

# **ESCUELA POLITÉCNICA NACIONAL**

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

### **APLICACIONES DE SOFTWARE EDUCATIVO**

**DESARROLLO DE UN PROTOTIPO SOFTWARE PARA QUE UN  
PROFESOR CON DISCAPACIDAD VISUAL PUEDA GESTIONAR  
LAS CALIFICACIONES DE SUS ALUMNOS MEDIANTE HOJAS DE  
CÁLCULO E INTERACTUAR CON LAS MISMAS**

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

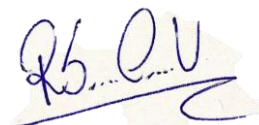
**RICARDO SEBASTIÁN CORRALES VITERI**  
**sebastiancv\_18@hotmail.com**

**DIRECTOR: DRA. ANA MARÍA ZAMBRANO VIZUETE**  
**ana.zambrano@epn.edu.ec**

**DMQ, Octubre 2022**

## CERTIFICACIONES

Yo, Ricardo Sebastián Corrales Viteri declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



---

**Ricardo Sebastián Corrales Viteri**

Certifico que el presente trabajo de integración curricular fue desarrollado por Ricardo Sebastián Corrales Viteri, bajo mi supervisión.



---

**Dra. Ana María Zambrano Vizuete**

**DIRECTOR**

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

---

SR. RICARDO SEBASTIÁN CORRALES VITERI

---

DRA. ANA MARÍA ZAMBRANO VIZUETE

## DEDICATORIA

A mis padres Sandra y Ricardo, en especial dedico con todo mi corazón este trabajo a mi madre, pues sin ella no lo habría logrado. Sus enseñanzas y apoyo incondicional me han llevado a ser una mejor persona. A mi padre, por su entrega, sacrificio y amor por nuestra familia.

A mis hermanas Andrea y Estefanía, por haber formado parte de esta valiosa etapa y ofrecerme su ayuda siempre que la he necesitado.

A mis amigos que han sido parte integral de mi desenvolvimiento personal y profesional. De manera especial a Andrés y Antonio por su apoyo en esta travesía.

***Ricardo Sebastián Corrales Viteri***

## **AGRADECIMIENTO**

Brindo un agradecimiento especial a la Dra. Ana María Zambrano, directora de este trabajo, por darme las directrices correctas, brindarme su apoyo y permitir que este trabajo se terminara a buen recaudo.

Agradezco a Dios y a mi familia, quienes siempre han creído en mí, dándome ejemplo de superación, humildad y sacrificio; enseñándome a valorar todo lo que tengo.

***Ricardo Sebastián Corrales Viteri***

# ÍNDICE DE CONTENIDOS

CERTIFICACIONES .....	I
DECLARACIÓN DE AUTORÍA .....	II
DEDICATORIA .....	III
AGRADECIMIENTO .....	IV
ÍNDICE DE CONTENIDOS .....	V
RESUMEN .....	VII
ABSTRACT .....	VIII
1 INTRODUCCIÓN.....	9
1.1 OBJETIVO GENERAL .....	10
1.2 OBJETIVOS ESPECÍFICOS .....	10
1.3 ALCANCE .....	10
1.4 MARCO TEÓRICO .....	13
1.4.1 ESTADO DEL ARTE EN SOFTWARE DE INCLUSIÓN PARA PERSONAS CON DISCAPACIDAD VISUAL .....	13
1.4.2 METODOLOGÍA KLM GOMS .....	14
1.4.3 METODOLOGÍA KANBAN.....	15
1.4.4 TECNOLOGÍAS Y LENGUAJES PARA IMPLEMENTACIÓN DE PROTOTIPO .....	15
2 METODOLOGÍA.....	17
2.1 DISEÑO .....	17
2.1.1 PLANTEAMIENTO DEL TABLERO KANBAN.....	17
2.1.2 REQUERIMIENTOS FUNCIONALES (RF) .....	18
2.1.3 REQUERIMIENTOS NO FUNCIONALES (RNF) .....	21
2.1.4 DEFINICIÓN DE CASOS DE USO .....	23
2.1.5 DISEÑO DE LA CAPA DE DATOS.....	24
2.1.6 DISEÑO DEL DIAGRAMA DE CLASES.....	25
2.1.7 DISEÑO DE LA CAPA DE PRESENTACIÓN .....	28
2.2 IMPLEMENTACIÓN.....	32
2.2.1 ACTUALIZACIÓN DEL TABLERO KANBAN.....	32
2.2.2 IMPLEMENTACIÓN DE LA CAPA DE DATOS.....	33
2.2.3 IMPLEMENTACIÓN DE LA CAPA DE NEGOCIO .....	35
2.2.4 IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN .....	36
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES .....	40

3.1	RESULTADOS.....	41
3.1.1	ACTUALIZACIÓN DEL TABLERO KANBAN.....	41
3.1.2	DEFINICIÓN DEL ENTORNO DE PRUEBAS.....	41
3.1.3	ESTADÍSTICAS DEL USO DEL PROTOTIPO CON LA METODOLOGÍA KLM GOMS .....	42
3.1.4	VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES .....	44
3.1.5	VALIDACIÓN DE REQUERIMIENTOS NO FUNCIONALES.....	46
3.1.6	CIERRE DEL TABLERO KANBAN.....	48
3.2	CONCLUSIONES Y RECOMENDACIONES.....	48
3.2.1	CONCLUSIONES.....	48
3.2.2	RECOMENDACIONES .....	49
3.2.3	TRABAJO FUTURO .....	50
4	REFERENCIAS BIBLIOGRÁFICA.....	51
5	ANEXOS .....	53
	ANEXO I. Glosario .....	54
	ANEXO II. Formato de la lista de estudiantes .....	55
	ANEXO III. Código del prototipo.....	56

## RESUMEN

El presente Trabajo de Integración plantea un prototipo Gestor de Calificaciones en el cual se puede simular una hoja de cálculo delimitada para la gestión de calificaciones de alumnos tutorados por un profesor con discapacidad visual, guiado en todo momento por traductores de texto a voz TTS (*Text to Speech*); además se puede obtener estadísticas (promedio, nota máxima y nota mínima) de la materia, con el fin de facilitar el manejo de la información de sus alumnos.

Durante el primer capítulo se explorarán los pormenores del Estado de Arte en Software de Inclusión para personas con discapacidad visual. También se detallará la metodología que se utiliza a lo largo del presente trabajo. La primera metodología es Kanban que busca tener un proceso eficiente y la metodología KLM GOMS que se utilizará para comprobar el funcionamiento adecuado del prototipo, esto gracias a tiempos definidos para realizar ciertas tareas. Por último, se explicarán las tecnologías, los diferentes lenguajes para la implementación del prototipo y la arquitectura en capas que se utilizará en el desarrollo del prototipo.

A lo largo del segundo capítulo se definen los patrones de diseño en las Capas de Datos, Negocio y Presentación para alcanzar un prototipo adaptado correctamente a usuarios con discapacidad visual, procurando seguir todos los requerimientos funcionales y no funcionales. Se detallan a su vez los pasos a seguir para implementar el prototipo y las funcionalidades que fueron desarrolladas.

Finalmente, se pone a prueba el prototipo desarrollado en un entorno de pruebas controlado. Se verifica el desenvolvimiento del prototipo en un entorno de pruebas controladas mediante el uso del prototipo por usuarios que simular ser individuos no videntes y se realizan las correcciones obtenidas mediante la retroalimentación de los usuarios de prueba

**PALABRAS CLAVE:** TTS, Text to Speech, Microsoft Visual Studio, Microsoft SQL Server, Kanban, KLM GOMS.



## ABSTRACT

This Curricular Integration Paper proposes a Grade Manager prototype in which a delimited spreadsheet can be simulated for managing grades of students tutored by a visually impaired teacher, always guided by TTS (Text to Speech) translators; In addition, you can obtain statistics (average, maximum grade, and minimum grade) of the subject, to facilitate the management of his students' information.

During the first chapter, the details of the State of the Art in Inclusion Software for people with visual disabilities will be explored. The methodology used throughout this work will also be detailed. The first methodology is Kanban that seeks to have an efficient process and the KLM GOMS methodology that will be used to verify the proper functioning of the prototype, thanks to defined times to perform certain tasks. Finally, the technologies, the different languages for the implementation of the prototype and the layered architecture that will be used in the development of the prototype will be explained.

Throughout the second chapter, the design patterns are defined in the Data, Business and Presentation Layers to achieve a prototype correctly adapted to users with visual disabilities, trying to follow all the Functional and Non-Functional Requirements. The steps to follow to implement the prototype and the functionalities that were developed are detailed.

Finally, the developed prototype is tested in a controlled test environment. The development of the prototype is verified in a controlled test environment through the use of the prototype by users who pretend to be blind individuals and the corrections obtained are made through the feedback of the test users.

**KEYWORDS:** TTS, Text to Speech, Microsoft Visual Studio, Microsoft SQL Server, Kanban, KLM GOMS.

# 1 INTRODUCCIÓN

En los últimos años el uso de programas basados en hojas de cálculo ha sido muy importante, pues gracias a estos, se puede llevar una gestión eficiente de la información de manera ordenada y sencilla. Sin embargo, las herramientas actuales que permiten la gestión de hojas de cálculo no están orientadas a personas con discapacidad visual, a pesar de ser una parte numerosa de la población. Esto ha sido una de las claves importantes para el desarrollo de una gran variedad de aplicaciones, pero muchas de estas no son inclusivas para personas con algún tipo de discapacidad o son basadas exclusivamente en lectores de pantalla externos que guían a la persona no vidente hasta cierto punto dejando algunos vacíos al momento de su uso. Por este motivo se propone desarrollar un programa de escritorio simulando una hoja de cálculo que permita a un profesor con discapacidad visual la gestión de calificaciones de alumnos tutorados por el mismo.

Para realizar este programa de escritorio se hará uso de la herramienta de Microsoft Visual Studio, ya que en este entorno mediante traductores de texto a voz se podrá guiar a la persona en todo momento para que su discapacidad no la limite de ninguna manera. Este módulo será programado en lenguaje C# y la sonorización TTS (Text to Speech) se lo realizará mediante una librería disponible en la misma aplicación. Además, el almacenamiento de la información se realizará mediante una base de datos en Microsoft SQL Server.

Crear un entorno en el cual se pueda almacenar, manejar e interpretar los datos de manera intuitiva y sencilla es posible para una persona invidente, esto gracias a que la persona podrá ingresar el número exacto de calificaciones para el semestre en curso, así como el número de alumnos, simulando una hoja de cálculo limitada donde podrá navegar conociendo a cada paso en donde se encuentra ubicado y qué acciones o no puede realizar. Además, le permitirá obtener las calificaciones finales de manera automática, así como obtener estadísticas de las calificaciones; también podrá descargar en formato de archivo .xlsx, .csv y PDF y así, tendrá la información para una interpretación posterior. Esto facilitará la gestión de calificaciones para la persona no vidente.

Al no existir una aplicación de escritorio que solviera el problema de una persona no vidente de interactuar con una hoja de cálculo en la gestión de calificaciones de sus alumnos, se desarrollará un prototipo sencillo e intuitivo que simulará una hoja de cálculo con guías auditivas que orientará al usuario en todo momento.

## **1.1 OBJETIVO GENERAL**

Implementar un prototipo software para que un profesor con discapacidad visual pueda gestionar las calificaciones de sus alumnos mediante la simulación de hojas de cálculo e interactuar con las mismas.

## **1.2 OBJETIVOS ESPECÍFICOS**

1. Analizar las herramientas necesarias y la metodología que se utilizarán en el desarrollo del prototipo.
2. Diseñar el módulo del prototipo.
3. Implementar el prototipo de acuerdo al diseño realizado.
4. Validar los requerimientos funcionales y no funcionales del Prototipo.
5. Analizar los resultados obtenidos en base a distintas métricas generadas mediante la metodología KLM GOMS con la ayuda de personas invidentes.

## **1.3 ALCANCE**

El presente Trabajo de Integración Curricular propone el desarrollo de un prototipo inclusivo simulando una hoja de cálculo para la gestión de calificaciones de un grupo de estudiantes tutorados por una persona invidente. El mismo podrá realizar cálculos matemáticos elementales, lo cual permitirá realizar una gestión sencilla y eficiente de las calificaciones de los estudiantes.

Para manipular el prototipo, la persona no vidente podrá crear materias, cargar estudiantes, ingresar notas y obtener estadísticas; todo esto a partir de atajos del teclado para facilitar la gestión de tareas. El prototipo permitirá calcular de manera automática las calificaciones del primer y segundo parcial, guardar la información y guiar en todo momento a la persona no vidente por un traductor de voz, que indique las opciones posibles de trabajo en la estructura de una hoja de cálculo y las acciones realizadas por el usuario.

El funcionamiento del Gestor de Calificaciones funciona de la siguiente manera:

- Se definirá únicamente un actor que será el Profesor no vidente, el cual accederá a este prototipo mediante una interfaz de escritorio.

- Existe una base de datos, que permitirá el almacenamiento de Materias, Alumnos (cargados desde el sistema) y las rúbricas de evaluación y su ponderación correspondiente.
- La base de datos guardará la información de las notas (1er Bimestre, 2do Bimestre y Examen Final).
- Las notas correspondientes a Alumnos vs. Rúbrica de evaluación se guardarán únicamente en el archivo Excel manejado
- La Rúbrica de evaluación corresponde a las diferentes calificaciones con las que el profesor evaluará al estudiante durante el semestre; cada una de las Rúbricas tiene su correspondiente ponderación.

El prototipo presenta un único módulo, con el siguiente funcionamiento.

1. El usuario no vidente crea una materia y las rubricas que tendrá la misma.
2. Una vez que existe al menos una materia, el usuario no vidente podrá cargar los estudiantes en dicha materia exportando la lista de estudiantes en un formato dado (Formato obtenido del SII-Académico **Anexo II**).
3. Una vez cargados los estudiantes en la materia se simulará una Hoja de Cálculo delimitada en donde el usuario no vidente, con ayuda de un traductor de voz, podrá colocar todas las notas de los alumnos.
4. Finalmente, el prototipo calculará las notas del primer y segundo parcial automáticamente.
5. Además de calcular las estadísticas de las calificaciones (promedio, nota máxima, nota mínima) con la opción de descargar una hoja de cálculo en formato .pdf, .xlsx, .csv.

### **A. Fase Teórica**

Se estudiará el Estado del Arte de software para personas no videntes en el desarrollo de aplicaciones específicas para este grupo de personas [1]. Se resumirá la arquitectura en 3 capas: Capa de Datos, Capa Lógica de Negocio y Capa de Presentación [2]; esta última capa tendrá una gran importancia en el desarrollo del prototipo. Finalmente, se resumirá la metodología Kanban y la metodología KLM GOMS [3] para realizar la recolección de estadísticas sobre atajos del teclado y tareas que sean usadas en el prototipo.

## **B. Fase de Diseño**

Basándose en la metodología Kanban [4], se organizarán tareas y adecuarán tiempos para la resolución de las mismas. Siguiendo, se obtendrán los requerimientos funcionales y no funcionales del prototipo.

Para lograr cumplir con los requerimientos del prototipo se realizarán diagramas que faciliten la comprensión de las tareas a realizar, como lo son: Diagrama Relacional, Diagrama de Clases, Diagrama de Actividades [5].

## **C. Fase de Implementación**

En base a las tareas definidas, y guiándose en el método de trabajo de Kanban se procede con la instalación de las herramientas necesarias para proceder a programar cada capa, siempre respetando los tiempos y el orden predefinido

Acorde al diagrama Relacional obtenido en la Fase de Diseño se procederá a programar la Base de Datos en SQL Server [6].

Para la programación de las tareas del prototipo correspondientes a la Capa Lógica de Negocio se hace uso del Lenguaje C# [7] para poder codificar la respectiva Capa en base a los Diagramas anteriormente realizados.

Con la programación de cada capa se creará a la par, la guía por voz de los pasos para el uso del prototipo, misma que se incluirá dentro de la Capa de Presentación.

Se procede a codificar la Capa de Presentación haciendo uso de Windows Forms [8]. Finalmente, se procede con la integración del prototipo para poder proseguir con la fase de pruebas.

## **D. Fase de Pruebas de Funcionamiento**

Una vez finalizada la lista de tareas del Diseño e Implementación del tablero Kanban, se procederá a realizar pruebas simulando la deficiencia visual en un ambiente de pruebas controlado [9]. En el caso que se requiera, se realizará las correcciones necesarias a cualquier capa del prototipo para cumplir a cabalidad los objetivos planteados. Una vez realizado esto, se procede a obtener estadísticas en base a las métricas definidas en la metodología KLM GOMS [10]. Finalmente se actualizará la información provista en el tablero Kanban para dar por culminada la realización del prototipo incluso simulando una estructura de una hoja de cálculo.

## 1.4 MARCO TEÓRICO

En el **Apartado 1.4.1**, se analizará el Estado del Arte en Software de Inclusión para Personas con Discapacidad Visual. En el **Apartado 1.4.2** se estudiará la Metodología KLM GOMS. Siguiendo, en el **Apartado 1.4.3** la Metodología Kanban y por último, en el **Apartado 1.4.4**, las Tecnologías y Lenguajes para la Implementación de Prototipo.

### 1.4.1 ESTADO DEL ARTE EN SOFTWARE DE INCLUSIÓN PARA PERSONAS CON DISCAPACIDAD VISUAL

La falta de sistemas informáticos de inclusión para personas con discapacidad visual es una forma de discriminación. En los últimos años, gracias a los avances tecnológicos se ha incrementado el desarrollo de productos de software, pero estos no han sido desarrollados para la inclusión de personas con algún tipo de discapacidad.

Según la referencia [11], las personas con discapacidad son un segmento social muy numeroso siendo alrededor del 10% de la población mundial, cuya inclusión a todas las áreas de la sociedad sigue siendo insuficiente. A pesar de los avances tecnológicos todavía las aplicaciones no son accesibles para personas con algún tipo de discapacidad. Un aspecto muy importante al momento de desarrollar software de Inclusión es tener en cuenta que la persona no vidente puede obtener mucha información a través de ayudas de texto a voz, siendo el sentido del oído uno de los medios más importante para recibir información. En la referencia [12], se desarrolla un Sistema Interactivo para Personas con Discapacidad Visual. Esta aplicación es un juego de memoria para personas ciegas con el objetivo de dar a conocer lugares turísticos y mejorar habilidades de memoria. Ha sido desarrollado considerando aspectos como: tamaño de botones, sonido, ergonomía, guías de voz entre otras. Se concluyó que la aplicación fue amigable con el usuario y fácil de utilizar.

Se ofrece un estudio acerca de las pautas imprescindibles para el diseño y creación de sitios web accesibles para personas ciegas [13]. En este trabajo se destacan algunos aspectos de accesibilidad mínimos que deberían tener una aplicación como usar texto alternativo para describir las funciones de ciertos elementos visuales, organización del programa estructurando bien la aplicación para que la persona con discapacidad visual no tenga la dificultad de saber en qué parte de la aplicación se encuentra. Los lectores de pantalla y atajos de teclado pueden ser de mucha importancia en el diseño de la aplicación permitiendo al usuario no vidente ser guiado en todo momento y tener una interfaz amigable y fácil de utilizar.

## 1.4.2 METODOLOGÍA KLM GOMS

El modelo predictivo es un modelo práctico en el que se da un enfoque ingenieril al proceso de diseño de la interfaz de usuario; esto a través de técnicas formales en las cuales se tratará de predecir la eficiencia entre el humano y el ordenador.

El diseñador de la aplicación deberá realizar tareas de modelado considerando posibles diseños, modelar las tareas de estos y elegir el mejor diseño en base a las predicciones y estimaciones de los tiempos para ejecutar cierta tarea.

Entre los modelos de diseño y evaluación de interfaces de usuario el más extendido es el modelo GOMS (Goals, Operatos, Methods and Selection Rules) propuesto por primera vez por Moran, Card y Newell [14].

Un modelo GOMS consiste en descripciones de los métodos necesarios para ejecutar tareas especificadas pudiendo ser explotadas de diferentes formas a nivel de evaluación de tiempos, estimación de complejidad en el uso del sistema o requisitos de diseño. La forma más simple del modelo GOMS es el modelo KLM (Keystroke-Level Model) [15] en el cual el tiempo de ejecución de una tarea se predice por el total de veces para las acciones elementales de nivel de teclas requeridas para realizar la tarea.

El modelo GOMS se compone de métodos que se utilizan para lograr objetivos específicos, estos están compuestos por:

- **Goals:** Se refiere a lo que el usuario está intentando conseguir, por ejemplo, acceder a una ventana de la interfaz, este objetivo se puede dividir en objetivos secundarios; todos estos deben cumplirse para alcanzar el objetivo general.
- **Operators:** Son un conjunto de operaciones atómicas con las que se compone la solución deseada. Los operadores pueden ser actos perceptuales, cognitivos, motores o una combinación de estos y que pueden afectar al sistema como por ejemplo presionar una tecla del teclado o mental como leer un cuadro de texto.
- **Methods:** Son secuencias de los operadores agrupados para cumplir un objetivo simple, teniendo la posibilidad de tener distintos caminos para conseguir el mismo objetivo. Por ejemplo, para abrir una ventana se lo puede hacer mediante el botón que diga Abrir (Método-Botón), pulsar una combinación de teclas (Método-Comando) o seleccionar un elemento del menú (Método-Menú).
- **Selection Rules:** Permitirán escoger el método a utilizar si existieran más de un método aplicable a un objetivo, entonces las reglas de selección son imprescindibles para representar el conocimiento del usuario sobre que método aplicar.

KLM es una versión simplificada de GOMS que a pesar de su relativa simplicidad ha demostrado ser exacto [16]. Su propósito es sencillo, modelar el tiempo que tarda un usuario en realizar una tarea con un determinado método en un sistema o aplicativo.

### 1.4.3 METODOLOGÍA KANBAN

Kanban es una palabra japonesa que significa letrero y utilizada en producción significa “que, cuando, donde y como producir” [17]. La metodología Kanban busca obtener un productivo, organizado y eficiente proceso. Esta técnica fue creada por Toyota para controlar el progreso de trabajo a través de una cadena de suministro [18].

Kanban es parte de una metodología de manufactura la cual es basada en la técnica JIT (Just in Time) que significa hacer solo lo que es necesario; además se la utiliza en muchas compañías de producción.

El principal objetivo de la metodología es asegurar una tasa de producción sostenible para así evitar excesos de productos terminados, cuellos de botella y retrasos en la entrega; entonces el trabajo debe ser organizado de acuerdo con la capacidad de los centros de trabajo y equipo.

Esta metodología consiste en un tablero separado en 3 tipos de trabajo, “Tareas por Completar”, “Tareas en Progreso” y “Tareas Completadas”. Así se mantiene un flujo de trabajo continuo y una visualización del trabajo generado.

### 1.4.4 TECNOLOGÍAS Y LENGUAJES PARA IMPLEMENTACIÓN DE PROTOTIPO

#### 1.4.4.1 Arquitectura en Capas

Se ocupará una arquitectura de 3 capas. La **Capa Física** es la encargada de realizar transacciones con la base de datos y con otros sistemas para colocar o extraer información. El manejo de los datos debe realizarse de una manera que garantice la consistencia de los datos debido que, al momento de extraer información de esta base de datos, la información debe ser precisa y consistente. Esta capa se comunica directamente con la **Capa Lógica de Negocio** para que sea procesada e ingresada en objetos que se necesite [19]. En la **Capa de Negocio** se definen todas las reglas que se deben cumplir para una correcta ejecución del programa y es el vínculo entre la **Capa de Datos** y la **Capa de Presentación**. La **Capa de Presentación** es el aplicativo ante el usuario, aquí se maneja las interfaces



que cumplen el objetivo principal del programa, el cual es facilitar la interacción con la aplicación; para esto la interfaz debe ser amigable con el usuario y fácil de utilizar[20]. Esta capa se comunica únicamente con la Capa Lógica de Negocio.

#### **1.4.4.2 Lenguaje y Entornos de Desarrollo por Capas**

Para el diseño de la Base de Datos se utilizará la herramienta Microsoft SQL Server [21] , el cual es un sistema de gestión de bases de datos relacionales producido por Microsoft que contiene características y herramientas que se pueden utilizar para desarrollar y administrar bases de datos; su principal lenguaje de consultas es Transact-SQL. El entorno integrado SQL Server Management Studio sirve para para acceder, configurar y administrar todos los componentes de SQL Server.

Para la Aplicación de Escritorio se usará Visual Studio [22]; el cual es un IDE (*Entorno de Desarrollo Integrado*) desarrollado por Microsoft. Este IDE es un conjunto de herramientas que permite desarrollar aplicaciones de escritorio, aplicaciones móviles, servicios WEB entre otros. Se utilizó .NET Framework [23] que proporciona Microsoft para una programación más sencilla aumentando la seguridad de los programas y disminuyendo las vulnerabilidades; el principal componente de este framework es CLR (*Common Language Runtime*). Visual Studio acepta los siguientes lenguajes de programación: C#, C++, Java, Python, PHP entre otros. El lenguaje C# [24] es el que se utilizará en el desarrollo del prototipo en donde se usará Programación Orientada a Objetos [25].

Para modelar la base de datos se utilizará EF (*Entity Framework*) [26] en la Capa de Datos; EF es el ORM (*Object Relational Mapping*) oficial de Microsoft, el cual elimina la necesidad de escribir la mayoría del código de acceso a datos; es decir EF permite acceder a una base de datos utilizando clases que representan cada una de las entidades de ésta, dando la posibilidad de realizar cualquier operación sobre los datos simplemente llamando a métodos de estas clases.

Para las Guías de Voz en la Capa de Presentación se utilizará la librería *System.Speech* [27] la cual contiene clases para inicializar, configurar un motor de síntesis de voz, crear mensajes, generar voz y responder a eventos; permitiendo crear un flujo de instrucciones por voz ordenada y lógica.

## 2 METODOLOGÍA

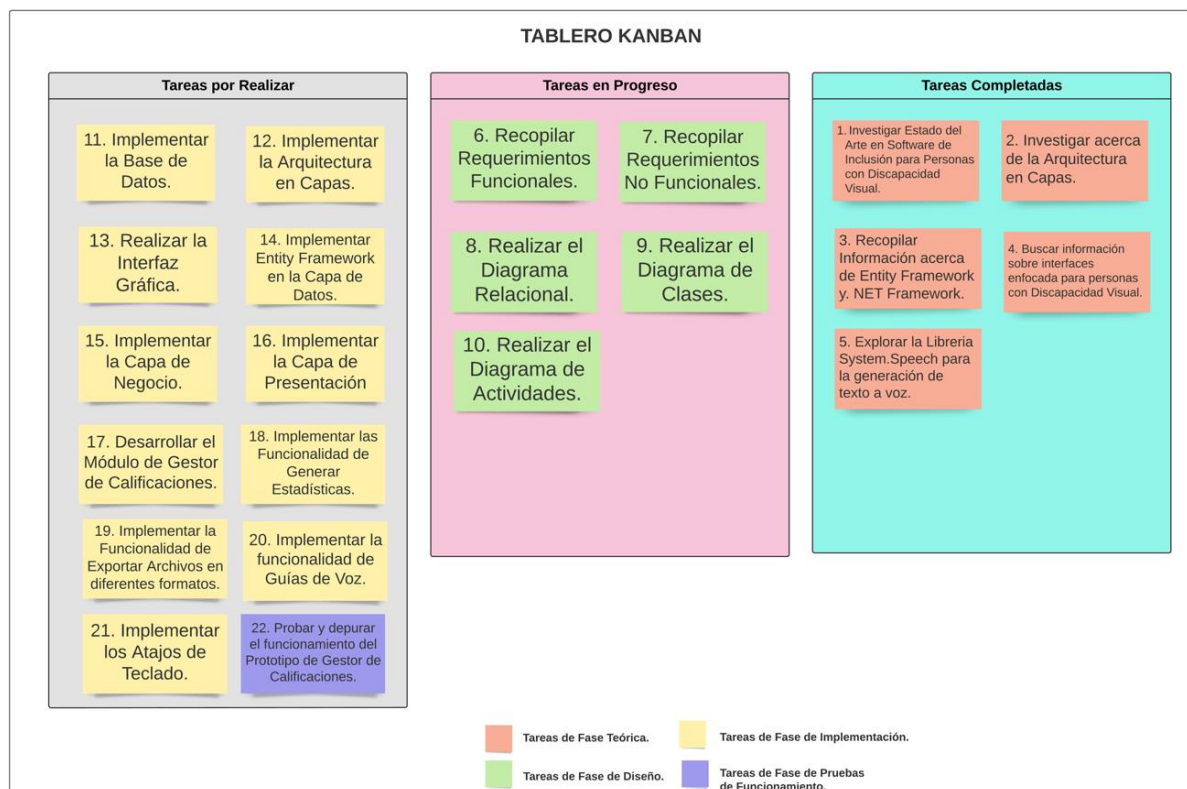
En el presente Capítulo se mostrarán los pormenores en la construcción de la Aplicación de Escritorio: las decisiones de diseño tomadas y los diferentes detalles de desarrollo realizadas.

A lo largo del **Apartado 2.1**, correspondiente al Diseño, se definirán las principales pautas para el desarrollo de la Aplicación de Escritorio, por otra parte; en el **Apartado 2.2**, Implementación, se explorarán las decisiones de desarrollo tomadas.

### 2.1 DISEÑO

En este Apartado se podrán observar las pautas tomadas para desarrollar la Aplicación de Escritorio; así como, los requerimientos Funcionales y No Funcionales del Prototipo y los diferentes Diseños de la Arquitectura en Capas. Finalmente, el diseño del Módulo de Gestión de Calificaciones será propuesto.

#### 2.1.1 PLANTEAMIENTO DEL TABLERO KANBAN



**Figura 2.1.** Planteamiento del Tablero Kanban en Etapa de Diseño.

Se utilizó la herramienta LucidSpark [28] para realizar el Tablero Kanban y actualizar los resultados a medida que se vayan consiguiendo los objetivos planteados. Según se puede observar en la Figura 2.1, durante la Fase de Diseño se deben realizar cinco Tareas, cuya finalización permitirá el correcto funcionamiento del Prototipo de Gestor de Calificaciones.

### 2.1.2 REQUERIMIENTOS FUNCIONALES (RF)

Los Requerimientos Funcionales son funcionalidades que debe cumplir el prototipo con el fin de cumplir todos los objetivos planteados [29]. Estos requerimientos se detallan en las siguientes Historias de Usuario.

RF 01	<b>Creación de una Materia</b>
<b>Usuario:</b>	Profesor No Vidente
<b>Prioridad:</b>	Alta
<b>Descripción:</b>	
Se requiere crear una nueva Materia en el Prototipo.	
<b>Criterios de Aceptación:</b>	
<ul style="list-style-type: none"> <li>• El Prototipo aceptará 4 campos (Nombre de la Materia, Número de Estudiantes, Número de Rúbricas del Primer Bimestre y Número de Rubricas del Segundo Bimestre).</li> <li>• En el campo Nombre de la Materia puede ingresar letras y números.</li> <li>• En el campo Número de Estudiantes solo se permite el ingreso de números y el rango permitido es de 0 a 50 estudiantes.</li> <li>• En el campo de Número de Rubricas para el Primer y Segundo Bimestre solo se permite el ingreso de números y el rango es de 1 a 5 Rúbricas por Bimestre.</li> <li>• El prototipo no permitirá el ingreso de una materia que ya exista.</li> <li>• El prototipo validará que, para crear la materia, todos los campos estén completos.</li> </ul>	

RF 02	<b>Cargar Estudiantes en la Materia Creada</b>
<b>Usuario:</b>	Profesor No Vidente
<b>Prioridad:</b>	Alta
<b>Descripción</b>	
Se requiere subir el archivo de los estudiantes en la Materia creada.	
<b>Criterios de Aceptación</b>	
<ul style="list-style-type: none"> <li>• El Prototipo aceptará el archivo en formato .xlsx caso contrario no permitirá ningún otro tipo de archivo.</li> <li>• El archivo permitido será obtenido del SII-Académico.</li> <li>• Para subir Estudiantes se debe primero crear la materia, después asignar las rúbricas correspondientes y finalmente se podrá subir el archivo de Estudiantes.</li> </ul>	

RF 03	<b>Creación Dinámica de Rúbricas</b>
<b>Usuario:</b>	Profesor No Vidente
<b>Prioridad:</b>	Alta
<b>Descripción</b>	
Se requiere crear dinámicamente las Rubricas del Primer y Segundo Bimestre.	
<b>Criterios de Aceptación</b>	
<ul style="list-style-type: none"> <li>• El Prototipo aceptará 2 campos (Nombre de la Rúbrica y Ponderación).</li> <li>• El Prototipo creará dinámicamente las rúbricas del 1er y 2do bimestre de acuerdo con la información ingresada al momento de crear la Materia.</li> <li>• En el Campo Nombre de la Rúbrica se permitirá el Ingreso de números y letras.</li> <li>• En el campo Ponderación se permitirá el ingreso de números en el rango de 0.1 a 10, con el punto como separador de decimales.</li> <li>• El prototipo validará que todos los campos estén completos para crear la Rúbrica.</li> <li>• Primero se debe crear la Materia para después crear dinámicamente las Rúbricas.</li> <li>• El Prototipo asignará de manera automática nombres generales de las Rúbricas creadas que se podrá cambiar de acuerdo con el criterio del Profesor No Vidente.</li> <li>• El Prototipo asignará automáticamente valores a las ponderaciones que posterior se podrá cambiar de acuerdo con el criterio del Profesor No Vidente.</li> </ul>	

RF 04	<b>Simular Hoja de Cálculo</b>
<b>Usuario:</b>	Profesor No Vidente
<b>Prioridad:</b>	Alta
<b>Descripción</b>	
Se requiere simular una Hoja de Cálculo para ingresar las calificaciones de los estudiantes.	
<b>Criterios de Aceptación</b>	
<ul style="list-style-type: none"> <li>• El Prototipo permitirá simular una Hoja de Cálculo siempre y cuando exista al menos una materia ya creada.</li> <li>• El Prototipo permitirá simular la Hoja de Cálculo cuando se haya seleccionado una materia.</li> <li>• El Prototipo permitirá la simulación de una Hoja de Cálculo cuando la materia se haya creado correctamente.</li> <li>• No se permitirá simular la Hoja de Cálculo cuando no se hayan asignado sus Rúbricas.</li> <li>• No se permitirá simular la Hoja de Cálculo cuando la suma de las ponderaciones de las Rúbricas de los bimestres no sea igual a 10.</li> <li>• No se permitirá simular la Hoja de Cálculo cuando no se haya subido ningún archivo de Estudiantes.</li> <li>• No se permitirá simular la Hoja de Cálculo cuando se haya subido un archivo incorrecto.</li> <li>• El prototipo informará cuando una materia no haya sido ingresada correctamente dando al usuario la opción de eliminar la materia.</li> </ul>	

RF 05	<b>Eliminar Materia</b>
<b>Usuario:</b>	Profesor No Vidente
<b>Prioridad:</b>	Media
<b>Descripción</b>	
Se requiere poder eliminar alguna materia.	
<b>Criterios de Aceptación</b>	
<ul style="list-style-type: none"> <li>• El Prototipo permitirá eliminar cualquier materia.</li> <li>• Una vez eliminada la materia se borrará la información de sus Rubricas, sus alumnos y Calificaciones Finales.</li> <li>• El Prototipo permitirá Eliminar Materias siempre que haya al menos una materia ya creada.</li> <li>• EL Prototipo no permitirá recuperar ninguna materia eliminada.</li> </ul>	

RF 06	<b>Editar ciertos atributos de la Materia</b>
<b>Usuario:</b>	Profesor No Vidente
<b>Prioridad:</b>	Media
<b>Descripción</b>	
Se requiere poder editar ciertos campos.	
<b>Criterios de Aceptación</b>	
<ul style="list-style-type: none"> <li>• El Prototipo permitirá editar cualquier materia que se haya creado correctamente.</li> <li>• El Prototipo permitirá Editar Materias siempre que haya al menos una materia ya creada.</li> <li>• El Prototipo solo permitirá editar el nombre de la materia, el nombre de las rúbricas y sus respectivas ponderaciones.</li> </ul>	

RF 07	<b>Computar Calificaciones</b>
<b>Usuario:</b>	Profesor No Vidente
<b>Prioridad:</b>	Alta
<b>Descripción</b>	
Se requiere computar las calificaciones de los estudiantes para obtener las notas finales y las estadísticas de las Rubricas.	
<b>Criterios de Aceptación</b>	
<ul style="list-style-type: none"> <li>• El Prototipo permitirá Computar las Calificaciones cuando se haya creado una materia.</li> <li>• El Prototipo permitirá el uso de un atajo de Teclado para Computar las Calificaciones de una manera más rápida y sencilla.</li> <li>• El prototipo calculará automáticamente las notas Primer Bimestre, Segundo Bimestre, Bimestre 1 + Bimestre 2 y Examen.</li> <li>• El Prototipo calculará las estadísticas (Promedio, Nota Máxima y Nota Mínima) de las Rubricas y Notas Finales.</li> <li>• El Prototipo obtendrá la Información General cuando se computen las calificaciones.</li> </ul>	

RF 08	<b>Asignar Calificaciones</b>
<b>Usuario:</b>	Profesor No Vidente
<b>Prioridad:</b>	Alta
<b>Descripción</b>	
Se requiere poder asignar calificaciones de los estudiantes.	
<b>Criterios de Aceptación</b>	
<ul style="list-style-type: none"> <li>• El Prototipo permitirá ingresar calificaciones de los estudiantes guiándolos en todo momento por guías de voz.</li> <li>• El Prototipo permitirá solo el ingreso de números con un rango permitido de 0 a 10 y el punto como separador decimal.</li> <li>• El prototipo no permitirá el ingreso de calificaciones en las notas: Primer Bimestre, Segundo Bimestre, Bimestre 1 + Bimestre 2 y Examen Final debido a que estas son calculadas automáticamente.</li> <li>• El prototipo permitirá la movilización en las celdas de calificaciones de la Hoja de Cálculo.</li> </ul>	

RF 09	<b>Obtener Información General de la Materia</b>
<b>Usuario:</b>	Profesor No Vidente
<b>Prioridad:</b>	Media
<b>Descripción</b>	
Se requiere obtener información de los estudiantes que reprueban la materia y los que deben rendir obligatoriamente el Examen Final.	
<b>Criterios de Aceptación</b>	
<ul style="list-style-type: none"> <li>• El Prototipo permitirá obtener la Información de la Materia cuando se haya Computado las calificaciones.</li> <li>• El Prototipo obtendrá los nombres de los estudiantes que pierden la materia y se quedan en supletorio.</li> <li>• El Prototipo permitirá obtener información de la materia siempre y cuando exista al menos una materia ya creada.</li> <li>• El Prototipo permitirá que al momento de computar</li> </ul>	

RF 10	<b>Exportar Información</b>
<b>Usuario:</b>	Profesor No Vidente
<b>Prioridad:</b>	Media
<b>Descripción</b>	
Se requiere descargar la información en formato .xlsx, .csv o PDF.	
<b>Criterios de Aceptación</b>	
<ul style="list-style-type: none"> <li>• El Prototipo permitirá exportar la información siempre y cuando exista al menos una materia creada.</li> <li>• El Prototipo habilitará las opciones de Descarga una vez que se haya computado las calificaciones.</li> <li>• El Prototipo exportará los archivos con un nombre general seguido de la fecha de descarga.</li> </ul>	

### 2.1.3 REQUERIMIENTOS NO FUNCIONALES (RNF)

Estos requerimientos son los que especifican criterios para evaluar la operación de un servicio y a su vez asegura el cumplimiento de los objetivos planteados inicialmente [29].

Dichos requerimientos se detallan en las siguientes Historias de Usuario

RNF 01	<b>Usabilidad del Prototipo</b>
<b>Usuario:</b>	Profesor No Vidente
<b>Prioridad:</b>	Alta
<b>Descripción</b>	
Se requiere tener una interfaz amigable, intuitiva y fácil de utilizar para que el usuario pueda utilizarla sin ninguna dificultad.	
<b>Criterios de Aceptación</b>	
<ul style="list-style-type: none"> <li>• El Prototipo tendrá los campos mínimos y necesarios para un funcionamiento óptimo.</li> <li>• La interfaz tendrá un tamaño de letra adecuado en caso de que usuarios sin discapacidad visual lo utilicen.</li> </ul>	

RNF 02	<b>Ejecución Lineal del Prototipo</b>
<b>Usuario:</b>	Profesor No Vidente
<b>Prioridad:</b>	Alta
<b>Descripción</b>	
Se requiere un flujo de ejecución ordenado y lógico para que el usuario no cometa errores de ingreso de información, es decir tendrá una ejecución lineal el prototipo.	
<b>Criterios de Aceptación</b>	
<ul style="list-style-type: none"> <li>• El Prototipo alertará de posibles acciones incorrectas ejecutadas por el Profesor No Vidente.</li> </ul>	

RNF 03	<b>Flujo de las Guías de Voz</b>
<b>Usuario:</b>	Profesor No Vidente
<b>Prioridad:</b>	Alta
<b>Descripción</b>	
Se requiere un flujo de ejecución ordenado de las Guías de Voz para que no exista varios eventos auditivos al mismo tiempo.	
<b>Criterios de Aceptación</b>	
<ul style="list-style-type: none"> <li>• El Prototipo dejará avanzar a través de la aplicación una vez que se haya terminado la Guía de Voz del lugar que se encuentre.</li> <li>• El sonido generado deberá tener un rango audible adecuado.</li> </ul>	

RNF 04	<b>Compatibilidad del Prototipo</b>
<b>Usuario:</b>	Profesor No Vidente
<b>Prioridad:</b>	Alta
<b>Descripción</b>	
Se requiere que el Prototipo sea compatible con el sistema Operativo Windows debido a que este fue desarrollado para dicho sistema.	
<b>Criterios de Aceptación</b>	
<ul style="list-style-type: none"> <li>• El Prototipo será una Aplicación de Escritorio.</li> </ul>	

RNF 05	<b>Atajos de Teclado</b>
<b>Usuario:</b>	Profesor No Vidente
<b>Prioridad:</b>	Media
<b>Descripción</b>	
Se requiere tener la opción de conocer los Atajos de Teclado del Prototipo.	
<b>Criterios de Aceptación</b>	
<ul style="list-style-type: none"> <li>• El Prototipo permitirá ingresar a una ventana donde se mostrará todos los Atajos de Teclado que se puede utilizar en el Prototipo.</li> <li>• El Prototipo aceptará un Atajo de Teclado para ir a la ventana de Ayuda.</li> </ul>	

## 2.1.4 DEFINICIÓN DE CASOS DE USO

El Diagrama de Casos de Uso permite visualizar las interacciones que podría tener un usuario con un Sistema [30]. Para realizar el Diagrama de Casos de Uso se utilizó la herramienta LucidChart [31].

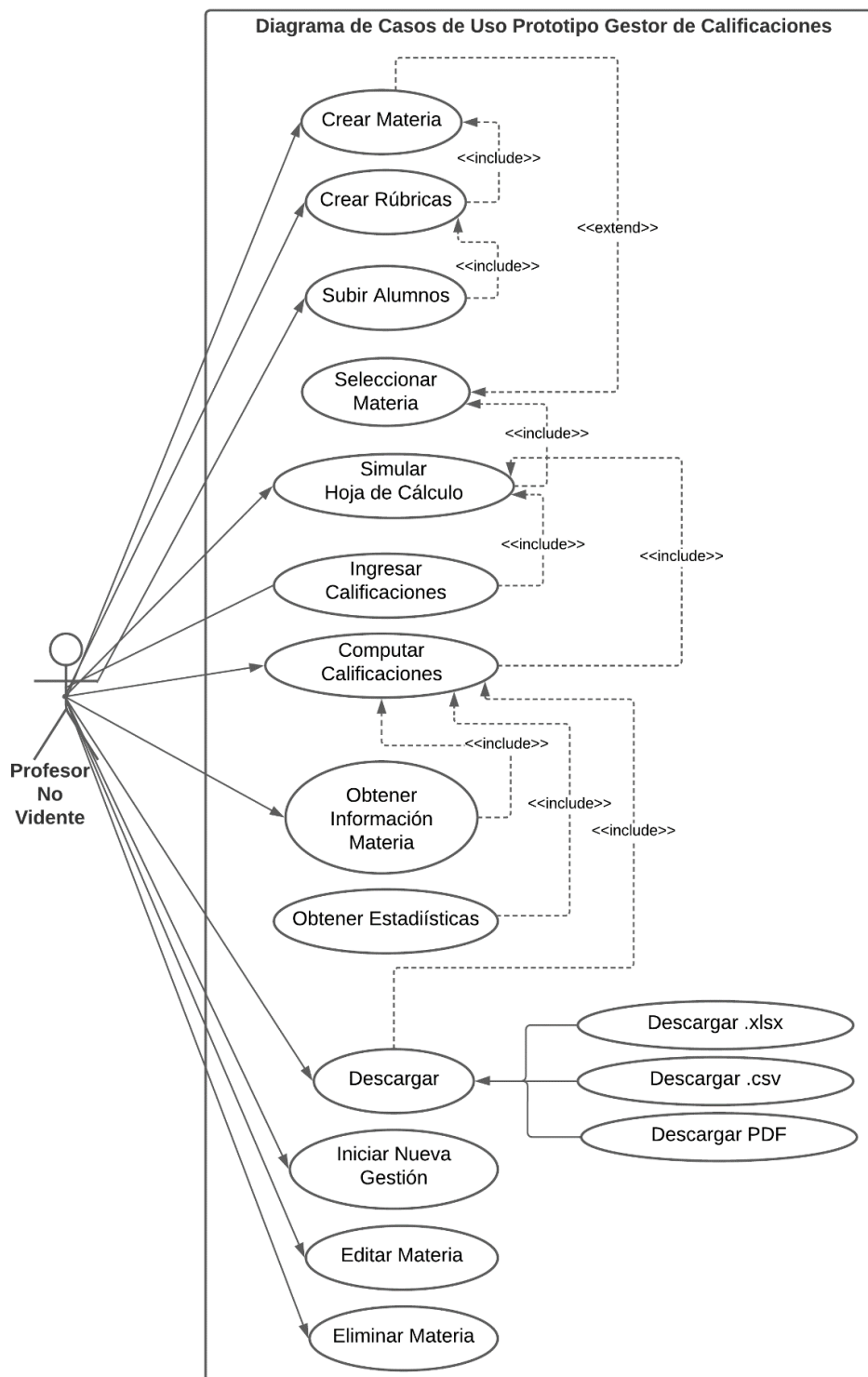


Figura 2.2. Diagrama de Casos de Uso del prototipo.



Se define al Profesor No Vidente como único actor del sistema dado que este actor es el único que interviene directamente en la ejecución de todas las funcionalidades del prototipo. En el Diagrama de Casos de Uso que se observa en la Figura 2.2 se define como único actor al Usuario No Vidente. Este podrá realizar todas las acciones que se detallan con la ayuda de guías de voz durante toda la ejecución del prototipo. Adicionalmente se utiliza una relación de extensión para la selección de la materia debido a que esta funcionalidad será opcional siempre que ya exista al menos una materia creada.

## 2.1.5 DISEÑO DE LA CAPA DE DATOS

La Capa de Datos es la capa que proporciona acceso simplificado a los datos almacenados en un Gestor de Base de Datos [32].

### 2.1.5.1 Diseño del Modelo Relacional para Bases de Datos

Para realizar el Diagrama Relacional se utilizó la herramienta LucidChart. En el Diagrama Relacional que se observa en la Figura 2.3. Este modelo tiene los campos necesarios: la tabla **Materia** tiene 6 campos, entre los cuales consta su clave primaria, el nombre de la materia, número de estudiantes, número de rúbricas primer bimestre, números de rúbricas del segundo bimestre y una clave foránea que relaciona la tabla **Materia** con la tabla **Profesor**. Se diseñó la tabla de esa manera debido a que posteriormente se necesitará simular una hoja de cálculo, la cual deberá tener dimensiones establecidas para que el profesor no vidente pueda ejecutar las funcionalidades del prototipo en una simulación de hoja definida sin la idea de que pueda ir fuera del rango donde se encuentra trabajando. También se puede observar que la Materia tiene relaciones con la tabla **Estudiante**, **Calificación** y **Rúbrica**. Esto se lo diseñó para un buen funcionamiento del prototipo.

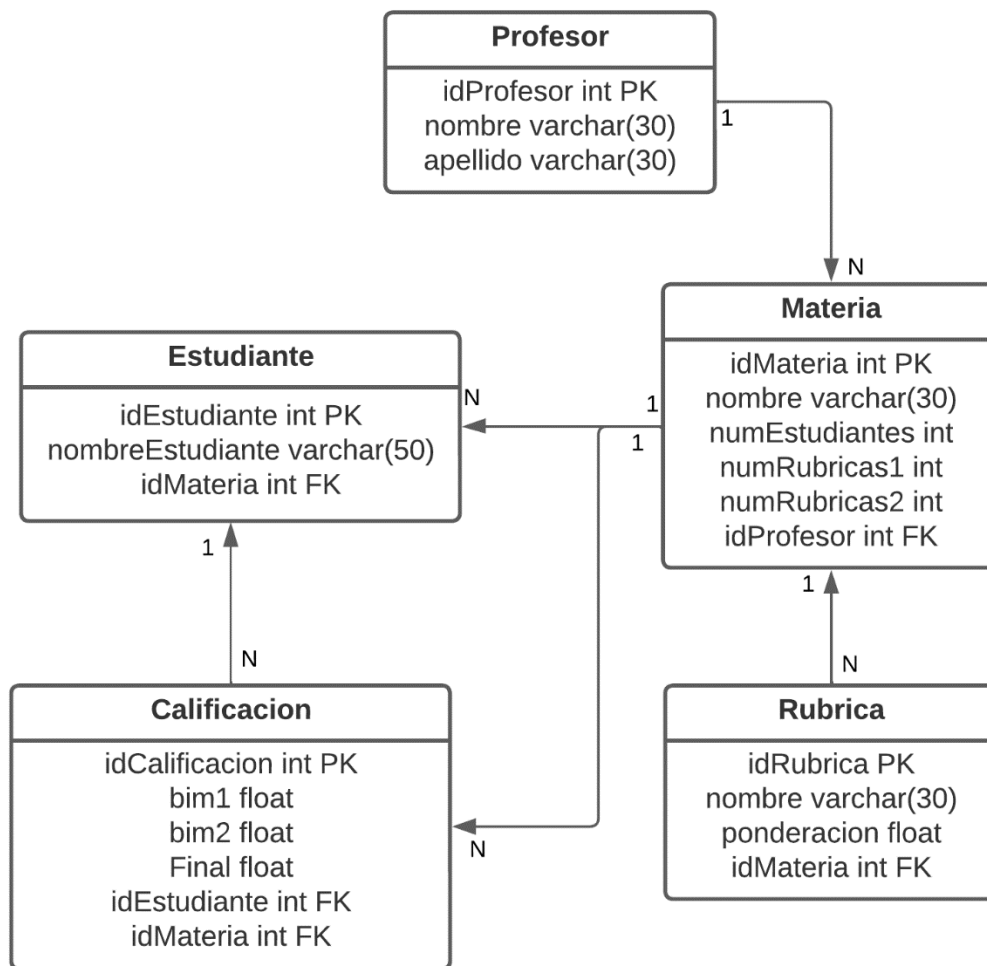


Figura 2.3. Diagrama Relacional del prototipo.

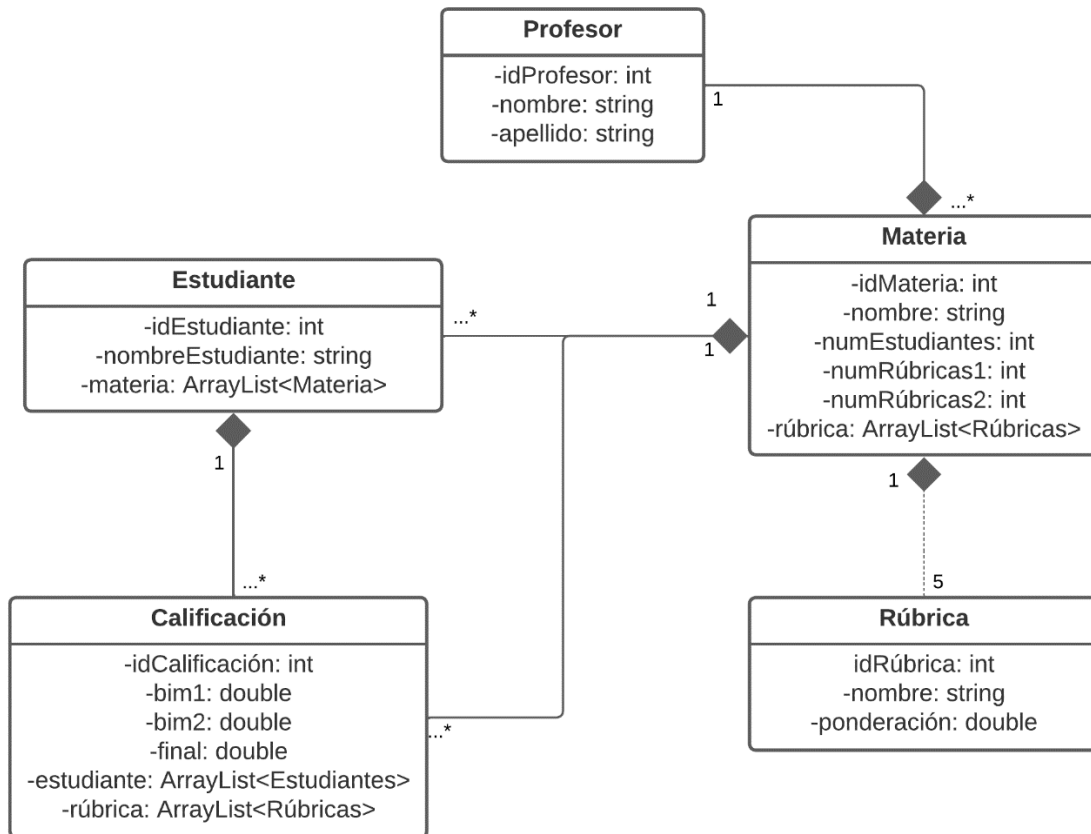
## 2.1.6 DISEÑO DEL DIAGRAMA DE CLASES

La Capa de Negocio consiste en la lógica que realizará las funciones principales del prototipo como: el procesamiento de los datos, implementación de funciones de negocio y administración de recursos externos.

### 2.1.6.1 Diseño del Diagrama de Clases

Para realizar el Diagrama de Clases se utilizó la Herramienta LucidChart. El Diagrama de Clases es el que describe la estructura del prototipo mostrando todas las clases, atributos y relaciones entre los objetos [33]. Se puede observar en la Figura 2.4, el Diagrama de Clases del Prototipo. En este Diagrama está la Clase **Materia**, con 6 atributos de tipo privado, el primero de ellos es un identificador de la materia del tipo int, el nombre de la

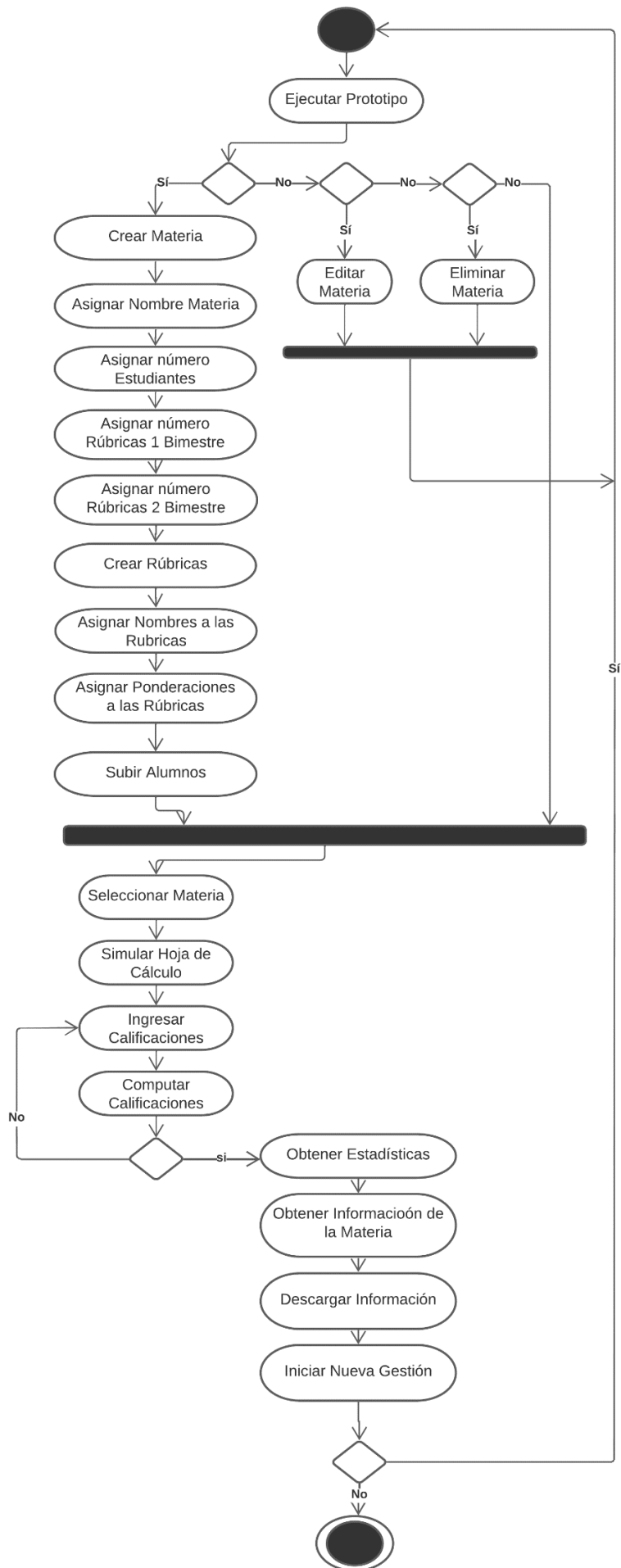
materia de tipo string, número de estudiantes, número de rúbricas del primer y segundo bimestre del tipo int y una lista de rúbricas del tipo ArrayList ya que esta materia podrá tener hasta un máximo de 5 rúbricas por materia. Se observa que existen relaciones de composición debido a que la materia se compone de la clase **Estudiante**, **Profesor** y **Rúbrica**, se puede decir que esta clase es una de las principales que se utilizará.



**Figura 2.4.** Diagrama de Clases del prototipo.

### 2.1.6.2 Diseño del Diagrama de Actividades

Para realizar el Diagrama de Actividades que se observa en la Figura 2.5 se utilizó la herramienta LucidChart. Mediante este Diagrama, se podrá modelar la secuencia lógica de funcionamiento del prototipo. De este modo se tendrá claro cuál es el orden de ejecución de las tareas y las condiciones previas que se deben cumplir.



**Figura 2.5.** Diagrama de Actividades del prototipo.

## 2.1.7 DISEÑO DE LA CAPA DE PRESENTACIÓN

La Capa de Presentación es la responsable de la interacción entre el usuario y la aplicación, aquí es donde se le muestra la información. Esta capa captura los datos del usuario en el formulario e invoca a la Capa de Negocio para transmitir requerimientos del usuario como almacenaje, edición y recuperación de información para las respectivas consultas [34].

### 2.1.7.1 Diseño de la Interfaz Gráfica

La Interfaz Gráfica está enfocada a un público con discapacidad visual pero no será de uso exclusivo de este público dado que se diseñará la interfaz para que utilice cualquier usuario con o sin discapacidad visual. Se diseñó la interfaz gráfica de forma secuencial, esto con el objetivo de que el usuario cometa la mínima cantidad de errores [35]; es decir el flujo de actividades del usuario estará pre establecido, guiando al usuario a seguir un orden específico en todo momento. En la

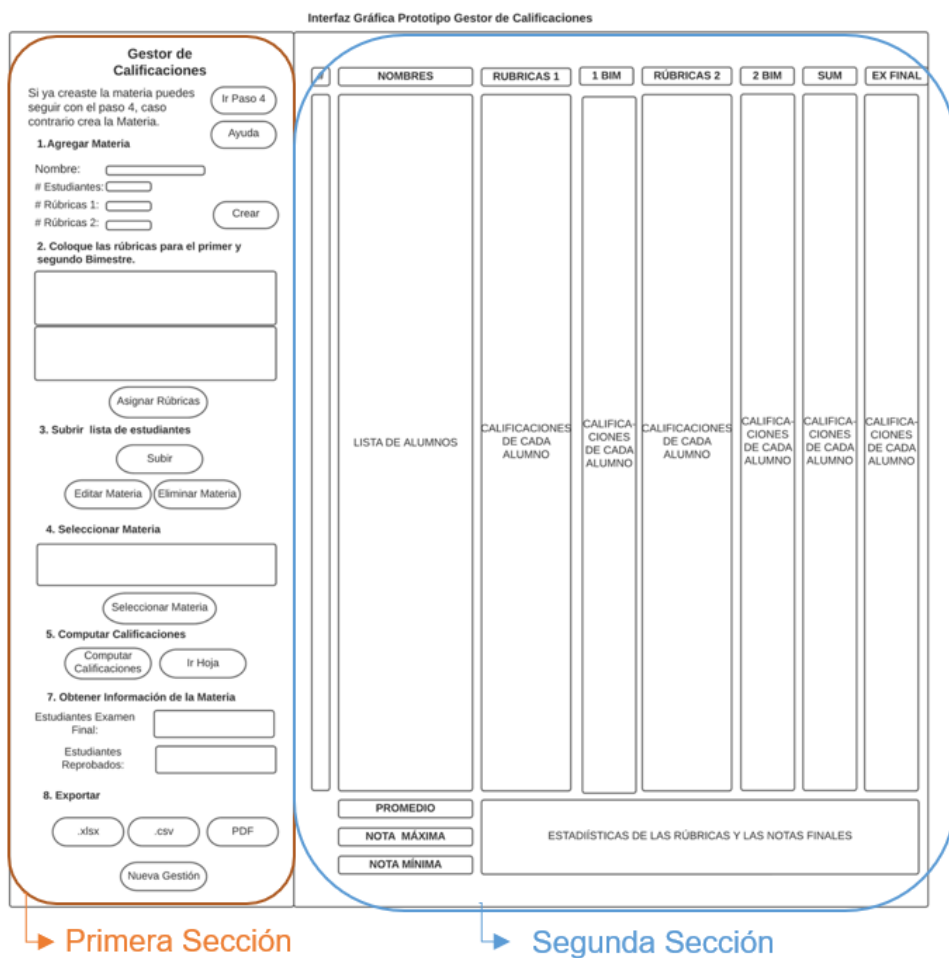
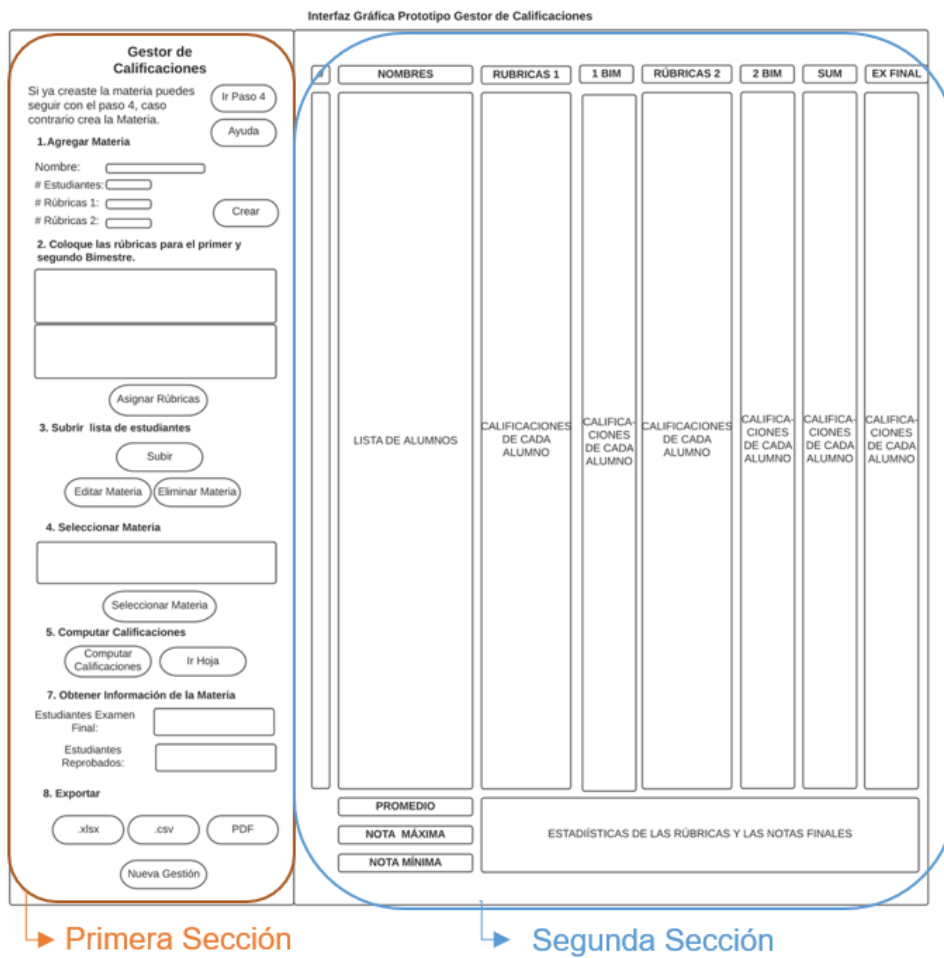


Figura 2.6 se observa el diseño de la pantalla principal del prototipo y consta de dos secciones.



**Figura 2.6.** Interfaz Principal del prototipo.

En la primera sección que se muestra en la

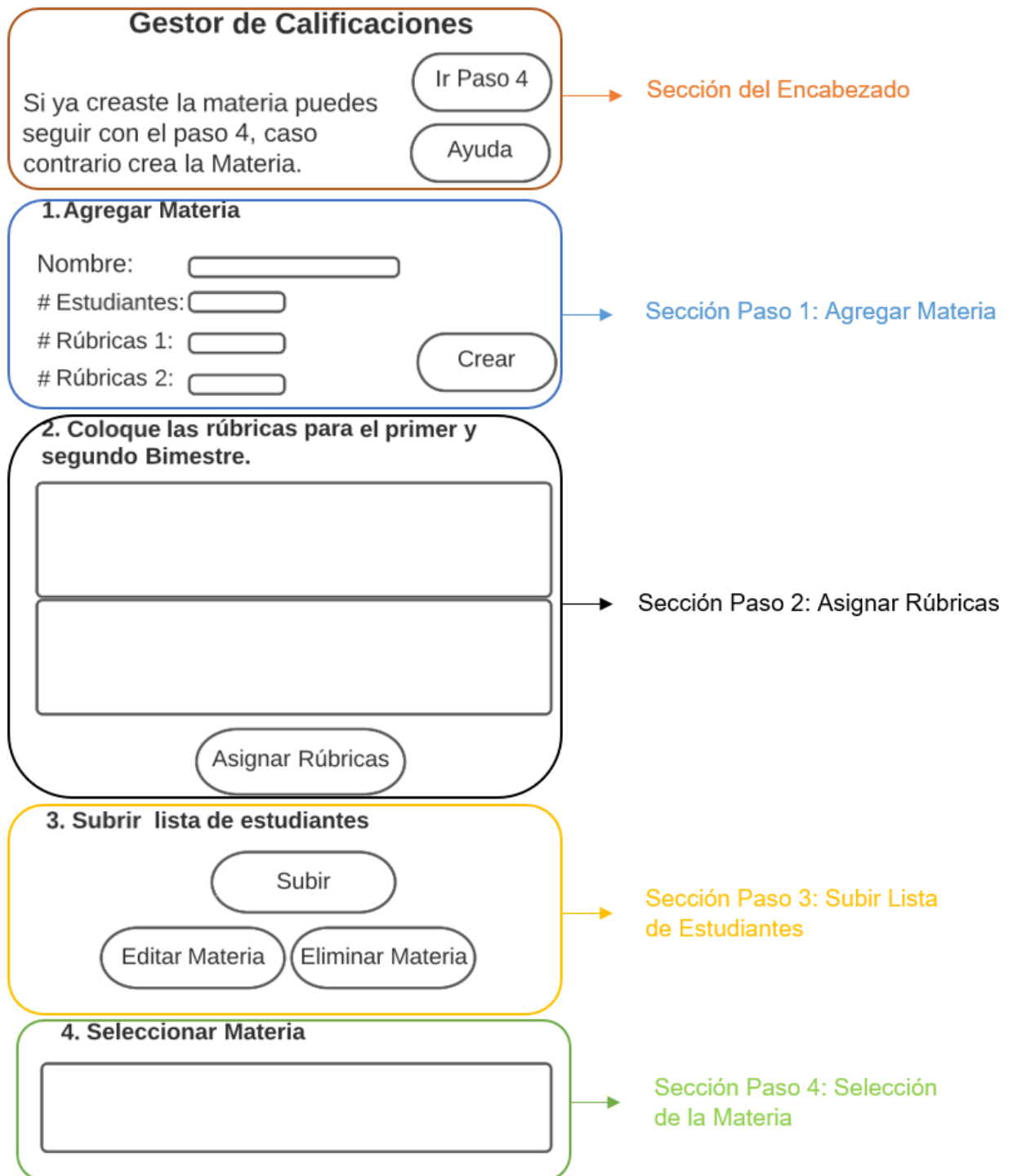


Figura 2. 7 y

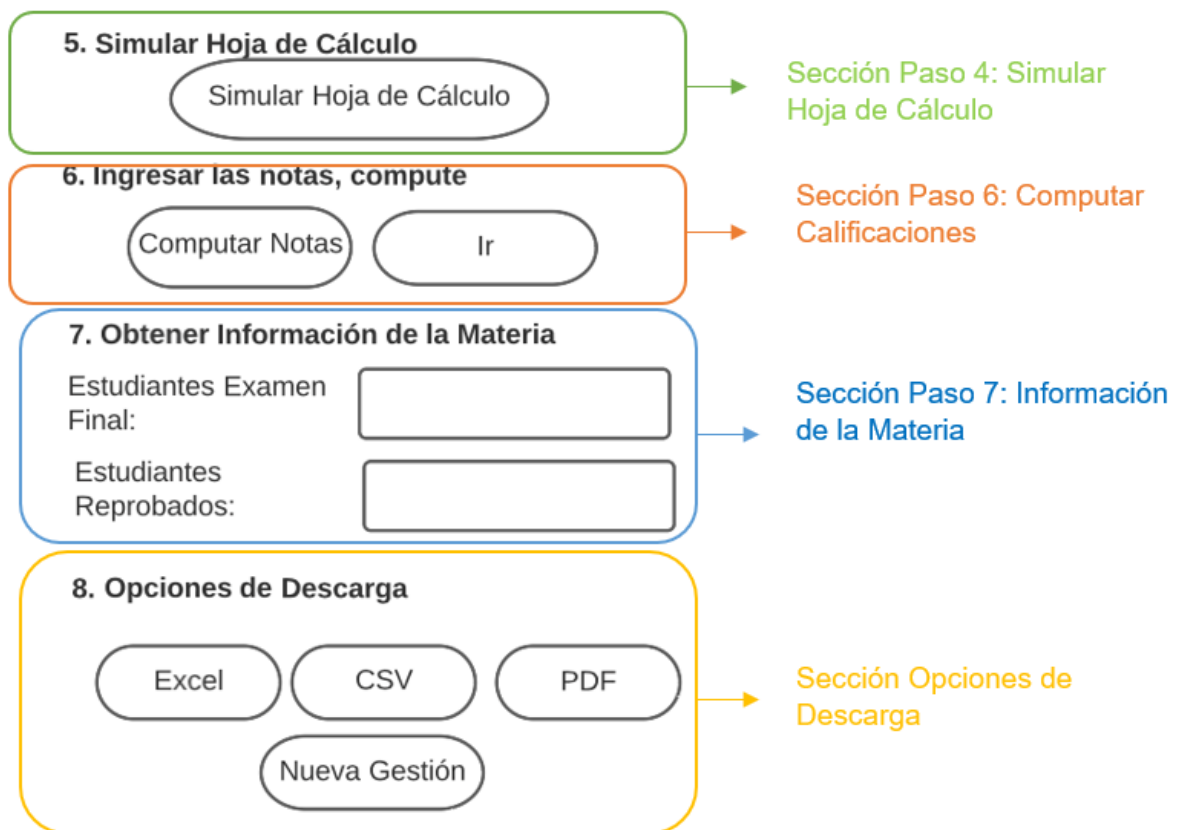


Figura 2. **8** se pedirá que se cree la Materia, asigne sus rúbricas, suba el archivo de alumnos, simule la hoja de cálculo y obtenga la información de la materia y tenga la posibilidad de descargar la información en archivo .xlsx, .csv o PDF.



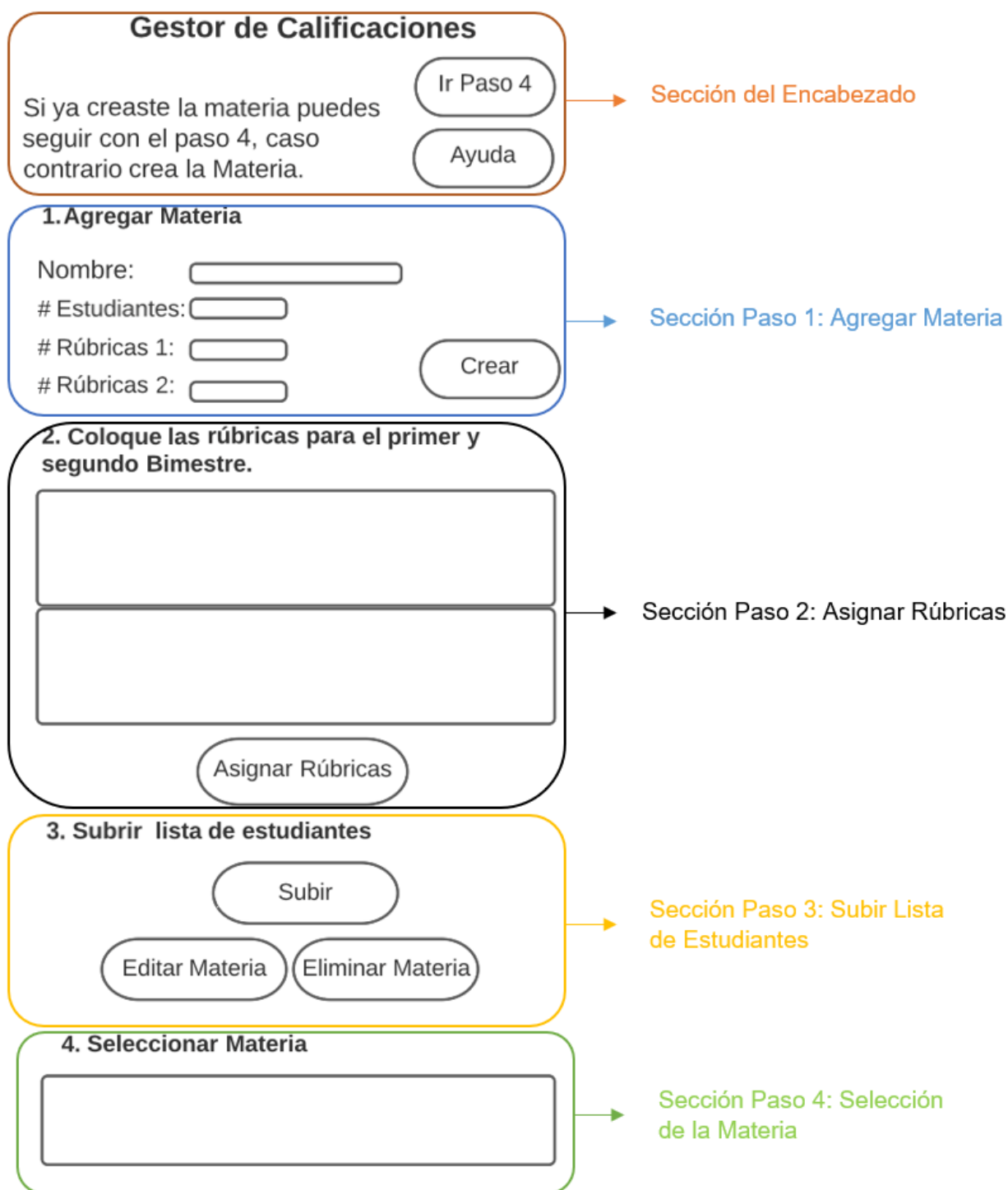
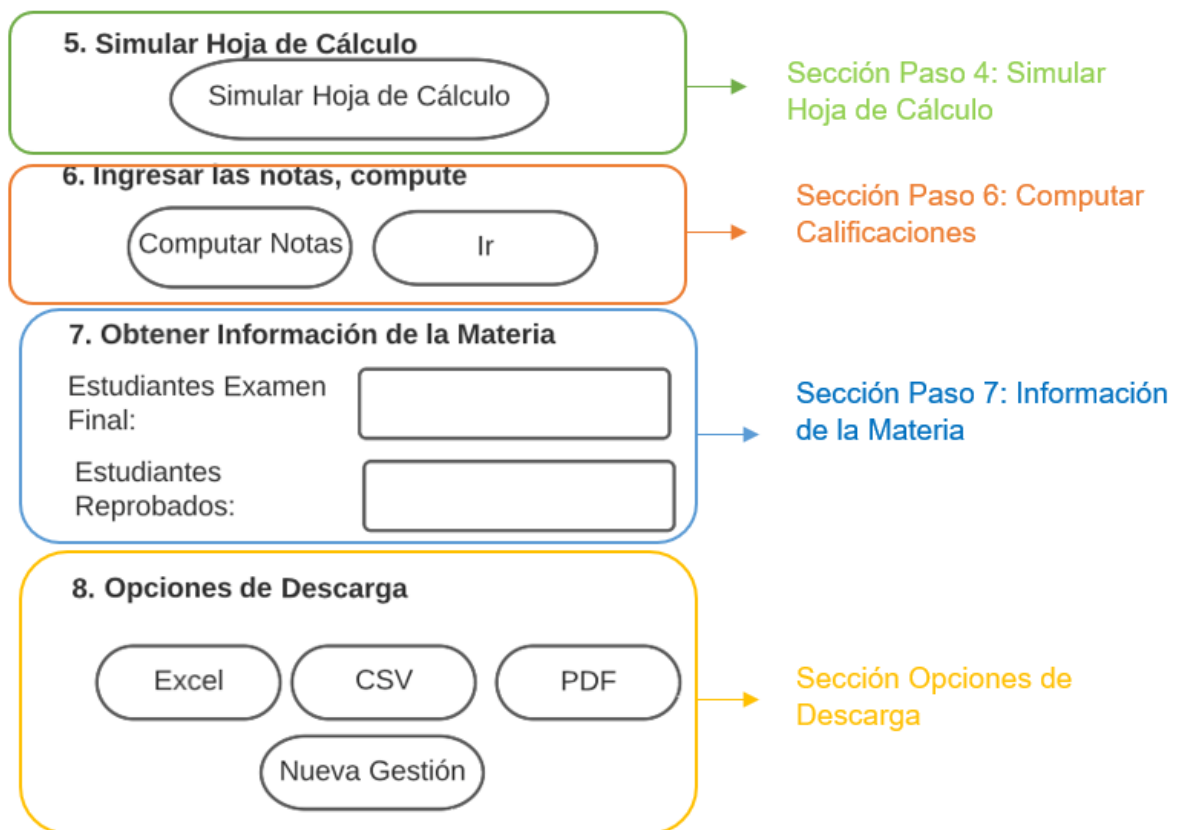


Figura 2. 7. Primera sección de la interfaz.



**Figura 2. 8.** Primera sección de la interfaz.

La segunda sección estará destinada exclusivamente a la simulación de la hoja de cálculo (Figura 2.10). En esta simulación de una hoja de cálculo estará la lista de estudiantes, las rúbricas creadas con sus respectivas ponderaciones del primer y segundo bimestre, las notas finales del primer y segundo bimestre, la suma de los dos bimestres y la nota del examen final. Al inferior de la simulación de la hoja de cálculo estarán las estadísticas, las cuales serán promedio, nota máxima y nota mínima. Estas estadísticas serán auto calculadas y estas corresponderán a la información de todos los alumnos de cada rúbrica del primer y segundo bimestre y cada nota final de la materia.

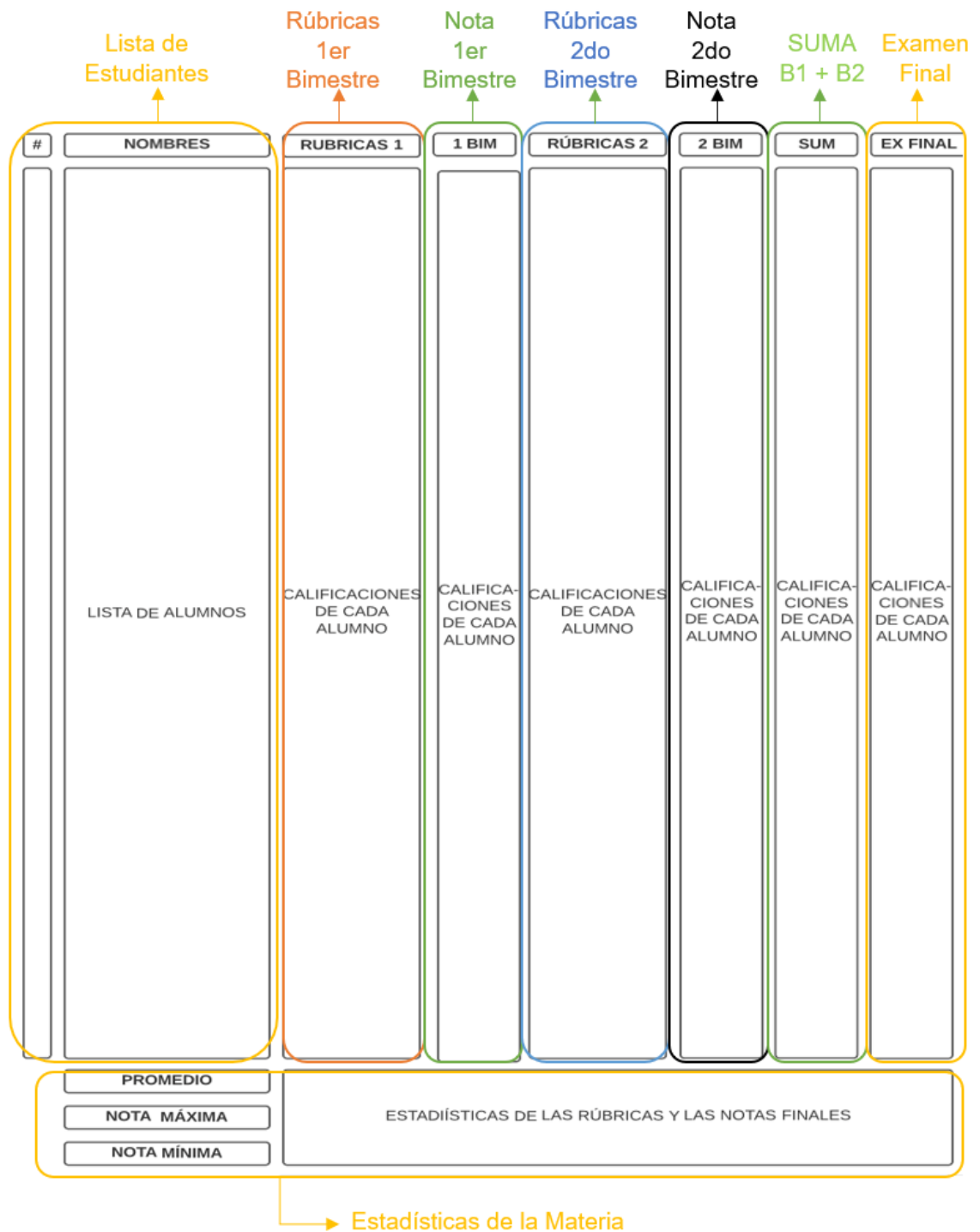


Figura 2. 9. Segunda sección de la interfaz.

### 2.1.7.2 Definición Atajos de Teclado

El atajo de teclado es una tecla o un conjunto de teclas que realizan una acción definida [36]. Los atajos de teclado pueden facilitar el uso de interfaces de usuarios, porque se puede realizar acciones con teclas específicas dando al usuario rapidez y ayudándolo a cometer menos errores.

**Tabla 2.1.** Descripción de los Atajos de Teclado disponibles en el Prototipo.

<b>Acción</b>	<b>Atajo de Teclado</b>
Ayuda	SHIFT + ALT + H
Computar	SHIFT + ALT + C
Ir al Paso 4	SHIFT + ALT + 4
Ir al Paso 7	SHIFT + ALT + 7
Editar	SHIFT + ALT + E
Eliminar Materia	SHIFT + ALT + SPR
Exportar	SHIFT + ALT + G
Nueva Gestión	SHIFT + ALT + N
Ir a la Hoja de Cálculo	SHIFT + ALT + I
Cerrar	SHIFT + ESC

Para definir los atajos de teclado se empleará el uso de teclas conocidas como SHIFT, ALT y la adición de otra tecla; de esta manera se evitará interferir con los comandos propios del sistema Windows [37]. En la Tabla 2.1 se muestran los atajos de teclado que tendrá el prototipo.

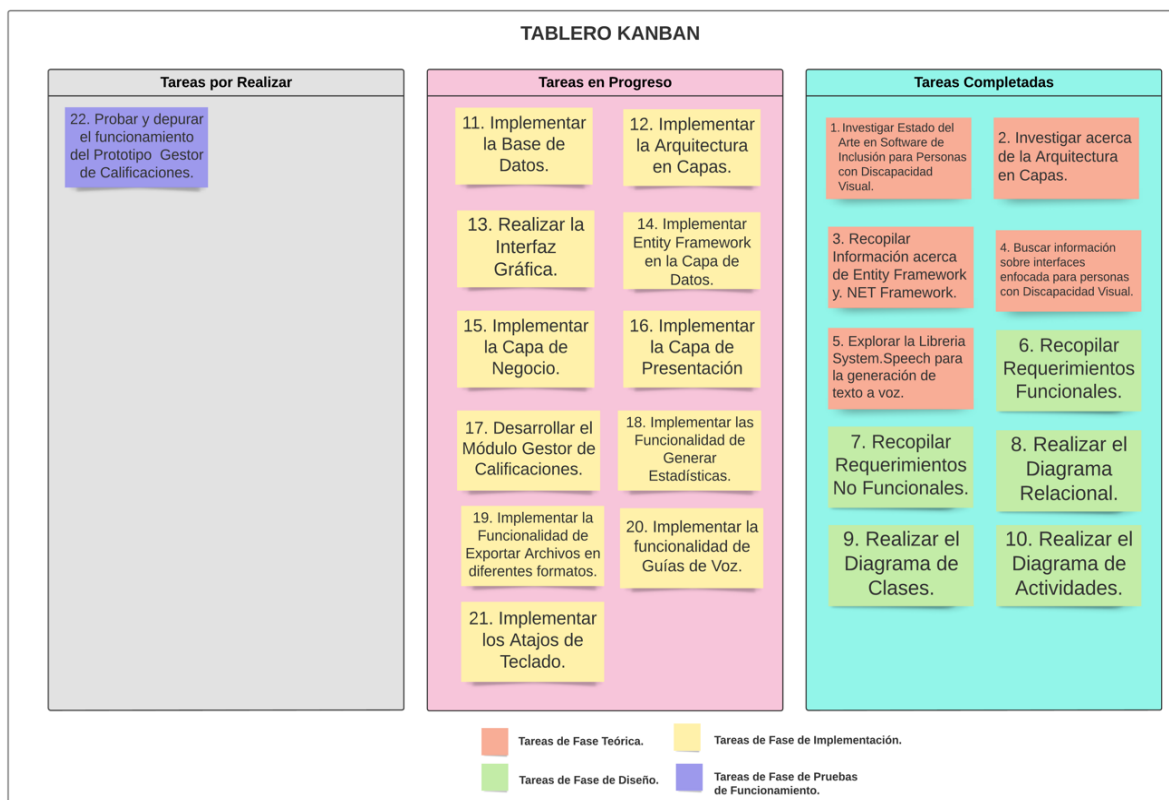
## 2.2 IMPLEMENTACIÓN

Durante esta sección se presentará el proceso de desarrollo y codificación del Prototipo Gestor de Calificaciones. En la Capa de Acceso a Datos se mapeará toda la base de datos del prototipo con la ayuda de EF. En la Capa de Negocio se tendrán todas funciones que permitirá al prototipo cumplir con los objetivos planteados. Finalmente, la Capa de Presentación muestra la interfaz del prototipo que se diseñó.

### 2.2.1 ACTUALIZACIÓN DEL TABLERO KANBAN

En la Figura 2.10 se observan las tareas que ya han sido completadas satisfactoriamente y las tareas que están en proceso. Adicionalmente, tras la finalización de todas las tareas

se procederá a la etapa final del presente Trabajo de Integración Curricular, correspondiente a las pruebas del prototipo desarrollado.



**Figura 2.10.** Actualización del tablero Kanban en la Etapa de Implementación.

## 2.2.2 IMPLEMENTACIÓN DE LA CAPA DE DATOS

Al momento de implementar la Capa de Datos desde la Herramienta de Visual Studio se realizó un mapeo de toda la base de datos creada anteriormente con la ayuda de EF. En el Código 2.1 se muestra la clase Materia donde se encuentran los atributos que se diseñó en la base de datos en el **Apartado 2.1.5.1**. En esta clase tenemos una cabecera que nos indica que el código que se visualiza se generó automáticamente dado a que se utilizó EntityFramework. Luego se muestra un constructor propio de la clase, posterior se tiene las propiedades *get* y *set* de todos los atributos para manejar la información de esta clase y por último se tiene atributos virtuales que hacen referencia a las claves foráneas de las tablas para así poder relacionarse con las demás clases. Aquí se observa que se podrá relacionar con la clase Calificación, Profesor y Rúbrica. Para observar toda la Implementación de la Capa de Datos observe el **Anexo III**.

```

//-----
//
// <auto-generated>
//     Este código se generó a partir de una plantilla.
//
//     Los cambios manuales en este archivo pueden causar un comportamiento
//     inesperado de la aplicación.
//     Los cambios manuales en este archivo se sobrescribirán si se regenera
//     el código.
// </auto-generated>
//-----
//
namespace CapaDatos
{
    using System;
    using System.Collections.Generic;

    public partial class Materia
    {
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
        "CA2214:DoNotCallOverridableMethodsInConstructors")]

        public Materia()
        {
            this.Calificacion = new HashSet<Calificacion>();
            this.Estudiante = new HashSet<Estudiante>();
            this.Rubrica = new HashSet<Rubrica>();
        }
        public int idMateria { get; set; }
        public string nombreMateria { get; set; }
        public int numEstudiantes { get; set; }
        public int numRubricasIB { get; set; }
        public int numRubricasIIB { get; set; }
        public Nullable<int> idTeacher { get; set; }

        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
        "CA2227:CollectionPropertiesShouldBeReadOnly")]

        public virtual ICollection<Calificacion> Calificacion { get; set; }

        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
        "CA2227:CollectionPropertiesShouldBeReadOnly")]
        public virtual ICollection<Estudiante> Estudiante { get; set; }

        public virtual Profesor { get; set; }
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
        "CA2227:CollectionPropertiesShouldBeReadOnly")]

        public virtual ICollection<Rubrica> Rubrica { get; set; }
    }
}

```

**Código 2.1.** Clase Materia del prototipo.

### 2.2.3 IMPLEMENTACIÓN DE LA CAPA DE NEGOCIO

La Capa de Negocio consiste en la lógica que realizará las funciones principales del prototipo. En el Código 2.2 se observa la clase MateriaNegocio, se la denominó de esa manera para que no exista confusión con la Clase Materia de la Capa de Acceso a Datos.

```
namespace CapaNegocio
{
    public class MateriaNegocio
    {
        public int numMateriasSistema()
        {
            using (dbNotas1Entities db = new dbNotas1Entities())
            {
                var lst5 = db.Materia; // aquí obtengo todos las Materias LINQ
                return lst5.Count();
            }
        }
        public int obtenerRubricasPrimero(int idSeleccion)
        {
            using (dbNotas1Entities db = new dbNotas1Entities())
            {
                var lst = (from d in db.Materia
                           where d.idMateria == idSeleccion
                           select d.numRubricasIB).SingleOrDefault();
                return lst;
            }
        }
        public int obtenerRubricasSegundo(int idSeleccion)
        {
            using (dbNotas1Entities db = new dbNotas1Entities())
            {
                var lst = (from d in db.Materia
                           where d.idMateria == idSeleccion
                           select d.numRubricasIIB).SingleOrDefault();
                return lst;
            }
        }
        public int obtenerNumeroEstudiantes(int idSeleccion)
        {
            using (dbNotas1Entities db = new dbNotas1Entities())
            {
                var lst = (from d in db.Materia
                           where d.idMateria == idSeleccion
                           select d.numEstudiantes).SingleOrDefault();
                return lst;
            }
        }
    }
}
```

**Código 2.2.** Clase MateriaNegocio del prototipo.

La clase MateriaNegocio tiene las funciones que se van a necesitar a lo largo del desarrollo del prototipo. Se observa que esta clase se encuentra dentro de la Capa Negocio.

- En primer lugar, se tiene la función ***numMateriasSistema()***, la cual devuelve un valor del tipo int, con el número de materias que se encuentran en el sistema. Esta función es de gran importancia debido a que se puede validar que exista al menos una materia en el prototipo para ejecutar algunas funcionalidades como el RF04 del **Apartado 2.1.2**, el cual indica que para simular una hoja de cálculo deberá existir al menos una materia creada.
- La segunda función ***obtenerRubricasPrimero(int idSeleccion)***, permite obtener el número de rúbricas del primer bimestre, con un parámetro de entrada, el cual es el idSeleccion, para de cual materia debe regresar la información solicitada.
- La tercera función es ***obtenerRubricasSegundo(int idSeleccion)***, la cual devuelve el número de rúbricas del segundo bimestre de una materia seleccionada con idSeleccion como parámetro de entrada.
- La cuarta función es ***obtenerNumeroEstudiantes(int idSeleccion)***, esta devuelve el número de estudiantes de una materia seleccionada, tiene como parámetro de entrada el idSeleccion. Las últimas tres funciones explicadas sirven para simular la hoja de cálculo y delimitarla, siendo el número de estudiantes las filas de la simulación de la hoja de cálculo y las rúbricas del primer y segundo bimestre las columnas de la simulación de la hoja de cálculo. Para observar toda la Implementación de la Capa de Negocio observe el **Anexo III**.

#### **2.2.4 IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN**

En primer lugar, se utilizó el Diseño de la Interfaz del Apartado 2.1.7.1 para empezar a diseñar la interfaz principal del prototipo en la Capa de Presentación dentro de un formulario de Windows. Se colocó exclusivamente los cuadros de diálogo necesarios para el funcionamiento adecuado del prototipo y se siguieron los parámetros definidos en los requerimientos funcionales y requerimientos no funcionales del Apartado 2.1.2 y 2.1.3. La interfaz final realizada se puede observar en la Figura 2.11. Para observar toda la Implementación de la Capa de Presentación observe el **Anexo III**.





Figura 2.11. Interfaz Principal del prototipo.

En la Figura 2.9 y Figura 2.10, se observa la primera sección del prototipo Gestor de Calificaciones. En esta se describe a detalle las secciones que contiene:

- La sección del encabezado contiene información previa a comenzar a utilizar el prototipo. En esta se encuentra la opción para ir al paso 4 y una opción de Ayuda donde se mostrará todos los atajos de teclado disponible en el teclado.
- La sección Agregar Materia contiene los campos necesarios para crear una nueva materia.
- La sección del Paso 2 (Agregar Rúbricas): contiene los campos generados dinámicamente para asignar las rúbricas y ponderaciones del primer y segundo bimestre.
- La sección del Paso 3 (Subir Lista de Estudiantes): contiene la opción de subir una lista de Estudiantes en un formato Excel predeterminado obtenido del SA II.
- La sección del Paso 4 (Seleccionar materia): contiene una lista donde se despliegan las materias que se encuentran creadas en el prototipo.

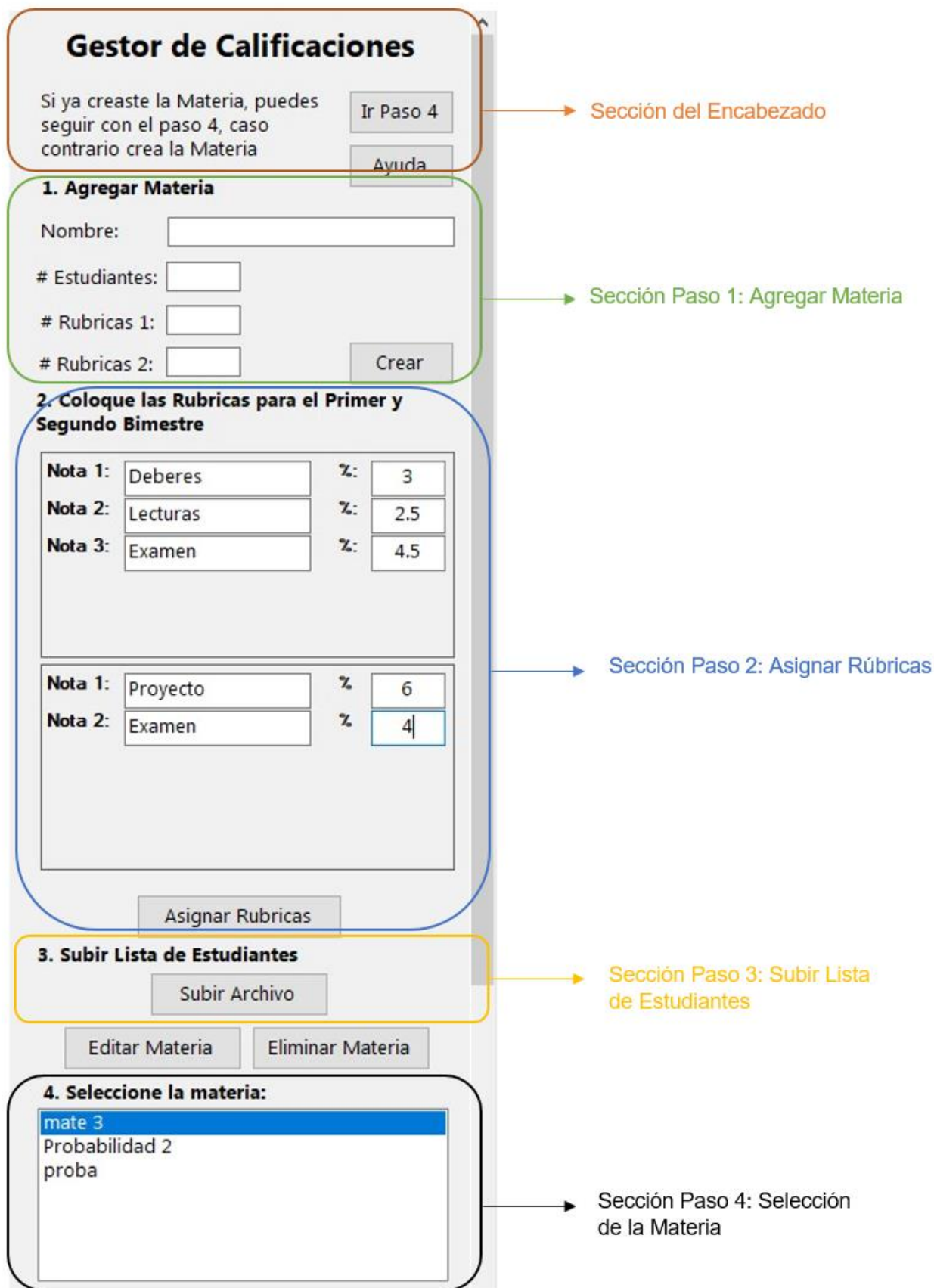
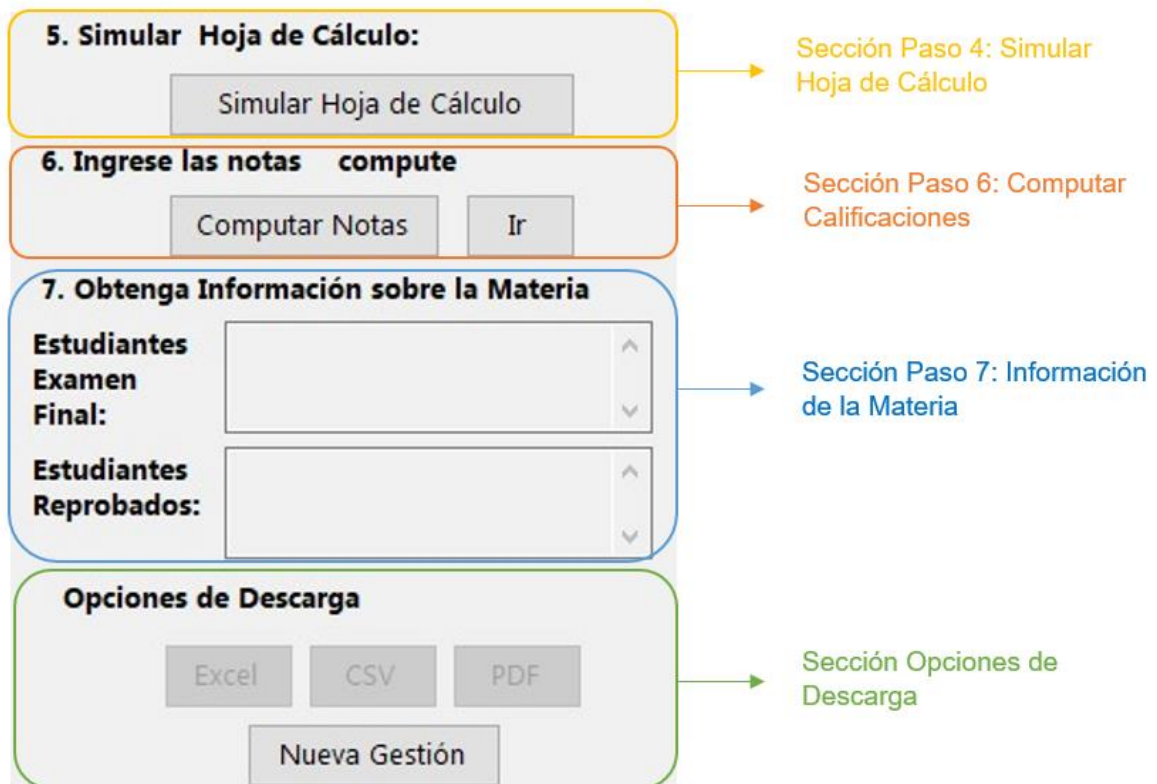


Figura 2.12. Primera sección de la interfaz.



**Figura 2.13.** Primera sección de la interfaz.

- La sección del Paso 5 permite simular la hoja de cálculo.
- La sección del Paso 6 permite computar las calificaciones de los alumnos.
- La sección del Paso 7 permite conocer la información de los alumnos que tienen que rendir el Examen Final y los alumnos que reprueban la materia.
- Por último, la sección de Descargas muestra los diferentes formatos que se puede descargar la información.

En la Figura 2.14 se observa la segunda sección de la Interfaz, en donde se simulará la hoja de cálculo. Aquí se puede observar cómo se encuentra dividida la simulación de la hoja de cálculo.

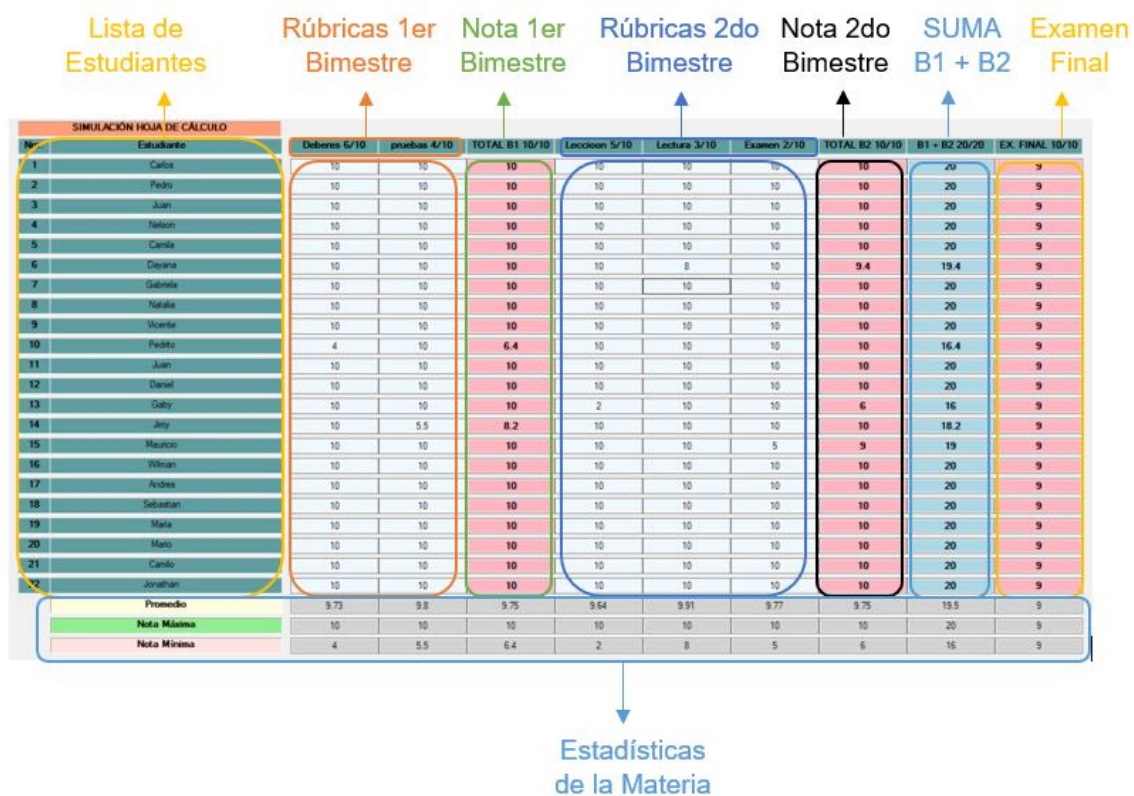


Figura 2.14. Segunda sección de la interfaz.

### 3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En el presente Capítulo se mostrarán los pormenores de las pruebas realizadas en el prototipo y los resultados obtenidos tras finalizar la etapa de implementación del prototipo, esto en contraste con los requerimientos planteados en la Etapa de Diseño en el **Apartado 2.1.2** y **Apartado 2.1.3**.

A lo largo del **Apartado 3.1**, correspondiente a Resultados se actualizará el tablero Kanban, se definirá el entorno de pruebas, la validación de Requerimientos Funcionales y No Funcionales. Siguiendo, en el **Apartado 3.2** se definirá las conclusiones obtenidas del prototipo y las recomendaciones en base a eventualidades al momento del desarrollo del prototipo.

### 3.1 RESULTADOS

Durante el presente Apartado se mostrarán los resultados obtenidos tras finalizar la etapa de implementación del prototipo Gestor de Calificaciones y las estadísticas del uso del prototipo con la metodología KLM GOMS.

#### 3.1.1 ACTUALIZACIÓN DEL TABLERO KANBAN

Se actualiza el tablero Kanban, mostrando la necesidad de completar las pruebas del prototipo desarrollado. Una vez finalizada esta tarea se procederá a dar por culminado el tablero.

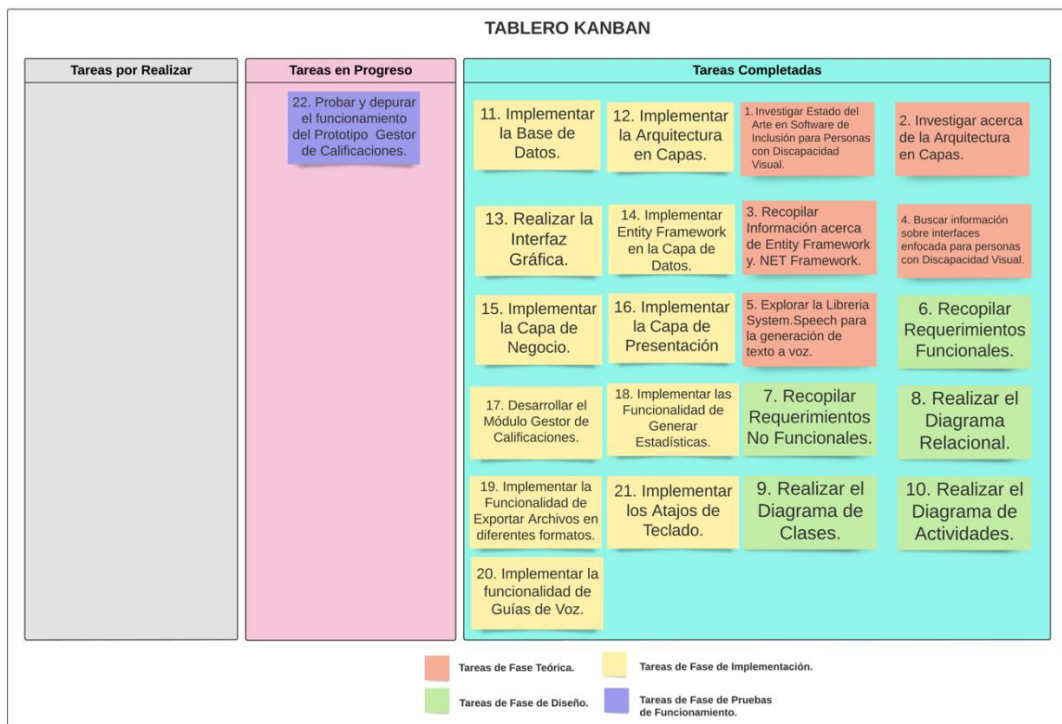


Figura 3.1. Actualización del tablero Kanban en la Etapa de Pruebas y Depuración.

#### 3.1.2 DEFINICIÓN DEL ENTORNO DE PRUEBAS

Para comprobar el funcionamiento del prototipo, se procedió a crear un entorno de pruebas controlado. Se tomó 5 personas, 3 hombres y 2 mujeres. Todos a partir de 30 años en adelante. Se escogió esta edad dado que el prototipo fue desarrollado para el uso de un profesor y se considera que 30 años es la edad mínima para que cumpla todos los requisitos para ser docente [38]. El entorno de pruebas presenta las siguientes condiciones:

- Los usuarios no tienen discapacidad visual, sin embargo, utilizaron el prototipo simulando ser personas no videntes. Esta condición se logra mediante el uso de vendas de tela de color negro para evitar que los usuarios puedan ver la pantalla del computador.
- El conocimiento del usuario del teclado se asume que es de un nivel adecuado para utilizarlo en las pruebas del prototipo.
- Las pruebas se desarrollan en un laboratorio de la universidad, con condiciones de luz ambientales normales, en un lugar tranquilo y sin ningún tipo de interrupción. Se ejecuta el prototipo en los ordenadores del laboratorio para evitar que los usuarios tengan que instalar algún tipo de aplicación de escritorio y minimizar el tiempo necesario para realizar las pruebas del prototipo.
- Los usuarios de prueba asisten a una etapa de inducción de alrededor de 2 horas, acerca del uso del prototipo, atajos de teclado que existen, separación decimal en las calificaciones con punto y pormenores generales.
- Para obtener una retroalimentación en cuanto al número de palabras que es adecuado al ser un prototipo que la guía de voz es parte fundamental. Se realiza pruebas con oraciones de menos de 5 palabras, de 5 a 10 palabras y más de 10 palabras. Se realiza pruebas con oraciones en un rango de longitud de 5 a 10 palabras con el fin de que el usuario perciba mejor el audio. Para esto se considera que las palabras con longitud menores de 3, serán omitidas.
- Una vez finalizadas las pruebas del prototipo, se realiza preguntas de manera oral a los usuarios para obtener una retroalimentación directa con la experiencia del usuario.
- Se realizan dos reuniones, la primera el viernes 24 de junio de 2022, tras la cual se toma una semana para corregir los errores encontrados en las pruebas realizadas. De esta manera, la segunda sesión tiene lugar el 30 de junio de 2022. Durante esta segunda sesión se verifica la corrección de los problemas encontrados.

### **3.1.3 ESTADÍSTICAS DEL USO DEL PROTOTIPO CON LA METODOLOGÍA KLM GOMS**

Se recalca que, no se ha encontrado el uso de la metodología KLM GOMS en otros proyectos de inclusión. Este trabajo, a más de probar Requerimientos Funcionales y No Funcionales en base a la percepción del Usuario, también utiliza esta metodología para comprobar el correcto y ágil funcionamiento del sistema.

Posterior a la capacitación, se empieza el periodo de pruebas bajo el escenario descrito realizando 2 tipos de pruebas: Primero, evaluando los tiempos de demora en la ejecución de los accesos rápidos y evaluándolos bajo la metodología. Como se observa en la Tabla 3.1, todos los tiempos de acceso a las funciones del sistema, cumplen los tiempos propuestos por la metodología bajo el operador **Button Press** [39].

**Tabla 3.1.** Tiempos de Referencia con la Metodología KLM GOMS vs Tiempo Real.

<b>Acción</b>	<b>Tiempo Real Prototipo (segundos)</b>	<b>Tiempo KLM GOMS (segundos)</b>
Ayuda	2.45	3
Computar	2.16	3
Ir al Paso 4	2.22	3
Ir al Paso 7	2.48	3
Editar Materia	2.11	3
Eliminar Materia	2.18	3
Exportar	2.05	3
Nueva Gestión	2.34	3
Ir a la Hoja de Cálculo	2.14	3
Cerrar	1.14	2

Segundo se valida, el tiempo total en realizar las diversas acciones (mostradas en la Tabla 3. 2). Se recalca que este tiempo no puede ser evaluado bajo ninguna metodología, pero demuestra, bajo temporizadores de código, que el sistema no satura al usuario en extensión y demora.

**Tabla 3. 2.** Tiempos de Referencia de Duración de las Funcionalidades del prototipo.

<b>Acción</b>	<b>Tiempo Real Prototipo (segundos)</b>
Crear Materia	29.31
Crear Rubricas	52.77
Subir Alumnos	15.17
Seleccionar Materia	7.18
Simular Hoja de Cálculo	4.72
Ingresar 1 Calificación	4.58
Computar Calificaciones	8.49
Descargar Excel	12.05

Posteriormente se encuentra que la longitud y velocidad del audio de ayuda para cada acción del sistema es un problema:




- No puede ser muy largo, dado que causa aburrimiento y la extensión para culminar tareas incrementa demasiado.
- No debe ser muy corto porque las personas invidentes necesitan una verdadera y completa retroalimentación por sus acciones.
- No debe ser muy lento porque causaría aburrimiento.
- No debe ser muy rápido porque causa confusión al usuario final.

Para satisfacer estas pautas se ha decidido usar oraciones de longitud promedio de 5 a 10 palabras. Las palabras con 3 o menos caracteres, han sido omitidas del conteo.

### 3.1.4 VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES







Las validaciones de los requerimientos funcionales se realizaron tomando como referencia la retroalimentación oral que se brindó una vez finalizada la ejecución de las pruebas del prototipo.

**Tabla 3.3.** Validación de Requerimientos Funcionales.



RF	Descripción	Estado		Observación
RF01	Creación de una Materia	Primera y Segunda Reunión		Los usuarios manifestaron conformidad al momento de crear una materia dado que cumple con los 4 parámetros que deben ingresar para crear.
RF02	Cargar Estudiantes en la Materia Creada	Primera y Segunda Reunión		Los usuarios manifestaron conformidad al momento Cargar un archivo de estudiantes dado que se les explicó que necesitarán una asistencia menor a 1 minuto para elegir el archivo al prototipo.
RF3	Creación Dinámica de Rúbricas	Primera y Segunda Reunión		Los usuarios manifestaron conformidad al momento crear rúbricas para el primer y segundo bimestre. El sintetizador de voz ayuda de gran manera al usuario a entender que es lo que debe hacer.



**Tabla 3.4.** Validación de Requerimientos Funcionales.

RF	Descripción	Estado		Observación
RF04	Simular Hoja de Cálculo	Primera y Segunda Reunión		Los usuarios manifestaron conformidad al momento simular la hoja de cálculo, ya que es el único paso que tenían como realizar. Los usuarios tienen un flujo lineal de las funcionalidades del prototipo.
RF05	Eliminar Materia	Primera y Segunda Reunión		Los usuarios manifestaron conformidad al momento de eliminar la materia. Esto debido a que se pide escoger la materia que se va a eliminar y con un enter se la elimina del prototipo. simular la hoja de cálculo,
RF06	Editar ciertos atributos de la Materia	Primera y Segunda Reunión		Los usuarios manifestaron conformidad moderada al momento de editar los atributos de la materia. Esto ocurrió porque no se puede editar todos los campos de la materia ya que generaría problemas con las funciones creadas para verificar el correcto funcionamiento del prototipo.
RF07	Computar Calificaciones	Primera y Segunda Reunión		Los usuarios manifestaron conformidad ya que para computar calificaciones solo se debe dar un enter y se computan las calificaciones automáticamente.
RF08	Asignar Calificaciones	Primera y Segunda Reunión		Los usuarios manifestaron conformidad moderada ya que para ingresar las calificaciones de todos los estudiantes se demoraron algún tiempo. Esto no hay forma de mejorar ya que depende del número de alumnos que existan en la materia y de las rúbricas.
RF09	Obtener Información General de la Materia	Primera y Segunda Reunión		Los usuarios manifestaron conformidad debido a que esta acción se la ejecuta de manera conjunto al momento de computar las calificaciones.

**Tabla 3.5.** Validación de Requerimientos Funcionales.

RF	Descripción	Estado		Observación
RF10	Exportar Información	Primera Reunión		Los usuarios manifestaron inconformidad, pues al momento de dar enter para descargar no sabían si se descargó la información.
		Segunda Reunión		Se corrigió añadiendo una guía de voz, para que al momento de descargar, brinde una retroalimentación de que el archivo ya se ha guardado.


La Tabla 3.3, Tabla 3.4 y Tabla 3.5 describen los procedimientos realizados para verificar el cumplimiento de los requerimientos planteados durante la Etapa de Diseño. La gran parte de requerimientos fueron satisfactorios con la excepción de uno, el cual se hizo las respectivas correcciones obtenidas de las recomendaciones de los usuarios de prueba.

### 3.1.5 VALIDACIÓN DE REQUERIMIENTOS NO FUNCIONALES






Los requerimientos no funcionales son los que favorecen la interacción con el usuario y dan una buena experiencia [29]. Al igual que los requerimientos funcionales, se validó mediante la retroalimentación oral por parte de los usuarios de prueba. En la Tabla 3.6 y

Tabla 3.7 se observa las validaciones y una corrección en el Flujo de las Guías de Voz que se implementó.

**Tabla 3.6.** Validación de Requerimientos No Funcionales.

RNF	Descripción	Estado		Observación
RNF01	Usabilidad del prototipo	Primera y Segunda Reunión		Los usuarios no manifestaron quejas con respecto a la usabilidad del proyecto. El tiempo que se demora en las guías de voz es una parte fundamental para este grupo de personas y están de acuerdo con estas guías.

**Tabla 3. 7.** Validación de Requerimientos No Funcionales.

RNF	Descripción	Estado		Observación
RNF02	Ejecución Lineal del prototipo	Primera y Segunda Reunión		Los usuarios manifestaron claridad al momento de utilizar el prototipo, los pasos numerados los ayuda a tener una idea clara en que sección se encuentran.
RNF03	Flujos de las Guías de Voz	Primera Reunión		Se probó 3 escenarios para las guías del flujo de voz, esto se lo hizo mediante el uso de instrucciones de voz de menos de 5 palabras, entre 5 y 10 palabras y más de 10 palabras. Al final los usuarios manifestaron que la instrucción entre 5 a 10 palabras, son más claras y evitan posibles errores.
		Segunda Reunión		Se corrigió las instrucciones para que en promedio sea de 5 a 10 palabras por instrucción. También se escogió una voz femenina con una vocalización moderada y entendible.
RNF04	Compatibilidad del Prototipo	Primera y Segunda Reunión		Los usuarios manifestaron satisfacción debido a que el 100% de usuarios de prueba utilizan sistema operativo Windows y este prototipo fue desarrollado para este sistema.
RNF05	Atajos de Teclado	Primera y Segunda Reunión		Los usuarios manifestaron satisfacción, esto después de las capacitaciones que se tuvo ya que se explicó cuáles son los atajos de teclado más utilizados y su facilidad para recordarlos.

### 3.1.6 CIERRE DEL TABLERO KANBAN

Finalmente, acabadas las pruebas en el prototipo y la depuración de los problemas encontradas durante su ejecución. Se procede a cerrar el tablero Kanban que se observa en la Figura 3.2, debido a que se culminó todas las tareas planteadas.

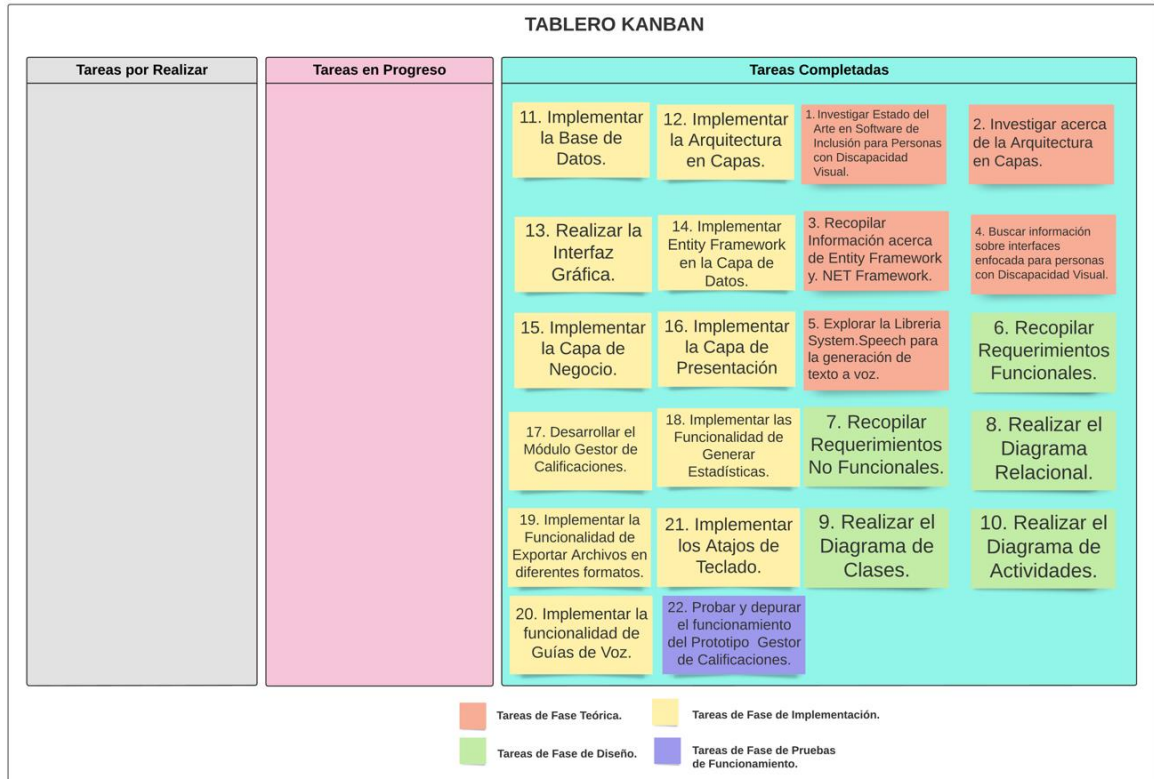


Figura 3.2. Finalización del tablero Kanban.

## 3.2 CONCLUSIONES Y RECOMENDACIONES

### 3.2.1 CONCLUSIONES

El prototipo fue desarrollado en base a cuatro etapas: La Fase Teórica, de Diseño, de Implementación y de Pruebas de Funcionamiento. A partir de este proceso realizado se desprenden las siguientes conclusiones.

- El estudio de las herramientas, lenguaje de programación y entornos utilizados en el desarrollo del presente trabajo como: la Programación Orientada a Objetos en C#, bases de datos de SQL Server, ORM utilizados para mapear la base de datos, la herramienta de Microsoft Visual Studio y la arquitectura en 3 capas facilitó el

desarrollo del prototipo cumpliendo los objetivos planteados y cumpliendo altos estándares de calidad.

- Generar guías de voz correctas, requiere un proceso ordenado de los datos que se quiere informar. Es importante comprender, que la información que se proporcione al usuario mediante estos traductores de voz debe ser concisa teniendo en cuenta el no sobrepasar el límite de palabras en las instrucciones ya que puede generar confusiones al momento de utilizar el prototipo. Por este motivo se utilizó las instrucciones necesarias para que el usuario no vidente no tenga dificultades al escuchar las guías de voz.
- El diseño lineal ayudó en la implementación del prototipo para que el profesor con discapacidad visual cometa la menor cantidad de errores, esto debido a que se limitó al usuario a seguir un orden específico al momento de utilizar el prototipo.
- Mediante el uso de la metodología KLM GOMS, se pudo encontrar los tiempos aproximados que se necesita para ejecutar una tarea y se validó mediante las pruebas realizadas que los tiempos para ejecutar las tareas del prototipo cumplen con los tiempos de la metodología propuesta.
- El segmento de personas con discapacidad visual es numeroso y no ha sido tomado en cuenta al diseñar aplicaciones en los últimos años (a pesar de los avances de la tecnología). La falta de software para la gestión de información basadas en hojas de cálculo para un profesor no vidente desembocó en el desarrollo del presente trabajo. Es fundamental que el desarrollo de aplicaciones hoy en día pueda ser utilizado por personas con o sin discapacidad visual. Mediante el desarrollo del prototipo Gestor de Calificaciones se ha logrado tener una herramienta para ayudar a un profesor no vidente a gestionar las calificaciones de sus alumnos, obtener información general de los alumnos y estadísticas de las rúbricas y calificaciones guiado en todo momento por traductores de texto a voz.

### **3.2.2 RECOMENDACIONES**

Durante las distintas fases del desarrollo del prototipo, se presentaron varios problemas que necesitaron soluciones de diversa índole, en base a las eventualidades encontradas se plantean las siguientes recomendaciones:

- Durante el desarrollo del prototipo, se recomienda diseñar una base de datos con los campos necesarios para el correcto funcionamiento del prototipo, esto debido a que en los primeros diseños que se realizó; se colocó información irrelevante en la

base de datos, haciendo que el prototipo se vuelva ineficiente para cumplir sus objetivos planteados.

- Existe la posibilidad de que se genere un problema con el idioma de las guías de voz, esto puede darse debido a que se trabaja con varios idiomas en el sistema operativo de Windows que se maneja, se recomienda colocar el idioma español como idioma predeterminado para así solventar este inconveniente.
- Es una buena práctica de desarrollo utilizar secciones que permita a los usuarios con discapacidad visual adaptarse fácilmente a los prototipos. De esta manera se dotará al usuario a tener una idea clara de donde se encuentra a lo largo del uso del prototipo.
- Durante el desarrollo del prototipo, se tuvo un problema al momento de simular la hoja de cálculo, esto debido a que al momento de navegar por esta hoja solo se podía avanzar de manera vertical. Para solucionar este problema se recomienda utilizar funciones que permitan al usuario la libre movilidad a través de toda la simulación de la hoja de cálculo.

### **3.2.3 TRABAJO FUTURO**

En base a la experiencia adquirida durante el desarrollo del prototipo se presentan recomendaciones enfocadas a desarrollos futuros sobre este prototipo o de proyectos similares.

- En el desarrollo del prototipo se tomó un traductor de texto a voz con una velocidad de voz normal. Sin embargo, el usuario no vidente al familiarizarse con el uso del prototipo, las instrucciones de voz pueden ser muy repetitivas y monótonas. Por lo que se recomienda incluir un nuevo módulo para modificar parámetros relativos al traductor de voz como la velocidad del audio permitiendo al usuario decidir cuál es la velocidad que se adapta a sus necesidades.
- En el desarrollo del prototipo realizado, se diseñó una interfaz en español, limitando el acceso a otras personas cuyo idioma nativo sea diferente al español. Este prototipo puede ser usado por diferentes personas con idiomas natales distintos. Siendo el inglés un idioma universal se recomienda que exista la posibilidad de permitir al usuario escoger el idioma en el cual desee que funcione el prototipo; ya sea este en inglés o español.
- Al subir la lista de estudiantes al prototipo para gestionar las calificaciones existe una limitante, pues este archivo de estudiantes se rige a un formato dado. En un

futuro debería cambiarse con la finalidad de que se pueda incluir cualquier formato al prototipo.

- En el desarrollo del prototipo realizado, se diseñó una interfaz para que acceda solo un profesor no vidente, limitando la opción de crear más usuarios. En un futuro se puede crear un módulo donde se pueda añadir a más profesores no videntes dentro del prototipo, para que así cada profesor gestione las materias que tienen asignados.

## 4 REFERENCIAS BIBLIOGRÁFICA

- [1] A. Aurum, H. Petersson, y C. Wohlin, “State-of-the-art: Software inspections after 25 years”, *Softw. Test. Verification Reliab.*, vol. 12, pp. 133–154, sep. 2002, doi: 10.1002/stvr.243.
- [2] F. A. Vásquez, “Programación por capas”, *Paradigmas*, vol. 2, núm. 1, Art. núm. 1, 2010, doi: 10.31381/paradigmas.v2i1.1498.
- [3] B. John y D. Kieras, “Using GOMS for User Interface Design and Evaluation: Which Technique?”, *ACM Trans. Comput.-Hum. Interact. TOCHI*, vol. 3, dic. 1999, doi: 10.1145/235833.236050.
- [4] T. Murino, G. Naviglio, y E. Romano, “Optimal size of kanban board in a single stage multi product system”, *WSEAS Trans. Syst. Control*, vol. 5, pp. 464–473, jun. 2010.
- [5] H. Herchi y W. B. Abdessalem, “From user requirements to UML class diagram”. arXiv, el 4 de noviembre de 2012. doi: 10.48550/arXiv.1211.0713.
- [6] “Microsoft SQL Server”, p. 6.
- [7] A. Hejlsberg, M. Torgersen, S. Wiltamuth, y P. Golde, *The C# Programming Language*. Pearson Education, 2008.
- [8] “User Interfaces in C#: Windows Forms and Custom Controls by MacDonald M.” <https://flylib.com/books/en/2.742.1/> (consultado el 21 de junio de 2022).
- [9] H. Burton, “Visual Cortex Activity in Early and Late Blind People”, *J. Neurosci.*, vol. 23, núm. 10, pp. 4005–4011, may 2003, doi: 10.1523/JNEUROSCI.23-10-04005.2003.
- [10] A. D. Rice y J. W. Lartigue, “Touch-level model (TLM): evolving KLM-GOMS for touchscreen and mobile devices”, en *Proceedings of the 2014 ACM Southeast Regional Conference*, Kennesaw Georgia, mar. 2014, pp. 1–6. doi: 10.1145/2638404.2638532.
- [11] “FuturoDiscapacidad.pdf”. Consultado: el 21 de junio de 2022. [En línea]. Disponible en: <https://www.cermi.es/sites/default/files/docs/colecciones/FuturoDiscapacidad.pdf>
- [12] A. L. Esparza-Maldonado, L. Y. Margain-Fuentes, F. J. Álvarez-Rodríguez, y E. I. Benítez-Guerrero, “Desarrollo y evaluación de un sistema interactivo para personas con discapacidad visual”, *TecnoLógicas*, vol. 21, núm. 41, pp. 149–157, ene. 2018, doi: 10.22430/22565337.733.
- [13] W. M. Consuegra, R. G. Peña, y M. Y. E. Raimundo, “PARTICULARIDADES DEL DISEÑO, DESARROLLO Y EVALUACIÓN DE SOFTWARE EDUCATIVO ACCESIBLE PARA PERSONAS CIEGAS”, *Rev. Varela*, vol. 7, núm. 18, Art. núm. 18, sep. 2007.
- [14] J. Falgueras, A. Carrillo León, y A. Guevara, “Enfoques en el estudio de las interfaces de usuario”, p. 60, sep. 2002.
- [15] “Predicting Human Performance Differences on Multiple Interface Alternatives: KLM, GOMS and CogTool are Unreliable | Elsevier Enhanced Reader”.

- <https://reader.elsevier.com/reader/sd/pii/S2351978915008070?token=EC9CF088D278B81A23417BCFC987519FA3D6FB278F11CAE5C452D71668C88236396915BCA667CE6F1DC8DB2EA8EB2A53&originRegion=us-east-1&originCreation=20220622030438> (consultado el 21 de junio de 2022).
- [16] D. Cunha, R. P. Duarte, y C. A. Cunha, “KLM-GOMS Detection of Interaction Patterns Through the Execution of Unplanned Tasks”, en *Computational Science and Its Applications – ICCSA 2021*, Cham, 2021, pp. 203–219. doi: 10.1007/978-3-030-86960-1\_15.
- [17] R.-U. Modrich y B. C. Cousins, “Digital Kanban boards used in design and 3D coordination”, jul. 2017. Consultado: el 21 de junio de 2022. [En línea]. Disponible en: <https://apo.org.au/node/304963>
- [18] D. J. Powell, “Kanban for Lean Production in High Mix, Low Volume Environments”, *IFAC-Pap.*, vol. 51, núm. 11, pp. 140–143, ene. 2018, doi: 10.1016/j.ifacol.2018.08.248.
- [19] C. de, “Característica de la Programación por Capas”, p. 9.
- [20] K. W. D. Sandoval, “Implementación del módulo de Compras del Sistema KOHINOR DBS utilizando Visual Studio y ASP.NET”, p. 43, 2021.
- [21] P. Nielsen y U. Parui, *Microsoft SQL Server 2008 Bible*. John Wiley & Sons, 2011.
- [22] L. Powers y M. Snell, *Microsoft Visual Studio 2008 Unleashed*. Pearson Education, 2008.
- [23] T. Thai y H. Lam, *.NET Framework Essentials*. O’Reilly Media, Inc., 2003.
- [24] B. G. Al-Bastami y S. S. A. Naser, “Design and Development of an Intelligent Tutoring System for C# Language”, *Eur. Acad. Res.*, vol. 4, núm. 10, 2017.
- [25] I. Intecca y A. C. C. Delfa, “Aprende a programar con C# (programación orientada a objetos)”, ene. 2022, Consultado: el 22 de mayo de 2022. [En línea]. Disponible en: <http://contenidosdigitales.uned.es/fez/view/intecca:GICCU-61604ed4397ab4e76f3ec7ac>
- [26] J. Lerman, *Programming Entity Framework: Building Data Centric Apps with the ADO.NET Entity Framework*. O’Reilly Media, Inc., 2010.
- [27] E. A. C. Espinosa, “SINTETIZADOR DE VOZ DESCRIPCION GENERAL Y APLICACIONES EN LA MEDICINA”, p. 6, 2014.
- [28] “Where teamwork and ideas”, *Lucidspark*. <https://lucidspark.com/landing> (consultado el 22 de mayo de 2022).
- [29] “66612870011.pdf”. Consultado: el 9 de junio de 2022. [En línea]. Disponible en: <https://www.redalyc.org/pdf/666/66612870011.pdf>
- [30] I. G. Quiroz y J. D. P. Pinedo, “RUP y UML: Un estudio sobre ¿qué es?, ¿para qué se usa? y ¿cómo se desarrolla? Un Diagrama de Caso de Uso”, p. 12.
- [31] “Intelligent Diagramming | Lucidchart”. <https://www.lucidchart.com/pages/> (consultado el 29 de mayo de 2022).
- [32] C. M. Ricardo, *Bases de datos*. McGraw Hill Educación, 2009. Consultado: el 21 de junio de 2022. [En línea]. Disponible en: <http://up-rid2.up.ac.pa:8080/xmlui/handle/123456789/1354>
- [33] C. L. Vidal, S. E. Rivero, L. P. López, y C. A. Pereira, “Propuesta y Aplicación de Diagramas de Clases UML JPI”, *Inf. Tecnológica*, vol. 25, núm. 5, pp. 113–120, 2014, doi: 10.4067/S0718-07642014000500016.
- [34] J. M. O. Cassá y P. A. O. D. Ufano, *Programación web en Java*. Ministerio de Educación, 2012.
- [35] M. C. Albornoz, “Diseño de interfaz gráfica de usuario”, presentado en XVI Workshop de Investigadores en Ciencias de la Computación, oct. 2014. Consultado: el 21 de junio de 2022. [En línea]. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/41578>
- [36] S. Tak, P. Westendorp, y I. van Rooij, “Satisficing and the Use of Keyboard Shortcuts: Being Good Enough Is Enough?”, *Interact. Comput.*, vol. 25, núm. 5, pp. 404–416, sep. 2013, doi: 10.1093/iwc/iwt016.



- [37] S. Rojbi y A. Rojbi, “KeybNav: A new system for web navigation through a keyboard”, en *2017 6th International Conference on Information and Communication Technology and Accessibility (ICTA)*, dic. 2017, pp. 1–2. doi: 10.1109/ICTA.2017.8336057.
- [38] L. B. C. Usinia, “Análisis de las Reformas Institucionales del Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior (CEAACES) en el Ecuador”, p. 194.
- [39] D. Cunha, R. P. Duarte, y C. A. Cunha, “KLM-GOMS Detection of Interaction Patterns Through the Execution of Unplanned Tasks”, en *Computational Science and Its Applications – ICCSA 2021*, Cham, 2021, pp. 203–219. doi: 10.1007/978-3-030-86960-1\_15.

## 5 ANEXOS

Los anexos adjuntados como parte complementaria al desarrollo del presente Trabajo de Integración Curricular se observan a continuación:

ANEXO I. Glosario

ANEXO II. Código del prototipo Gestor de Calificaciones.

ANEXO III. Código del prototipo Gestor de Calificaciones.

## ANEXO I. Glosario

---

### **B**

#### **Button Press**

Operador de la metodología GOMS que indica el tiempo de ejecución de la pulsación de una tecla. · 43

---

### **C**

#### **C#**

Es un lenguaje de programación orientado a objetos. · IX

#### **CLR**

Es un entorno de ejecución para los códigos de los programas que corren sobre la plataforma Microsoft .NET. · 16

---

### **E**

#### **EF**

Es un ORM que permite a los desarrolladores convertir sus estructuras de datos en clases para poder trabajar con esa información usando objetos de .NET. · 16

---

### **G**

#### **GOMS**

Es un modelo de procesador de información especializado para la observación de la interacción persona-computadora. · VII

---

### **I**

#### **IDE**

Es una colección de herramientas e instalaciones que los programadores necesitan para el desarrollo y diseño de software. · 16

---

### **K**

#### **KLM**

Es la abreviatura de Keystroke-level model, este modelo predice cuánto tiempo tomará a un usuario realizar una tarea mediante la pulsación de una tecla. · VII

---

### **O**

#### **ORM**

Es un modelo de programación que permite mapear las estructuras de una base de datos. · 16

---

### **T**

#### **TTS**

Es la abreviatura de Text to Speech, una tecnología que convierte texto en voz. Se la puede utilizar para ayudar a personas con discapacidad visual. · VII

## **ANEXO II. Formato de la lista de estudiantes**

En el siguiente enlace se puede descargar el Formato obtenido del SII-Académico. A continuación, se muestra el archivo.

[Formato SII.xlsx](#)

### **ANEXO III. Código del prototipo**

El código del prototipo Gestor de Calificaciones está disponible en el siguiente enlace.

[Código Gestor de Calificaciones](#)