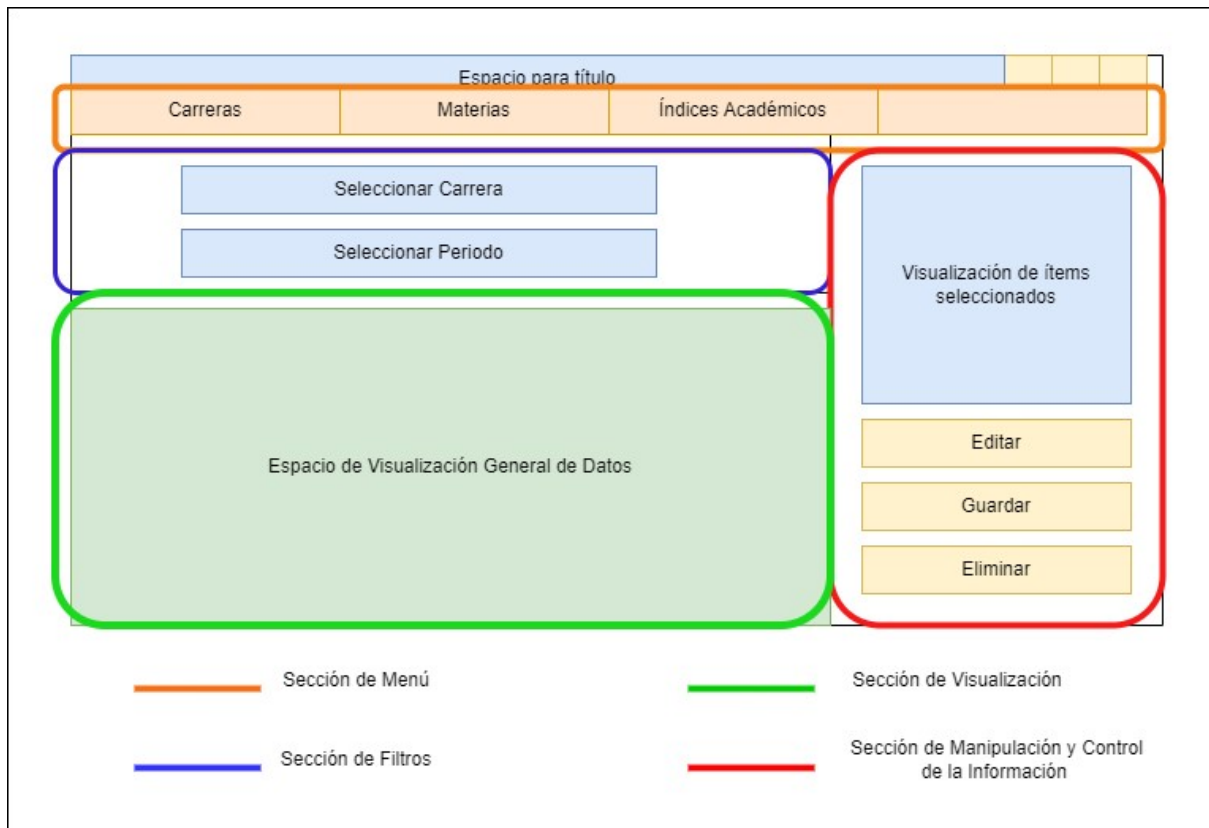


**Figura 2.10.** Diseño del formulario Generar Proyecciones.

En el bosquejo de la Figura 2.10 se puede observar en la parte superior una sección de filtros, en donde se permitirá escoger la carrera y el periodo de los índices académicos que se requieran calcular la proyección; una vez seleccionados se podrá visualizar los datos correspondientes a los filtros seleccionados en el espacio central. En la sección de la parte izquierda de tendrá que escoger el periodo de proyección y mediante los botones adyacentes se podrá realizar el cálculo y abrir el resultado en la sección de Reportería de la aplicación.

El diseño antes mostrado de la Figura 2.10 servirá también como plantilla para las secciones de Configuración de Datos y Reportería de la aplicación; de igual manera contará con la sección de filtros en la parte superior, la sección de visualización en la parte central y la sección de manipulación y control de la información en la parte lateral izquierda. La diferencia será en la parte superior del formulario, ya que se añadirá un menú el cual indicará el conjunto de datos de los que se requiera hacer CRUD, como son las carreras, las materias y los índices académicos. Otra diferencia es que, en la sección de manipulación y control de la información, se tendrá los espacios necesarios para ingresar los datos de cada ítem seleccionado que se desee modificar y seguido de estos espacios, los botones que realizarán las respectivas acciones como crear, editar, guardar y eliminar.

Esta representación del bosquejo para la sección de Configuración de Datos, se lo puede observar en la Figura 2.11 que se encuentra a continuación:



**Figura 2.10.** Diseño del formulario Configuración de Datos.

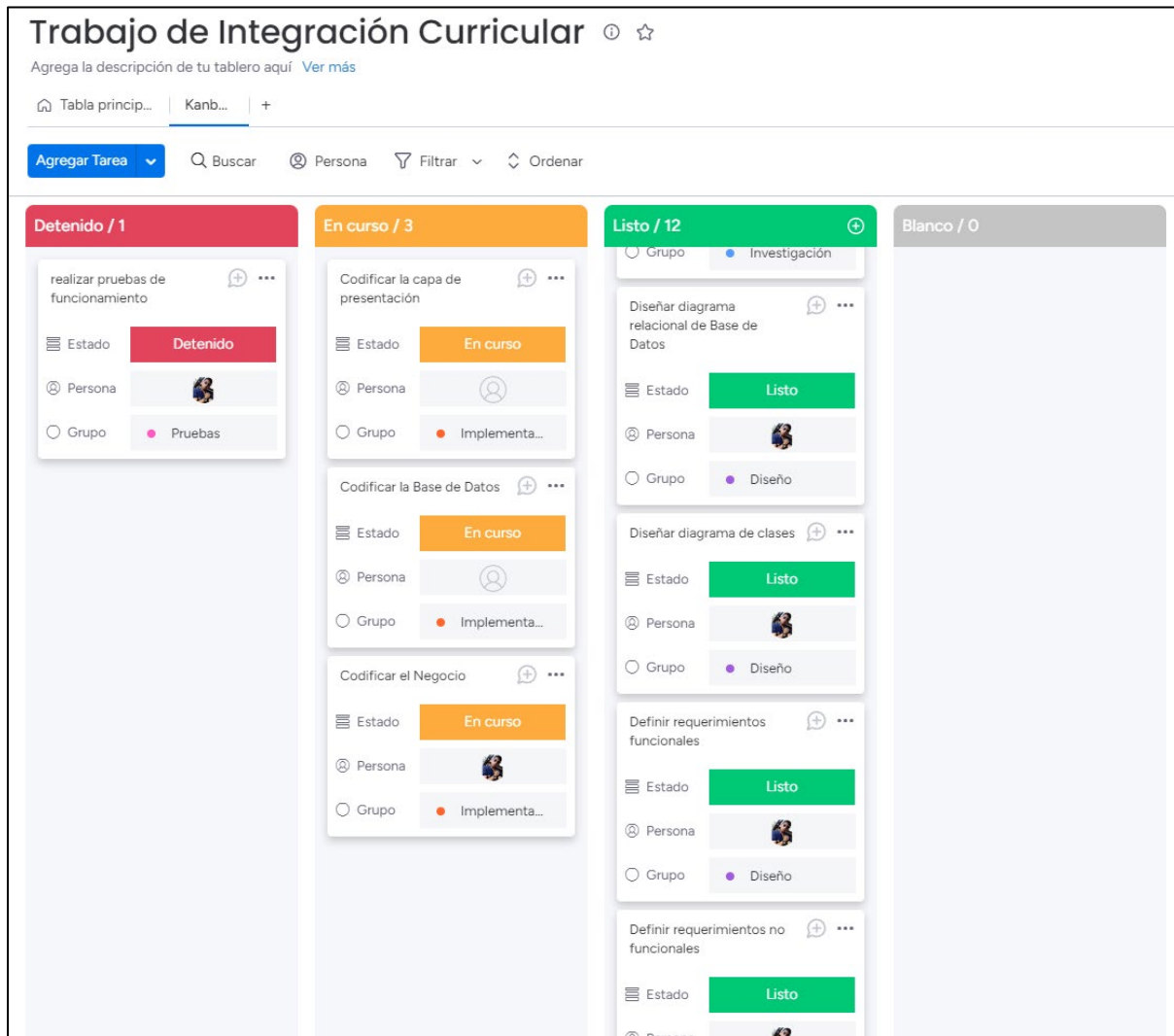
## 2.2. IMPLEMENTACIÓN

En esta sección, se detallará el proceso de codificación de las principales funciones implementadas en la creación de la aplicación de Proyección de Cursos Académicos. Se comienza con la actualización del tablero Kanban, donde se registrarán y gestionarán las tareas y el progreso de las actividades establecidas. A continuación, se describirá el proceso de codificación de la base de datos, aplicando el diseño establecido en el **Apartado 2.1.4**. Además, se mostrará parte de la implementación lógica del negocio de la aplicación, destacando los algoritmos y reglas que se aplicaron para obtener el funcionamiento adecuado y la precisión de los cálculos. Por último, se abordará la interacción de las interfaces en la capa de presentación, describiendo cómo se diseñaron y desarrollaron las interfaces de usuario para garantizar una comunicación intuitiva entre el usuario y la aplicación.

### 2.2.1. ACTUALIZACIÓN DEL TABLERO KANBAN

Se realiza la actualización del tablero para reflejar el progreso de las tareas asignadas.

En esta fase de implementación las tareas asignadas en la fase de Diseño pasan a la etapa de “Listo”, y las actividades que anteriormente estaban establecidas en la etapa “Detenido” migrarán a la etapa “En curso” con el fin de actualizar el tablero. Como se puede observar en la Figura 2.11, se tienen 3 actividades que están en curso las cuales son: codificar la Capa de Presentación, codificar la Base de Datos, y codificar la Capa de Negocio.



**Figura 2.11.** Actualización del tablero Kanban en etapa de implementación.

## 2.2.2. CODIFICACIÓN DE LA BASE DE DATOS

### 2.2.2.1. Creación de la Base de Datos en SQL Server

En esta parte de la implementación, se inicia con la creación de la base de datos utilizando el software Microsoft SQL Server. Como es común en este software, la implementación se puede llevar a cabo mediante código y también mediante la interfaz gráfica. En este caso, se

ha optado por utilizar la interfaz gráfica para implementar las tablas correspondientes al diagrama relacional de la Figura 2.2.

Como ejemplo se tomará en cuenta la tabla "Materia" la cual consta de 8 campos, donde 1 campo se designa como clave principal y 2 campos se establecen como claves foráneas para mantener relaciones con otras tablas. A través de la interfaz gráfica, se puede asignar el tipo de dato para cada campo. En este caso, las claves se definen como "nchar(10)" para representar cadenas de texto de longitud 10, los nombres o cadenas de texto se asignan como "varchar(100)", y los números enteros se especifican como "int".

Además, durante la implementación, se define si cada campo permite o no valores nulos, lo cual es crucial para mantener una estructura adecuada en el registro de los datos. La creación de la tabla "Materia" mediante la interfaz gráfica se la puede observar en la Figura 2.12:

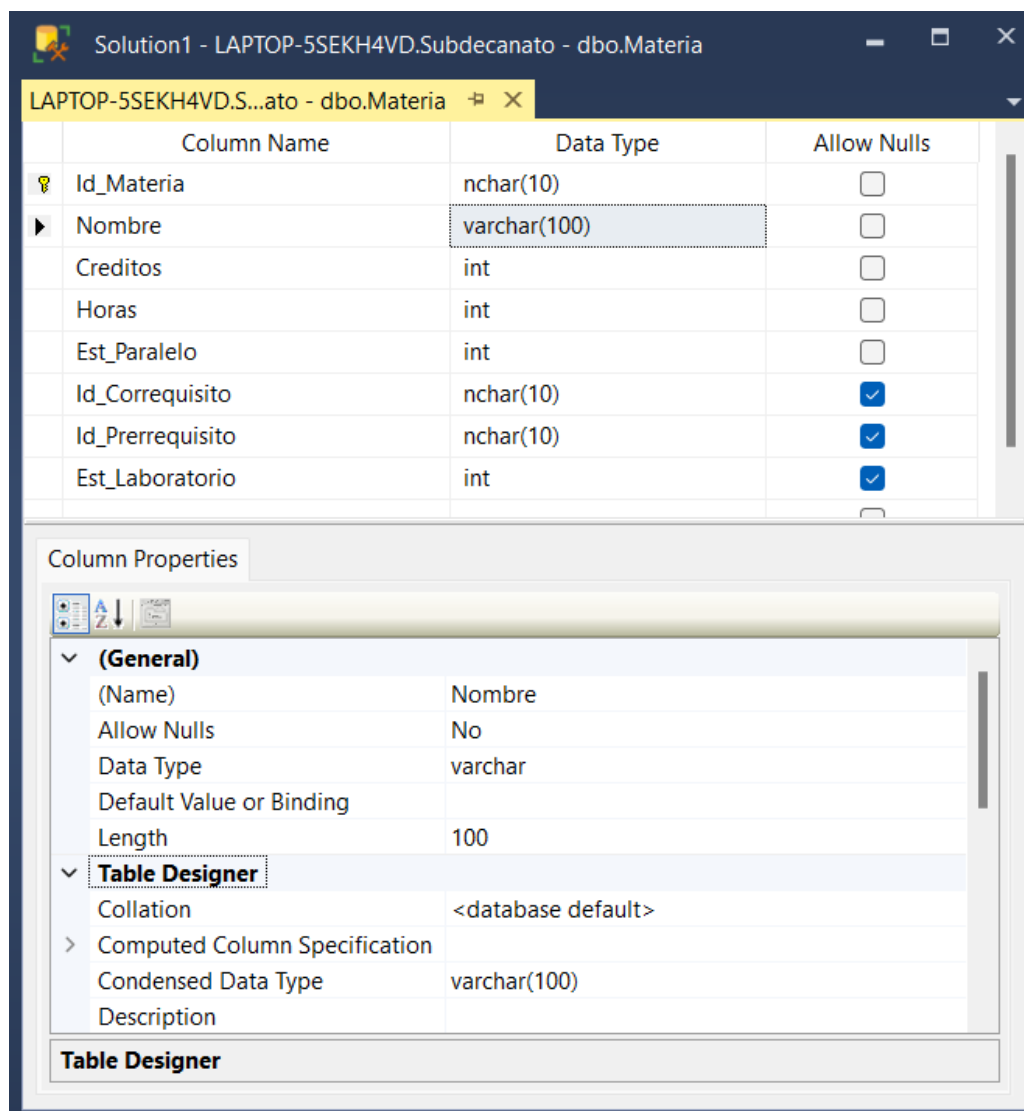
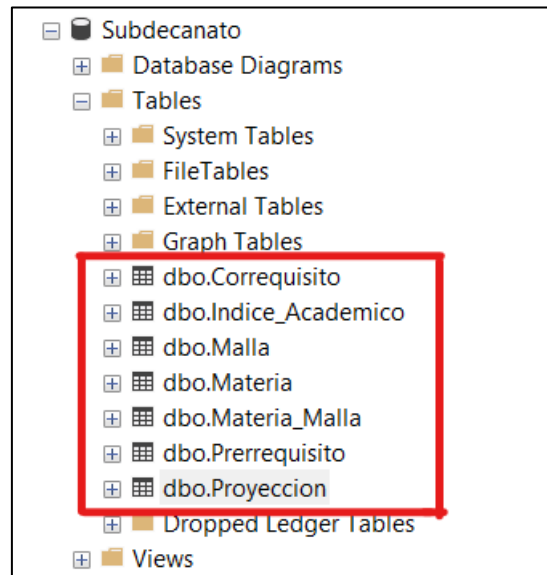


Figura 2.12. Implementación de la tabla Materia mediante interfaz gráfica de SQL Server.

El uso de la interfaz gráfica en el software simplifica el proceso de implementación de tablas, y facilita la configuración de los campos con los tipos de datos correspondientes, asegurando una estructura consistente y coherente para el almacenamiento de la información. De esta manera se han creado las tablas necesarias para la aplicación el cual conforman 7 tablas las cuales se las puede observar en la Figura 2.13:



**Figura 2.13.** Tablas implementadas en la Base de Datos.

El uso de Procedimientos Almacenados es fundamental al crear una base de datos, y por esta razón se han programado varios procedimientos almacenados para esta aplicación. Estos procedimientos se centran principalmente en realizar operaciones CRUD en las diferentes tablas creadas. Además, se han implementado procedimientos almacenados específicos para satisfacer las necesidades particulares de la aplicación, como la búsqueda de datos o la obtención de vistas.

Como ejemplo, se considera la codificación del procedimiento almacenado "ingreso\_indice", el cual es necesario para la interacción en la aplicación. El proceso de creación comienza definiendo el nombre del procedimiento y los parámetros que aceptará. Estos parámetros son utilizados para recibir valores de entrada que se utilizarán en las operaciones dentro del procedimiento. A continuación, se especifica la estructura del procedimiento, que incluye las instrucciones SQL necesarias para realizar las acciones deseadas. En este caso se ha implementado la función para ingresar los índices académicos en la tabla mediante el comando "INSERT INTO" como se muestra en el Código 2.1:



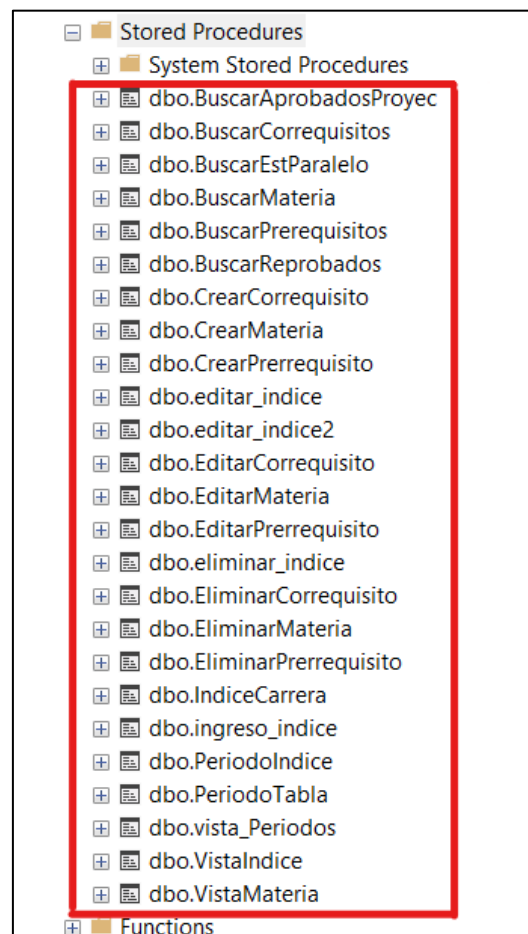
```

--PROCEDIMIENTOS ALMACENADOS---
--Ingreso de índices academicos
CREATE PROCEDURE ingreso_indice
    @matriculados int,
    @perdidos int,
    @aprobados INT,
    @periodo VARCHAR(20),
    @id_Materia nchar(10),
    @reales int,
    @porcentaje float,
    @estAprobados int,
    @estReprobados int
AS
INSERT INTO Indice_Academico
(Matriculados,Perdidos,Aprobados,Periodo,Id_Materia,RealesMatriculados,Porcentaje,
EstAprobados,EstReprobados)
VALUES(@matriculados,@perdidos,@aprobados,@periodo,@id_Materia,@reales,@porcentaje
,@estAprobados,@estReprobados)

```

**Código 2.1.** Procedimiento para insertar datos en la tabla “Indice\_Académico”.

Para la aplicación se han creado los diferentes procedimientos almacenados necesarios para que la aplicación fluya de mejor manera, por tal razón se ha obtenido un total de 25 procedimientos almacenados, que se muestran en la Figura 2.14:



**Figura 2.14.** Procedimientos Almacenados implementadas en la Base de Datos.

### 2.2.2.2. Codificación de la Capa de Acceso a Datos

Para la creación de esta capa, se ha creado una Solución en Visual Studio, la cual contendrá varias bibliotecas de clases, donde una de ellas tendrá el nombre “**Capa\_Datos**”, y esta contendrá las clases necesarias para que la aplicación interactúe con la base de datos.

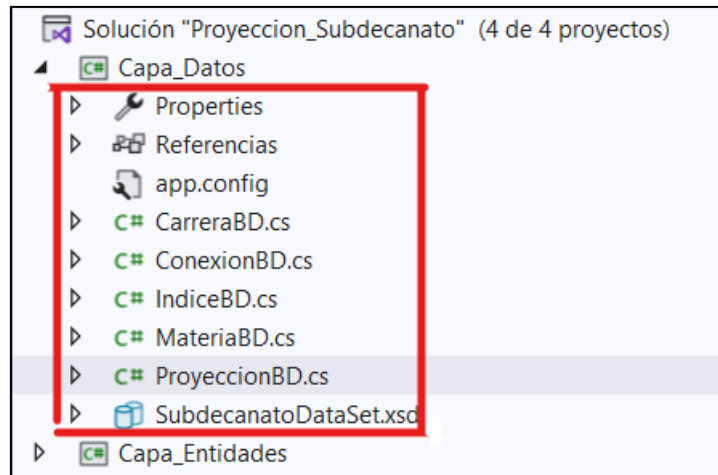


Figura 2.15. Clases de la Biblioteca “Capa\_Datos”.

En la biblioteca se crean 5 clases, la más importante toma el nombre de “**ConexionBD**”, esta clase es primordial ya que tiene los métodos necesarios para comunicarse con el servidor de la base de datos.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Data;
7 using System.Data.SqlClient;
8
9 namespace Proyeccion_Subdecanato
10 {
11     class ConexionBD
12     {
13         private SqlConnection conexion = new
14             SqlConnection("Data Source=LAPTOP-5SEKH4VD;Initial " +
15                 "Catalog=Subdecanato;Integrated Security=True");
16
17         public SqlConnection AbrirConexion()
18         {
19             if (conexion.State == ConnectionState.Closed)
20                 conexion.Open();
21             return conexion;
22         }
23         public SqlConnection CerrarConexion()
24         {
25             if (conexion.State == ConnectionState.Open)
```

```

26         conexion.Close();
27         return conexion;
28     }
29     public SqlConnection Conexion()
30     {
31         return conexion;
32     }
33 }
34 }

```

**Código 2.2.** Código de la clase "ConexionBD".

En la biblioteca de clases mencionada, también se encuentra la clase **ProyeccionBD**. Esta clase contiene un objeto de la clase **ConexionBD** que se utiliza para abrir y cerrar conexiones con la base de datos. Además, la clase **ProyeccionBD** tiene sus propios métodos que están vinculados a un procedimiento almacenado para obtener registros de la base de datos. Para lograr esto, se utiliza un objeto del tipo "SqlCommand" que, a través de su método "ExecuteReader", ejecuta el procedimiento almacenado y devuelve el resultado en forma de tabla. Esta tabla se almacena en un objeto del tipo "DataTable", lo que permite utilizar los datos en la parte del negocio de la aplicación.

Como ejemplo de lo mencionado anteriormente se muestra la codificación de la clase **ProyeccionBD** en el Código 2.3:

```

1 using System;
2
3 namespace Proyeccion Subdecanato
4 {
5     public class ProyeccionBD
6     {
7         private ConexionBD conexion = new ConexionBD();
8         SqlDataReader leer;
9         public DataTable BuscarPrerrequisito (String idMatreia)
10        {
11            SqlCommand comando = new SqlCommand();
12            DataTable tabla = new DataTable();
13            comando.Connection = conexion.AbrirConexion();
14            comando.CommandText = "BuscarPrerrequisitos";
15            comando.CommandType = CommandType.StoredProcedure;
16            comando.Parameters.AddWithValue("@id_Materia", idMatreia);
17            leer = comando.ExecuteReader();
18            tabla.Load(leer);
19            comando.Parameters.Clear();
20            conexion.CerrarConexion();
21            return tabla;
22        }
23        public DataTable BuscarAprobados (String idMatreia)
24        {
25            DataTable tabla1 = new DataTable();
26            SqlCommand comando1 = new SqlCommand();
27            comando1.Connection = conexion.AbrirConexion();

```

```

28         comando1.CommandText = "BuscarAprobadosProyec";
29         comando1.CommandType = CommandType.StoredProcedure;
30         comando1.Parameters.AddWithValue("@id_Materia", idMatreia);
31         leer = comando1.ExecuteReader();
32         tabla1.Load(leer);
33         comando1.Parameters.Clear();
34         conexion.CerrarConexion();
35         return tabla1;
36     }
37     public DataTable BuscarMateria(String idMatreia)
38     {
39         SqlCommand comando3 = new SqlCommand();
40         DataTable tabla2 = new DataTable();
41         comando3.Connection = conexion.AbrirConexion();
42         comando3.CommandText = "BuscarMateria";
43         comando3.CommandType = CommandType.StoredProcedure;
44         comando3.Parameters.AddWithValue("@idMateria", idMatreia);
45         leer = comando3.ExecuteReader();
46         tabla2.Load(leer);
47         comando3.Parameters.Clear();
48         conexion.CerrarConexion();
49         return tabla2;
50     }
51 }
52 }

```

**Código 2.3.** Código de la clase “ProyeccionBD”.

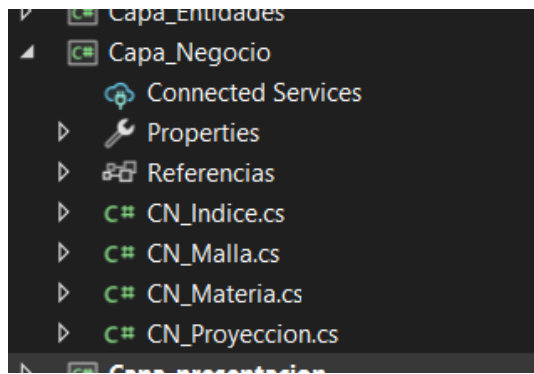
De esta manera se codifica las demás clases que se encuentran en la biblioteca “Capa Datos” haciendo uso de los procedimientos almacenados ya creados, y de los métodos de cada clase, para poder obtener y enviar los datos necesarios para continuar con el trabajo e interacción de la base de datos con la aplicación.

El resto de la codificación se encuentra en el **Anexo I**.

### 2.2.3. CODIFICACIÓN DE LA CAPA DE NEGOCIO

En este Apartado se mostrará las funciones lógicas principales que realizará el prototipo, es decir, se codificará el proceso necesario para tener los resultados esperados de la aplicación, para ello, al igual que la codificación de la capa de datos, se crea una biblioteca de clases la cual tomará el nombre de **Capa\_Negocio**, esta biblioteca de clases va a contener las clases: **CN\_Indice**, **CN\_Malla**, **CN\_Materia** y **CN\_Proyeccion**, las cuales son necesarias para implementar la parte del negocio de la aplicación.

La muestra de cómo queda estructurada la biblioteca **Capa\_Negocio**, se indica en la siguiente figura:



**Figura 2.15.** Clases de la Biblioteca “Capa\_Negocio”.

Dentro del Código 2.4, se encuentra la clase **CN\_Proyeccion**, que es una de las más importantes. Esta clase contiene el método principal llamado "CalcularProyeccion", el cual calcula la proyección académica. El método recibe un parámetro de entrada, que es un objeto “IndiceEnt” (índice académico), y devuelve un objeto “ProyeccionEnt” (proyección). En resumen, lo que hace el método es buscar los prerrequisitos existentes. A partir del prerrequisito encontrado, suma el número de estudiantes que han aprobado la materia con el número de estudiantes que la han reprobado. Este total representa el número estimado de estudiantes, el cual se utilizará para establecer la cantidad de estudiantes en cada aula mediante una división. Por último, el método almacena temporalmente los valores obtenidos en un objeto de tipo “ProyeccionEnt”, que retornará al llamar al método. El código completo de esta clase, se lo puede observar en el **Anexo I**.

```

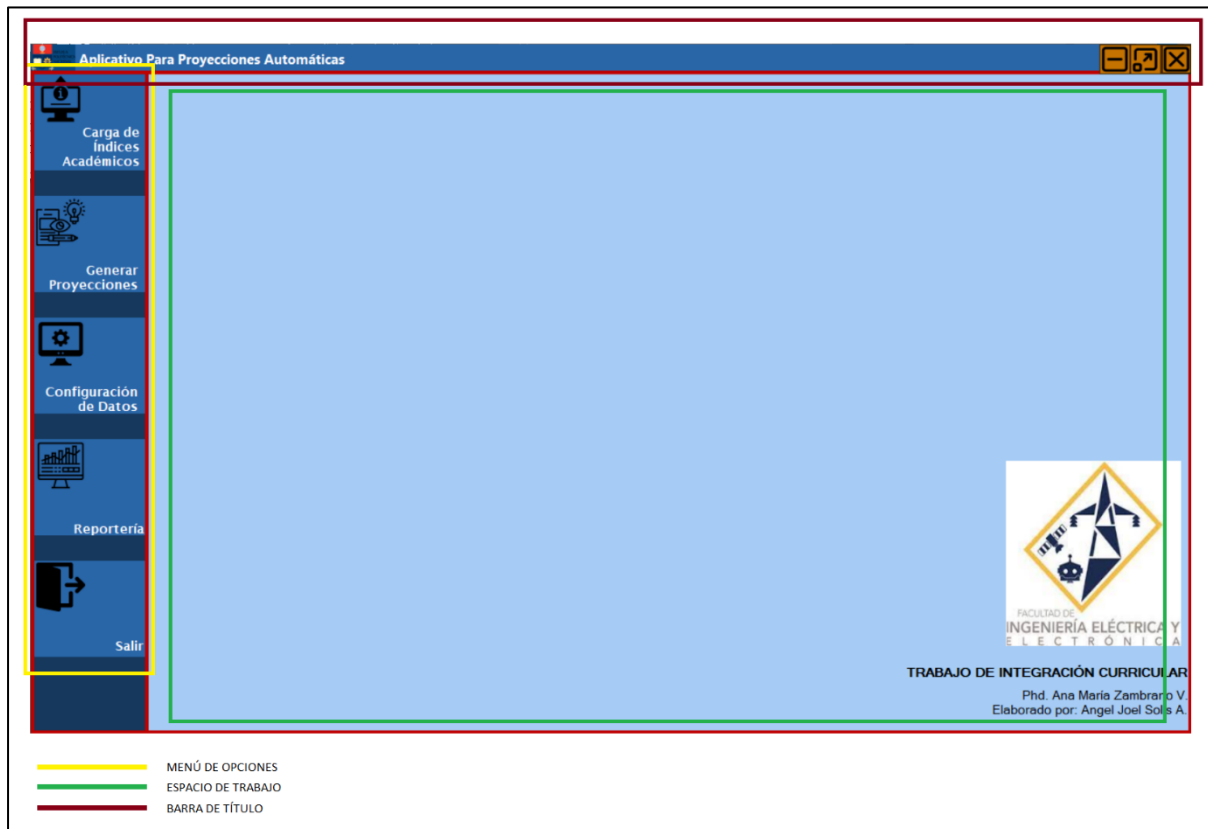
1 private ProyeccionEnt CalcularProyeccion(IndiceEnt ind)
2     {
3         DataTable d = pro.BuscarPrerrequisito(ind.codMateria);
4         ProyeccionEnt c = new ProyeccionEnt();
5         for (int i = 0; i < d.Rows.Count; i++)
6         {
7             prerrequisito = d.Rows[i]["Prerrequisitos"].ToString();
8         }
9         for (int i = 0; i < a.Rows.Count; i++)
10        {
11            aprobados = int.Parse( a.Rows[i]["Aprobados"].ToString());
12        }
13        d = pro.BuscarMateria(ind.codMateria);
14        c.codMateria = ind.codMateria;
15        c.nomMateria = ind.nomMateria;
16        c.nuevos = aprobados;
17        c.repetidos = ind.reprobados;
18        c.totales = aprobados + ind.reprobados;
19        c.numEstParalelo = estudiantes;
20        double div = (double)c.totales / estudiantes;
21        return c; }

```

**Código 2.4.** Código del método para calcular la Proyección Académica.

## 2.2.4. CODIFICACIÓN DE LA CAPA DE PRESENTACIÓN

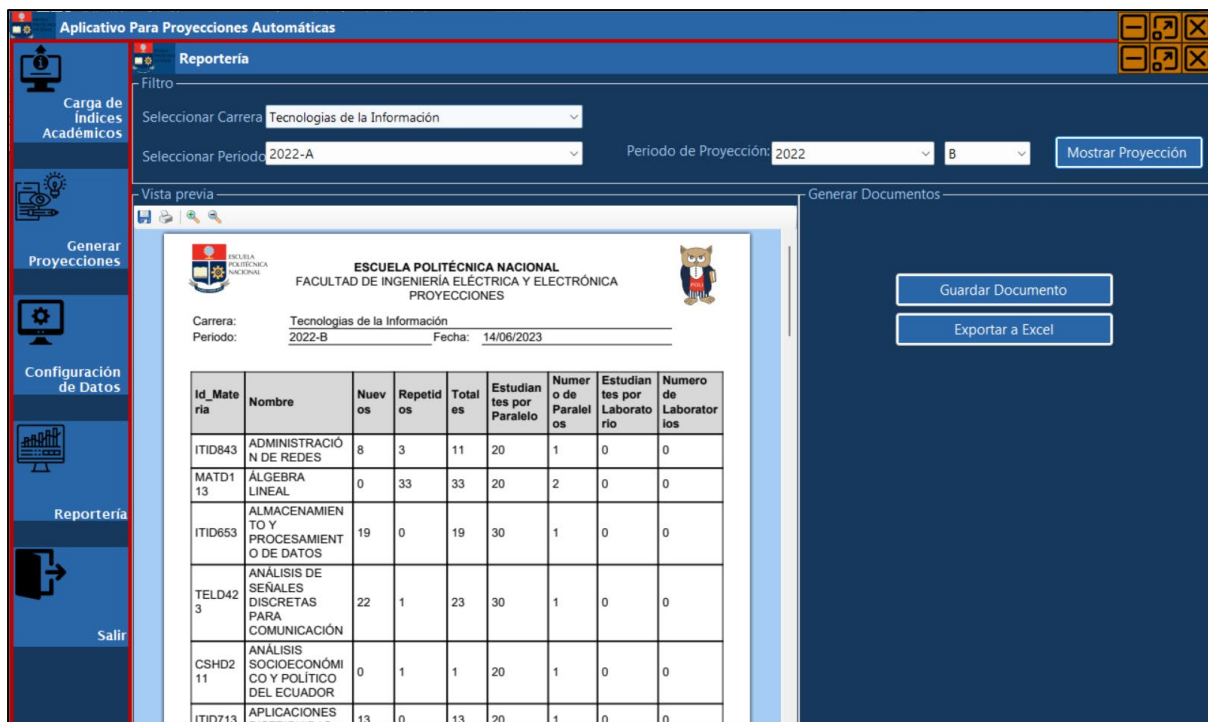
Para realizar la implementación de la interfaz gráfica, se ha seguido el diseño presentado en el **Apartado 2.1.6.1**, el cual corresponde a la pantalla principal de la aplicación que se muestra en la Figura 2.16. A partir de este formulario se dará el acceso a las demás ventanas implementadas en el prototipo las cuales servirán para realizar las acciones como: cargar índices Académicos, Generar Proyecciones, Configurar Datos, Reportería y Salir.



**Figura 2.16.** Pantalla principal del prototipo.

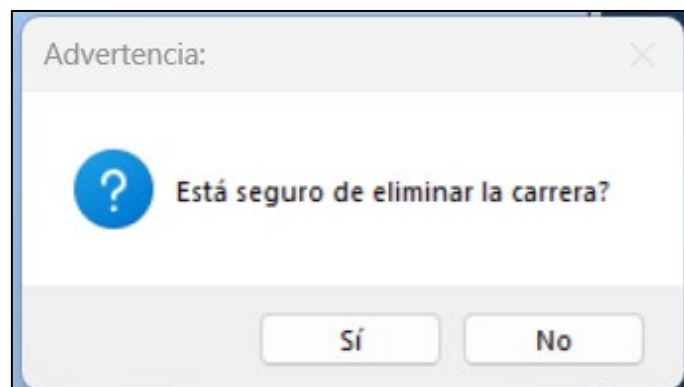
Como se mencionó en el **Apartado 2.1.6.1**, la pantalla principal (formulario padre) que se muestra en la Figura 2.16, trabaja como un contenedor. Los demás formularios se mostrarán en el espacio de trabajo como formularios secundarios (formularios hijos), y en la parte izquierda se tendrá los botones que permitirán que los formularios hijos sean visibles.

En la Figura 2.17, se muestra cómo en la pantalla principal está contenido el formulario de reportería, donde se pueden apreciar las áreas de trabajo divididas por secciones. Además, se muestra una previsualización del archivo PDF generado por la aplicación, que es el documento que contine los resultados de la proyección académica. La codificación para crear y mostrar el documento PDF, se encuentra en el **Anexo I**.



**Figura 2.17.** Pantalla principal y formulario de reportaría.

Dado que se requiere realizar validaciones antes de llevar a cabo acciones en partes importantes de la aplicación, se ha implementado mensajes emergentes mediante cuadros de texto que se mostrarán en la pantalla actual en la que se esté trabajando, estos pueden advertir, informar y preguntar antes de realizar una acción en la aplicación. Un ejemplo de estos cuadros informativos se puede observar en la Figura 2.18:



**Figura 2.18.** Pantalla emergente de error.

Para implementar estos mensajes emergentes, se utiliza el comando `“MessageBox()”`, añadiendo los parámetros necesarios, como título, mensaje, icono y botones de respuesta. En el Código 2.5, se puede indicar un fragmento de código que ejemplifica la implementación de estas ventanas o mensajes emergentes.

Las demás implementaciones se pueden observar en el código adjuntado en el **Anexo I**.

```
1 private void btnEliminarMalla_Click(object sender, EventArgs e)
2     {
3         string message = "Está seguro de eliminar la carrera?";
4         string title = "Advertencia:";
5         MessageBoxButtons buttons = MessageBoxButtons.YesNo;
6         DialogResult result = MessageBox.Show(message, title,
7             buttons, MessageBoxIcon.Question);
8         if (result == DialogResult.Yes)
9             {
10                CN_Malla malla = new CN_Malla();
11                string id = dgvMallas.CurrentRow.Cells["Id_Malla"].
12                    Value.ToString();
13                string mensaje = malla.EliminarCarrera(id,
14                    txtCarrera.Text);
15                MessageBox.Show(mensaje, "Registro Exitoso");
16                Cargar_DGVMalla();
17            }
18     }
```

**Código 2.5.** Evento al eliminar un registro.



### 3. RESULTADOS Y DISCUSIÓN

En este capítulo se presenta a detalle las pruebas que han sido realizadas en la aplicación y los resultados obtenidos a partir de la culminación de la fase de implementación.

En primer lugar, se llevará a cabo la actualización del tablero Kanban hasta esta fase, luego se desarrollará un entorno de pruebas para verificar los resultados y la validación de los requerimientos funcionales y no funcionales detallados en el **Apartado 2.1.2.** y en el **Apartado 2.1.3.** Una vez realizada las validaciones necesarias, este capítulo finalizará con el cierre del tablero Kanban.

#### 3.1. ACTUALIZACIÓN DEL TABLERO KANBAN

En la Figura 3.1 se puede observar que, las tareas que se encontraban en la etapa “En curso” en la fase de Implementación, han sido culminadas, y por tanto, las actividades que se encontraban en la etapa “Detenido” pasarán a la fase “En curso”; de tal manera que el tablero quede actualizado para poder realizar las pruebas de funcionamiento.



Figura 3.1. Tablero Kanban en fase de Pruebas.

### 3.2. DEFINICIÓN DEL ENTORNO DE PRUEBAS

Para conocer los resultados de la aplicación, se incluye la realización de pruebas necesarias para evaluar el correcto funcionamiento del programa. Este ambiente de pruebas se lo ha realizado mediante la información de índices académicos, y el resultado de la proyección académica correspondiente al semestre 2022-A, recopilados por el Subdecanato de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional.

Con la ayuda de esta información, se ha procedido a utilizar la aplicación con el fin de percibir el correcto funcionamiento de las opciones que brinda el prototipo, y de igual manera, cumplir con las tareas asignadas en los requerimientos, tomando en cuenta los siguientes pasos y consideraciones que se indican en la Tabla 3.1:

**Tabla 3.1** Consideraciones y pasos del entorno de pruebas.

| TAREA                                | CONSIDERACIONES  | PASOS A REALIZAR   |
|--------------------------------------|--|--|
| Ingresar Datos de Índices Académicos | Tener el documento de Excel con las columnas: Id_Materia, Registros, Aprueban, Reprueban, y Reales Matriculados. | <ul style="list-style-type: none"> <li>• Abrir la aplicación.</li> <li>• Seleccionar la opción “Carga de Índices Académicos”.</li> <li>• Abrir el documento de Excel con las consideraciones descritas.</li> <li>• Validar el documento.</li> <li>• Corregir documento en caso de errores.</li> <li>• Insertar periodo académico.</li> <li>• Seleccionar la opción “Subir Documento”.</li> </ul> |
| Calcular Proyección Académica        | Tener los índices académicos ingresados.   | <ul style="list-style-type: none"> <li>• Abrir la aplicación.</li> <li>• Seleccionar la opción “Generar Proyecciones”.</li> <li>• Seleccionar la carrera y el periodo.</li> <li>• Ingresar el periodo de proyección.</li> <li>• Agregar índice de error en caso de ser necesario.</li> <li>• Seleccionar la opción “Generar Proyección”.</li> </ul>  |

|                                  |  |   |
|----------------------------------|--|---|
| Realizar CRUD de datos           | Tener nueva información que se requiera ingresar modificar o eliminar. | <ul style="list-style-type: none"> <li>• Abrir la aplicación.</li> <li>• Seleccionar la opción “Configuración de Datos”.</li> <li>• Seleccionar cualquier opción como: Carrera, Materias, o Índices.</li> <li>• Realizar los cambios necesarios.</li> <li>• Verificar los cambios.</li> </ul>               |
| Obtener documentos de reportería | Tener los índices académicos ingresados.                               | <ul style="list-style-type: none"> <li>• Abrir la aplicación.</li> <li>• Seleccionar la opción “Reportería”.</li> <li>• Seleccionar la carrera y el periodo.</li> <li>• Ingresar el periodo de proyección.</li> <li>• Visualizar o guardar documento PDF.</li> <li>• Exportar documento a Excel.</li> </ul> |

Para el ambiente de la elaboración de pruebas se tuvo en consideración los siguientes parámetros:

- Las pruebas del prototipo se han realizado en la oficina de la coordinación de la carrera de Tecnologías de la Información de la EPN, junto con docentes de la institución.
- Se han llevado a cabo las pruebas utilizando el ordenador en el cual se ha desarrollado la aplicación.
- En el transcurso de la ejecución de pruebas, se instruyó a los participantes el funcionamiento y las actividades que posee la aplicación.
- El personal pudo intervenir cuando lo deseó y tuvo la libertad de realizar cambios intencionales para evaluar el funcionamiento del programa.
- Una vez terminada la prueba, se habrá procedido a cuestionar a los usuarios de manera oral, con el fin de adquirir retroalimentación y evaluar la satisfacción del usuario ante el prototipo.

### 3.3. VALIDACIÓN DE REQUERIMIENTOS

Tomando en cuenta el ámbito de pruebas que se ha descrito en el anterior apartado, se ha obtenido una retroalimentación adecuada la cual permitió realizar la validación de los

requerimientos funcionales y no funcionales descritos en la etapa de Diseño. Estos han sido acogidos satisfactoriamente y han sido corregidos mediante varias sesiones realizadas.

### 3.3.1 VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES

Para validar los requerimientos funcionales se han tomado en cuenta las opiniones y recomendaciones descritas de forma oral de los usuarios presentes en el entorno de pruebas; como evidencia, se presenta la siguiente tabla:

**Tabla 3.2** Validación de Requerimientos Funcionales.

| CÓD. | DESCRIPCIÓN  | ESTADO          |             | OBSERVACIÓN   |
|------|--|-----------------|-------------|---|
| RF01 | Se requiere que la aplicación cuente con una interfaz la cual permita crear, leer, actualizar, y eliminar registros, ya sea materias, carreras o índices académicos.   | Primera reunión | No Validado | Los usuarios pudieron observar el manejo de la opción “Configuración de Datos” de la aplicación, en donde se pudo evidenciar que se puede observar, crear actualizar y eliminar datos correspondientes a materias, carreras o índices académicos. |
|      |  | Segunda reunión | Validado    | Se optó por insertar una opción adicional, para eliminar toda la información de los índices académicos por periodo, y no solo de manera individual.   |
| RF02 | Se requiere que la aplicación produzca documentos, los cuales permitan observar las proyecciones calculadas a partir de los índices académicos en diferentes formatos. | Primera reunión | No Validado | Al momento de utilizar la opción de “Reportería”, los usuarios pudieron obtener dos tipos de archivos en formato <b>.pdf</b> , y <b>.xlsx</b> respectivamente, los cuales reflejan los resultados de las Proyecciones Académicas,                 |
|      |  | Segunda reunión | Validado    | Se modifica y se implementa una presentación agradable al sentido visual de los archivos.   |

|      |   |                 |             |  |
|------|---|-----------------|-------------|--|
| RF03 | Se requiere que la aplicación realice el cálculo de las proyecciones para el siguiente periodo académico. | Primera Reunión | No Validado | Se ha comparado el resultado obtenido con el esperado, basándose en resultados de anteriores periodos. Existieron variaciones en los resultados obtenidos  |
|      |   | Segunda Reunión | Validado    | Los valores obtenidos coinciden con los esperados, tomando en cuenta los índices de errores agregados.   |
| RF04 | Se requiere que la aplicación permita ingresar un índice de error en las estimaciones calculadas.         | Primera Reunión | No Validado | Los usuarios pudieron observar que se despliega una pantalla emergente, que permite agregar un índice de error a la materia seleccionada, sin embargo, no se evidencia un cambio a simple vista. |
|      |   | Segunda Reunión | Validado    | Se modifica, permitiendo que, al guardar el índice de error, este se viera reflejado en la vista principal de las estimaciones de las proyecciones.  |

### 3.3.2 VALIDACIÓN DE REQUERIMIENTOS NO FUNCIONALES

De igual manera, con la retroalimentación obtenida en el entorno de pruebas, se procede a validar los requerimientos no funcionales mediante la Tabla 3.3:

**Tabla 3.3** Validación de Requerimientos No Funcionales.

| CÓD.  | DESCRIPCIÓN   | ESTADO          |             | OBSERVACIÓN   |
|-------|---|-----------------|-------------|---|
| RNF01 | Es necesario contar con una interfaz gráfica que sea amigable, de fácil uso e intuitiva, de manera que el usuario pueda emplearla sin | Primera Reunión | No Validado | Se probó y se mostró la interfaz final del prototipo. Se decidió adecuar textos, botones, colores y espacios en algunas ventanas. |
|       |   | Segunda Reunión | Validado    | En la reunión realizada, los usuarios manifestaron oralmente una conformidad aceptable del  |

|       |  |                 |          |   |
|-------|--|-----------------|----------|---|
|       | enfrentar obstáculos significativos.   |                 |          | bosquejo, y añadieron que la información visual que despliega es clara para su uso.   |
| RNF02 | El sistema debe ser capaz de tener una conexión con una base de datos externa, esta conexión debe incluir el acceso a los datos almacenados y por ende también la consulta, inserción, actualización y eliminación de registros. | Primera Reunión | Validado | Durante la realización de las pruebas, los usuarios pudieron observar que la aplicación está conectada con una base de datos, la cual se encuentra en el mismo computador donde se estaba ejecutando. De igual manera, observaron que en la base reflejaba los cambios al momento de realizar CRUD desde la interfaz gráfica. |
| RNF03 | La aplicación debe ser rápida y responder de manera eficiente. Se pueden establecer requisitos de tiempo de respuesta máximo para asegurar un rendimiento óptimo.  | Primera Reunión | Validado | Se ha evidenciado que el tiempo de respuesta es inmediato al momento de manipular el prototipo, el tiempo de espera más alto se lo obtiene al realizar el cálculo de proyección, el cual da respuesta después de 4.59 segundos aproximadamente, lo cual es considerable debido al procesamiento realizado en esa actividad.   |
| RNF04 | Se requiere que la aplicación sea compatible con el Sistema Operativo Windows debido a la preferencia del usuario que manipulará la aplicación.  | Primera Reunión | Validado | Se muestra satisfacción ante este requerimiento, ya que en el computador donde será utilizada la aplicación maneja el Sistema Operativo Windows.  |

|       |   |                 |             |  |
|-------|---|-----------------|-------------|--|
| RNF05 | El sistema realizará una validación de datos para evitar errores y mantener un formato adecuado de los datos. | Primera Reunión | No Validado | El sistema muestra un mensaje cuando los datos no coinciden con la información esperada. Es importante destacar que en algunas partes falta incrementar las validaciones necesarias. |
|       |   | Segunda Reunión | Validado    | Se mostró satisfacción con respecto a este requerimiento, la validación aplica en todo el programa de manera consistente.  |

### 3.4. CIERRE DEL TABLERO KANBAN

Una vez terminada la fase de pruebas de la aplicación, se da por concluida la actividad que se encontraba en el estado “En Curso”, lo cual hace que la actividad pase a estado “Listo”, y por tal razón ya no se tenga más actividades pendientes en el tablero Kanban, lo cual quedaría de la siguiente manera:



Figura 3.2. Cierre del Tablero Kanban.

## **3.5. CONCLUSIONES Y RECOMENDACIONES**

### **3.5.1. CONCLUSIONES**

El desarrollo del prototipo se llevó a cabo siguiendo cuatro etapas fundamentales: la Fase Teórica, la de Diseño, la de Implementación y la de Pruebas de Funcionamiento. Tras completar este proceso, se obtuvieron las siguientes conclusiones.

- Al desarrollar una aplicación de escritorio para el Subdecanato de la FIEE que permite la proyección de cursos académicos, se proporciona una herramienta eficiente y efectiva que mejora la planificación y organización de la oferta académica, facilitando así la toma de decisiones y la optimización de tiempo de los recursos disponibles.
- El desarrollo de la aplicación de escritorio fue posible gracias al análisis de herramientas y tecnologías como la metodología Kanban, la arquitectura en capas, el lenguaje de programación en C# utilizado en Microsoft Visual Studio y el sistema gestor de base de datos de SQL Server. Estas elecciones permitieron cumplir exitosamente con los objetivos específicos establecidos en el alcance del proyecto.
- Con la ayuda del estudio de la metodología Kanban se planificó las tareas necesarias para el cumplimiento del desarrollo de la aplicación, y de igual manera resultó útil para llevar un cronograma de las demás actividades que se realizaron como parte del proyecto.
- La planificación y el diseño del prototipo resultaron de gran utilidad al proporcionar un orden y una comprensión clara para su implementación. Esto ayudó a establecer la forma adecuada de incorporar todos los módulos y opciones necesarios que controlarán el funcionamiento de la aplicación.
- Con la implementación de la aplicación en base al diseño planteado, se consigue obtener una aplicación con varias interfaces que permiten obtener resultados de Proyecciones Académicas de forma automatizada, lo cual facilita el trabajo al usuario y hace que su tiempo se optimice.
- Implementar una aplicación de escritorio no conlleva solo escribir código, también se debe tomar en cuenta lo que es la planificación, y la documentación, gracias a ello se puede solventar de mejor manera el desarrollo de la aplicación y también tener un respaldo en cuanto al alcance del mismo.
- Tras llevar a cabo la fase de Pruebas de Funcionamiento, se puede concluir que se hizo un excelente uso de las herramientas y tecnologías consultadas durante el



desarrollo de este proyecto de integración curricular. Como resultado, se logró crear una aplicación de escritorio que cumple satisfactoriamente con los requerimientos establecidos por el usuario.

### **3.5.2. RECOMENDACIONES**

Durante la ejecución de este proyecto, se ha enfrentado a una serie de desafíos que requerían una solución inmediata. A raíz de estas situaciones, se identifican algunas recomendaciones clave que consideramos importantes a tomar en cuenta:

- Una recomendación importante es adoptar la práctica de reutilizar código en el desarrollo de software. Al hacerlo, se pueden crear componentes de código más flexibles, modulares y fáciles de aprovechar en diversos contextos. Sin embargo, es fundamental realizar un análisis y adaptación del código reutilizable para que se ajuste a las necesidades específicas de cada proyecto. Esto implica realizar los ajustes necesarios para asegurar su correcto funcionamiento y compatibilidad con el resto del código existente. Al seguir esta recomendación, se maximiza la eficiencia y la calidad del desarrollo de software al evitar duplicaciones innecesarias y optimizar el tiempo y los recursos invertidos.
- En el caso de desarrollar una aplicación de escritorio, es recomendable prestar especial atención al diseño de la interfaz gráfica. Es importante asegurarse de que los usuarios puedan acceder fácilmente a las funciones y características sin enfrentar confusiones innecesarias. Además, es crucial que la interfaz sea adaptable a diferentes resoluciones de pantalla, evitando así distorsiones o cambios drásticos al utilizar una resolución distinta.
- Es recomendable adoptar una mentalidad positiva y enfocada en la investigación al desarrollar aplicaciones. Ante cualquier problema o desafío que surja, es importante estar abierto a explorar diferentes enfoques y soluciones alternativas a las que originalmente se tenían en mente. A través de la investigación exhaustiva, se pueden encontrar nuevos caminos para resolver problemas y adquirir experiencia y conocimiento valiosos en el campo del desarrollo de aplicaciones.
- En el actual proyecto, se ha llevado a cabo el desarrollo de una aplicación de escritorio para automatizar las proyecciones académicas. Sin embargo, en futuros trabajos, se recomienda ampliar la aplicación y convertirla en una aplicación web, no limitándola únicamente al entorno de escritorio. Esto permitirá una mayor accesibilidad, flexibilidad y aprovechamiento de las ventajas que ofrece el entorno en línea.

## 4. REFERENCIAS BIBLIOGRÁFICAS

- [1] <https://www.facebook.com/APDasociacion>, «Metodología Kanban: en qué consiste y cómo utilizarla», *APD España*, 8 de junio de 2021. <https://www.apd.es/metodologia-kanban/> (accedido 28 de julio de 2023).
- [2] <https://www.facebook.com/APDasociacion>, «Metodología Kanban: en qué consiste y cómo utilizarla», *APD España*, 8 de junio de 2021. <https://www.apd.es/metodologia-kanban/> (accedido 28 de julio de 2023).
- [3] «¿Qué es Kanban? Principales características y funciones», *Kanban Software for Agile Project Management*. <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban> (accedido 28 de julio de 2023).
- [4] «¿Qué es la arquitectura de tres niveles? | IBM». <https://www.ibm.com/mx-es/topics/three-tier-architecture> (accedido 28 de julio de 2023).
- [5] Datademia, «¿Qué es SQL?», *Datademia*, 4 de noviembre de 2019. <https://datademia.es/blog/que-es-sql> (accedido 28 de julio de 2023).
- [6] «¿Qué es SQL? - Explicación de lenguaje de consulta estructurado (SQL) - AWS», *Amazon Web Services, Inc.* <https://aws.amazon.com/es/what-is/sql/> (accedido 28 de julio de 2023).
- [7] «¿Qué es Microsoft SQL Server y para qué sirve?», *Intelequia*. <https://intelequia.com/blog/post/que-es-microsoft-sql-server-y-para-que-sirve> (accedido 28 de julio de 2023).
- [8] «¿Qué es Microsoft SQL Server y para qué sirve?» <https://intelequia.com/blog/post/qu%C3%A9-es-microsoft-sql-server-y-para-qu%C3%A9-sirve> (accedido 28 de julio de 2023).
- [9] anandmeg, «Características avanzadas - Visual Studio (Windows)», 19 de junio de 2023. <https://learn.microsoft.com/es-es/visualstudio/ide/advanced-feature-overview?view=vs-2022> (accedido 28 de julio de 2023).
- [10] «Qué es NET Framework», *OpenWebinars.net*, 22 de julio de 2019. <https://openwebinars.net/blog/que-es-net-framework/> (accedido 28 de julio de 2023).
- [11] «C# (C Sharp): Qué es, dónde se utiliza y para qué sirve», *Tribalyte Technologies*, 20 de octubre de 2021. <https://tech.tribalyte.eu/blog-c-sharp-que-es-para-que-sirve> (accedido 28 de julio de 2023).
- [12] «Qué es C#: Introducción», *OpenWebinars.net*, 29 de marzo de 2017. <https://openwebinars.net/blog/que-es-c-introduccion/> (accedido 28 de julio de 2023).
- [13] P. por E. Guille, «Qué es Windows Forms». <https://www.ingenieriasystems.com/2012/10/programacion-plataforma-desktop-vb-net.html> (accedido 28 de julio de 2023).
- [14] adegeo, «What is Windows Forms - Windows Forms .NET», 2 de junio de 2023. <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-7.0> (accedido 28 de julio de 2023).
- [15] T. D. P. Manager, «Mejora tu Flujo de Trabajo: Las 10 Mejores Herramientas Kanban en 2023 (Alternativas a Trello)», *The Digital Project Manager*. <https://thedigitalprojectmanager.com/es/tools/mejores-alternativas-trello-mejores-herramientas-kanban/> (accedido 28 de julio de 2023).
- [16] A. M. Zambrano, «Ingeniería de Requerimientos», Quito.
- [17] M. Arias Chaves, «La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software», n.º 2215-2458, pp. 1-13.
- [18] A. Yaseen, «Resumen de la estructura de la tabla de SQL Server», *SQL Shack - articles about database auditing, server performance, data recovery, and more*, 2 de agosto de 2018. <https://www.sqlshack.com/es/resumen-de-la-estructura-de-la-tabla-de-sql-server/> (accedido 28 de julio de 2023).
- [19] «Diagrama de Clases», *manuel.cillero.es*. <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/diagrama-de-clases/> (accedido 28 de julio de 2023).

- [20] «3c-relaciones.pdf». Accedido: 28 de julio de 2023. [En línea]. Disponible en: <http://elvex.ugr.es/decsai/java/pdf/3c-relaciones.pdf>
- [21] «▷ Diagrama de actividades», *DiagramasUML.com*. <https://diagramasuml.com/actividades/> (accedido 28 de julio de 2023).
- [22] «Capa de Presentación». <http://itcelenes.mx.tripod.com/unidad9Capadepresentacion.html> (accedido 28 de julio de 2023).

## **5. ANEXOS**

A continuación, se presentan los apéndices que han sido anexados como complemento al desarrollo de este Trabajo de Integración Curricular.

**5.1. ANEXO I:** [CÓDIGO DE LA APLICACIÓN](#)

**5.2. ANEXO II:** [SCRIPT DE LA BASE DE DATOS](#)

**5.3. ANEXO III:** [MANUAL DE USUARIO](#)

**5.4. ANEXO IV:** [PLANTILLA DE INGRESO DE ÍNDICES ACADÉMICOS](#)