

APP YOUR DATABASE

1.- Definiciones y especificaciones de Requerimientos

a) Definición general del proyecto de software:

El proyecto tiene como objetivo principal la migración de una base de datos gestionada por MySQL a un nuevo sistema de gestión de bases de datos (DBMS) utilizando SQL Server. Este proceso de migración incluye la conversión de esquemas, datos y estructuras de la base de datos, asegurando la integridad de los datos y la funcionalidad del sistema después de la migración. Además, se desarrollará una interfaz de usuario que permita gestionar la base de datos en SQL Server de forma eficiente y rápida.

El sistema tendrá como propósito la mejora de la gestión y administración de la base de datos, aprovechando las características avanzadas de SQL Server como la escalabilidad, seguridad, y la integración con otras herramientas del ecosistema Microsoft.

Los objetivos del desarrollo incluyen:

- El sistema nuevo debe reemplazar el sistema actual, siendo el sistema nuevo más rápido, confiable y escalable.
- Realizar la migración de las tablas, vistas, procedimientos almacenados, índices y relaciones desde MySQL a SQL Server.
- Crear una interfaz de usuario amigable e intuitiva que facilite la administración y manejo de la base de datos SQL Server.
- Optimizar el rendimiento de consultas y transacciones en SQL Server. Garantizar que no haya pérdida de datos ni fallos en la funcionalidad del sistema durante y después de la migración.

b) Especificación de requerimientos del proyecto:

Los requerimientos técnicos y generales para la migración de la base de datos y la creación de la interfaz de gestión son los siguientes:

Requerimientos funcionales:

Migración de datos: El sistema debe migrar toda la información contenida en las bases de datos de MySQL a SQL Server sin pérdida de datos.

Migración de estructuras: Deben migrarse tablas, claves primarias y foráneas, índices, procedimientos almacenados, y otros objetos de la base de datos de MySQL a SQL Server.

Interfaz de usuario: El sistema debe contar con una interfaz de usuario gráfica (GUI) que permita la administración de la base de datos en SQL Server, incluyendo operaciones como la consulta, inserción, actualización, y eliminación de registros.

c) Procedimientos de instalación y prueba:

Herramientas utilizadas:

Python: Lenguaje de programación utilizado para desarrollar la interfaz gráfica y gestionar la conexión con la base de datos SQL Server.

MySQL: Sistema de gestión de base de datos de origen para la migración de datos hacia SQL Server.

SQL Server: Sistema de gestión de base de datos de destino para almacenar los datos migrados y manejar las operaciones a través de la interfaz.

Tkinter: Biblioteca de Python utilizada para crear la interfaz gráfica de usuario (GUI) en la aplicación.

Treeview: Widget de Tkinter utilizado para visualizar y administrar los datos en una estructura tipo árbol dentro de la interfaz gráfica.

Pyodbc: Biblioteca de Python para la conexión con bases de datos SQL Server, permitiendo la ejecución de consultas y la manipulación de datos.

Pymysql: Biblioteca de Python para la conexión con la BD de datos Mysql Server, permitiendo la ejecución de consultas y la manipulación de datos (Esta se utiliza en el proceso de migración de datos).

Procedimiento de Desarrollo:

El procedimiento de desarrollo de este proyecto de migración se divide en varias fases, asegurando que la transición de MySQL a SQL Server sea fluida y que la interfaz de gestión funcione correctamente con SQL Server.

Fase 1: Planificación y análisis

1. Análisis de la base de datos existente (MySQL):
 - a. Revisión del esquema de la base de datos de MySQL, incluyendo tablas, relaciones, claves primarias y foráneas, índices, vistas, procedimientos almacenados y triggers.
 - b. Identificación de posibles incompatibilidades entre MySQL y SQL Server que puedan afectar a la migración (por ejemplo, tipos de datos no soportados, funciones específicas de MySQL que no tengan equivalente directo en SQL Server).
2. Especificación de Requisitos:
 - a. Definir los requisitos funcionales y no funcionales del proyecto (ya cubiertos en la sección anterior).
 - b. Identificar las herramientas y tecnologías necesarias para el desarrollo de la interfaz de usuario y la migración de datos (por ejemplo, lenguajes de programación, frameworks de interfaz gráfica).

Fase 2: Diseño

1. Diseño del esquema de la base de datos en SQL Server:

- a. Crear un diagrama de la base de datos en SQL Server con las mismas tablas, relaciones y restricciones que en la base de datos MySQL.
 - b. Ajustar el diseño para que aproveche las características y capacidades de SQL Server, como la optimización de índices o el uso de procedimientos almacenados específicos de SQL Server.
2. Diseño de la interfaz de usuario:
 - a. Especificar el diseño de la interfaz gráfica de usuario (GUI) para la administración de la base de datos en SQL Server.
 - b. Definir las funcionalidades que la interfaz debe soportar, como la conexión a la base de datos, la ejecución de consultas, la gestión de usuarios, la visualización y modificación de datos, entre otros.

Fase 3: Desarrollo

1. Desarrollo de la migración de datos:
 - a. Escribir scripts de migración: Utilizar herramientas de migración o escribir scripts de PYTHON que transformen el esquema de MySQL a SQL Server, adaptando los tipos de datos y estructuras que no son directamente compatibles.
 - b. Migración de datos: Desarrollar un proceso para migrar los datos desde MySQL a SQL Server, usando scripts de PYTHON y verificando la integridad de los datos migrados.
2. Desarrollo de la interfaz de gestión:
 - a. Utilizar un lenguaje de programación adecuado (En este caso se usó Python con Tkinter) para desarrollar la interfaz de usuario que interactúe con la base de datos SQL Server.
 - b. Crear las funcionalidades básicas de la interfaz, como la conexión a la base de datos, la ejecución de consultas SQL, la visualización de resultados, y la gestión de datos a través de formularios.

Fase 4: Integración

1. Integración de la base de datos con la interfaz de usuario:

- a. Asegurarse de que la interfaz de usuario esté correctamente conectada a SQL Server.
 - b. Realizar pruebas de integración para asegurarse de que la interfaz de usuario pueda realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) correctamente en la base de datos SQL Server.
2. Optimización de rendimiento:
 - a. Evaluar el rendimiento de la interfaz y la base de datos, asegurándose de que las consultas se ejecuten de manera eficiente y que la interfaz sea reactiva.

Fase 5: Pruebas

1. Pruebas unitarias:
 - a. Desarrollar pruebas unitarias para las funciones individuales tanto de la migración de datos como de la interfaz de usuario.
 - b. Asegurarse de que cada componente funcione correctamente por separado (por ejemplo, que los scripts de migración transformen correctamente las tablas y que los formularios de la interfaz gestionen los datos adecuadamente).
2. Pruebas de integración:
 - a. Realizar pruebas en conjunto de la base de datos y la interfaz de usuario, validando que ambas partes del sistema trabajen juntas sin problemas.
 - b. Probar la interacción entre la interfaz de usuario y SQL Server, verificando que se ejecuten correctamente las operaciones CRUD y las consultas SQL.

Fase 6: Implementación

1. Despliegue de la migración de base de datos:
 - a. Desplegar los scripts de migración, asegurándose de que la migración no interrumpa el funcionamiento de las aplicaciones dependientes.
 - b. Verificar que los datos hayan sido migrados correctamente y que la base de datos en SQL Server esté operativa.

2. Despliegue de la interfaz de usuario:
 - a. Instalar y configurar la interfaz de usuario en las máquinas cliente.
 - b. Asegurarse de que la interfaz esté correctamente conectada a la base de datos SQL Server y que los usuarios puedan acceder y administrar la base de datos sin problemas.

Fase 7: Mantenimiento

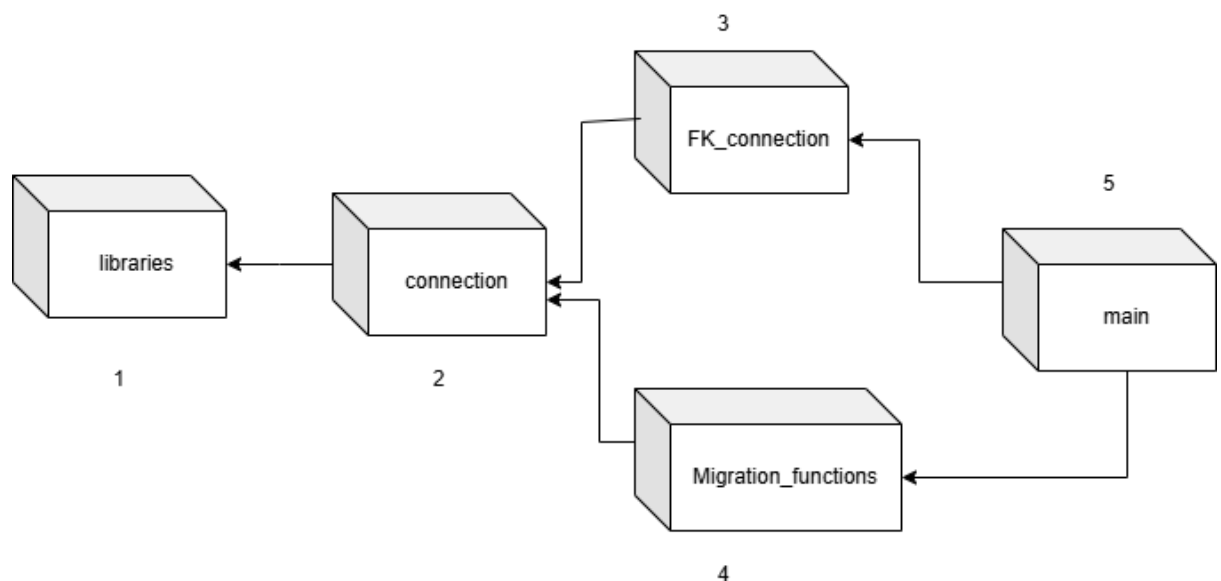
1. Monitoreo y soporte:
 - a. Realizar un monitoreo continuo del rendimiento de la interfaz y la base de datos en SQL Server, identificando y corrigiendo posibles cuellos de botella o fallos.

2.- Arquitectura del Sistema

La fase de desarrollo se divide en 2 partes: **Migración de datos** y **Desarrollo de una interfaz de gestión**.

1.- Migración de datos:

- **Diagrama de módulos:**



Descripción individual de los módulos:

Módulo 1: Libraries

Este módulo agrupa e importa todas las librerías necesarias para la operación del sistema. Es la base para los demás módulos, ya que proporciona las herramientas clave para realizar las tareas de conexión y migración. Las librerías utilizadas incluyen:

- **pyodbc:** Permite conectar y ejecutar consultas en diversas bases de datos usando el protocolo ODBC. Soporta sistemas como SQL Server, MySQL y PostgreSQL, entre otros.
- **pymysql:** Especializada para trabajar con bases de datos MySQL y MariaDB, esta librería facilita la ejecución de consultas y la gestión de conexiones de forma eficiente.

Este módulo centraliza la gestión de las dependencias del proyecto, asegurando que todas las herramientas estén disponibles para el correcto funcionamiento del sistema.

Módulo 2: Connection

Este módulo depende del **Módulo 1: Libraries**, ya que utiliza las librerías importadas para establecer conexiones a las bases de datos. Su principal funcionalidad es:

- **Establecer conexiones seguras y configurables** con bases de datos MySQL y SQL Server.
- **Parámetros configurables:** Soporta la personalización de parámetros como nombres de host, puertos, usuarios y contraseñas, garantizando flexibilidad en entornos de desarrollo y producción.

Este módulo abstrae los detalles técnicos de las conexiones, proporcionando una interfaz uniforme para interactuar con las bases de datos.

Módulo 3: FK_Connections

Este módulo depende del **Módulo 2: Connection**, ya que requiere las conexiones establecidas para interactuar con las bases de datos. Su principal objetivo es gestionar las relaciones entre tablas a través de las claves foráneas (Foreign Keys). Las funcionalidades incluyen:

- **Creación y gestión de claves foráneas:** Configura relaciones entre tablas para mantener la integridad referencial de los datos.
- **Verificación de restricciones:** Asegura que las tablas cumplan con las dependencias definidas antes de realizar migraciones o inserciones.
- **Documentación de relaciones:** Registra los vínculos establecidos, lo que facilita el mantenimiento y futuras migraciones.

Este módulo es esencial para preservar la estructura lógica de las bases de datos durante el proceso de migración.

Módulo 4: Migration_Functions

Este módulo depende del **Módulo 2: Connection**, ya que requiere acceso a las bases de datos para realizar la migración de datos. Sus responsabilidades principales son:

- **Creación de tablas:** Genera la estructura de las tablas en el sistema de destino (SQL Server), asegurando que sean equivalentes a las del sistema de origen (MySQL).
- **Inserción de datos:** Transfiere registros desde MySQL hacia SQL Server, respetando los formatos y restricciones definidos.
- **Gestión de datos transformados:** Soporta la transformación de datos cuando las estructuras de las bases de datos no son idénticas.

Este módulo actúa como el núcleo del sistema, encargándose de la transferencia efectiva de datos entre plataformas.

Módulo 5: Main

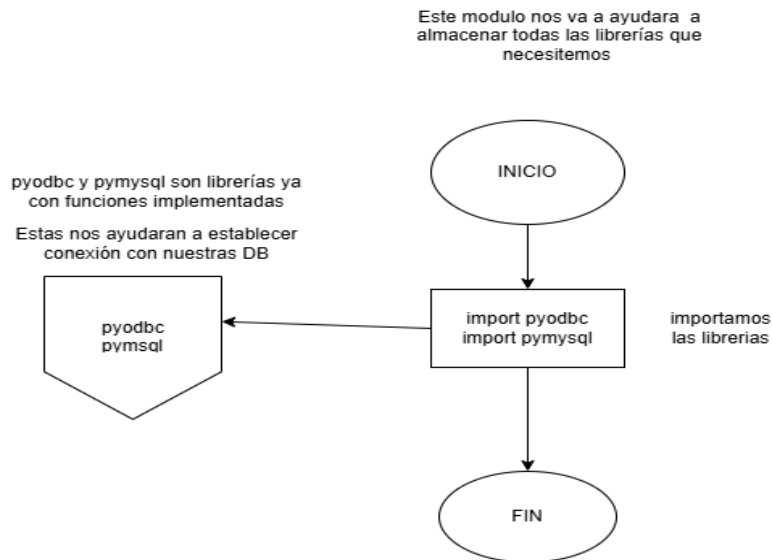
Este módulo depende de los módulos **FK_Connections** y **Migration_Functions**, ya que integra sus funcionalidades para ejecutar el proceso de migración completo. Sus características incluyen:

- **Orquestación del flujo de trabajo:** Llama a las funciones principales de los módulos importados para realizar la migración de datos y establecer las claves foráneas.
- **Validación y manejo de errores:** Implementa bloques `try-except` para capturar excepciones y notificar sobre posibles fallos durante la ejecución.
- **Interfaz principal del sistema:** Centraliza la ejecución de las tareas, ofreciendo un punto de entrada único para los usuarios.

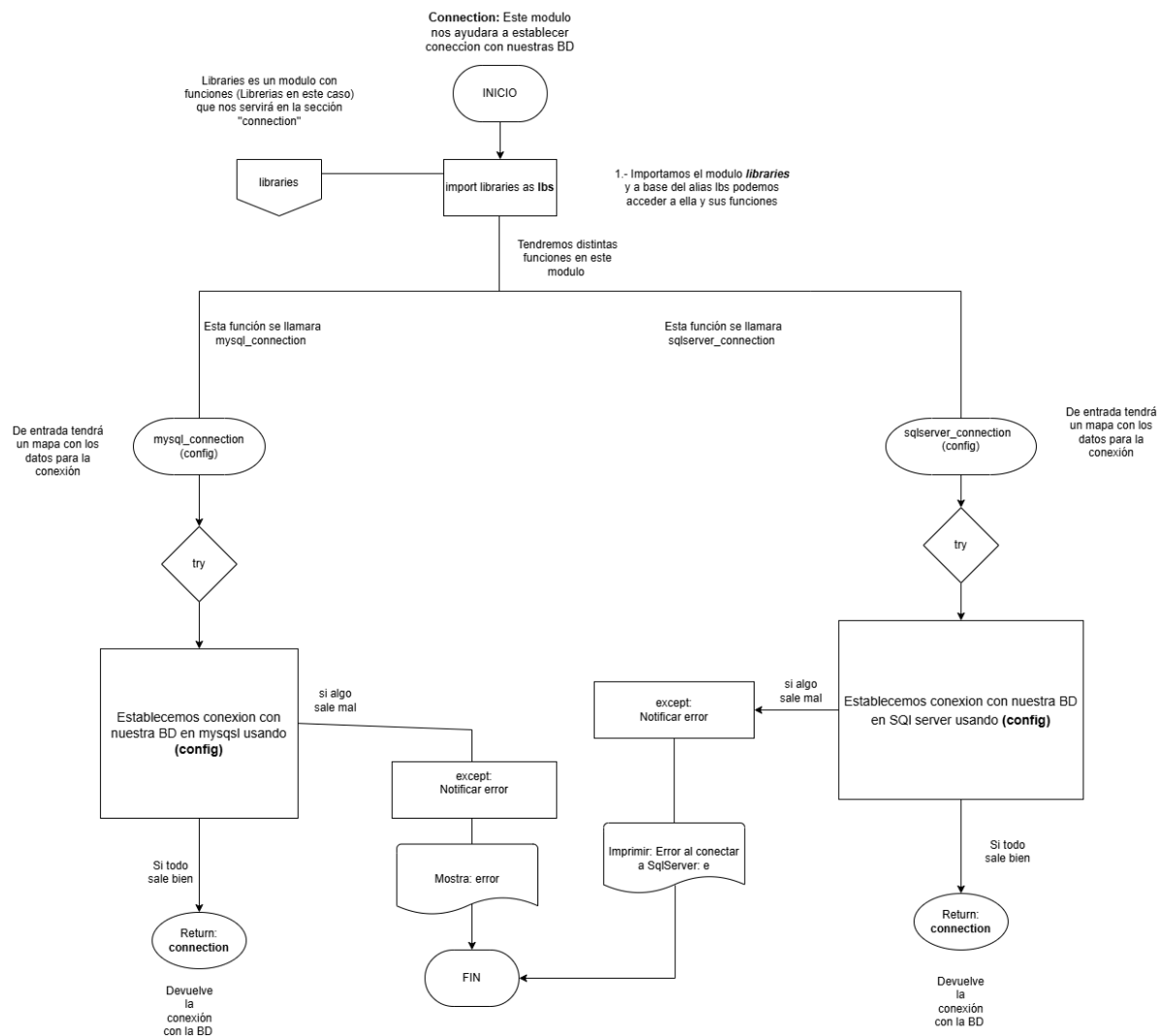
En resumen, este módulo es el controlador que integra todas las piezas del sistema, garantizando que la migración se lleve a cabo de manera eficiente y sin interrupciones.

Diagramas de flujo:

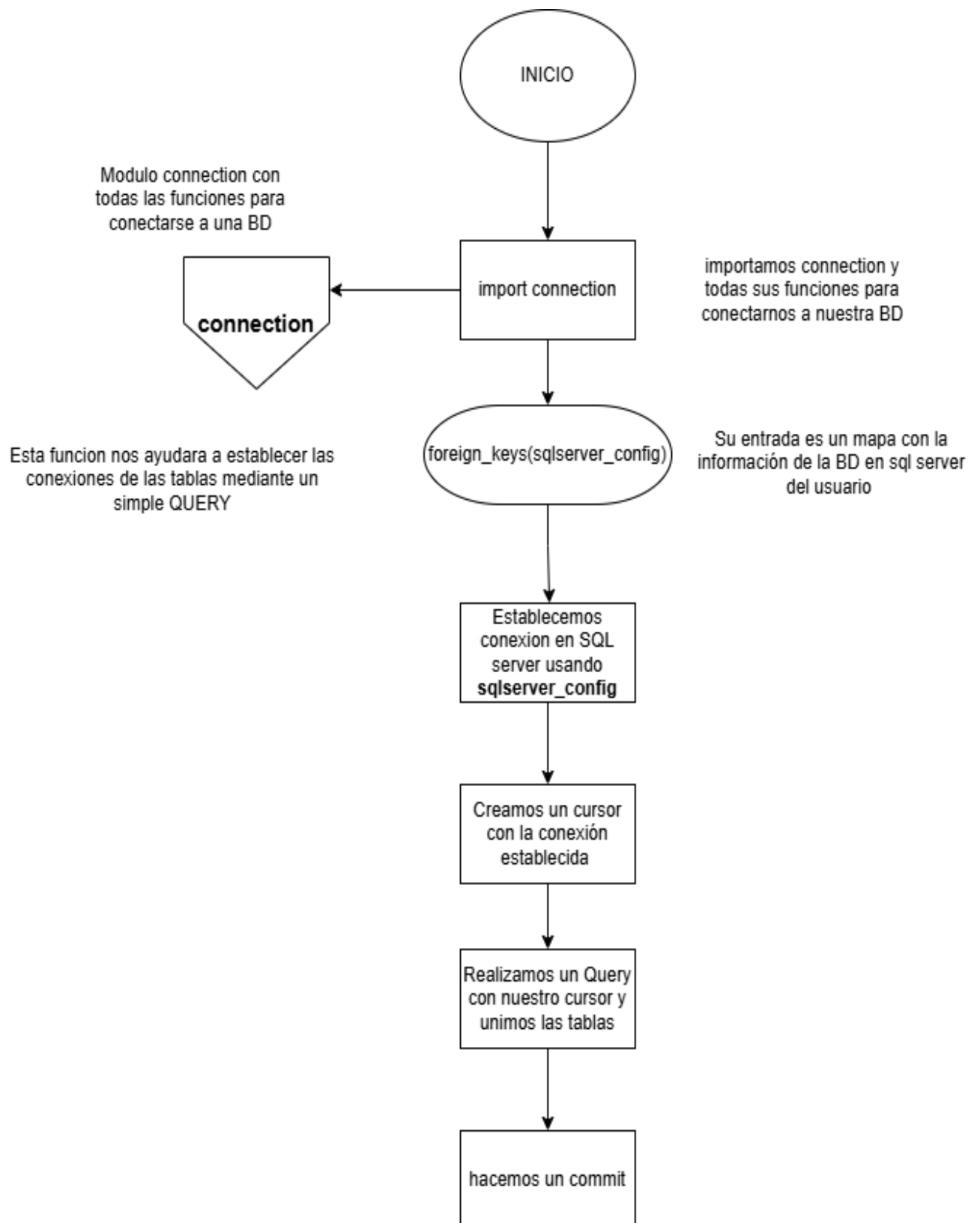
Módulo 1: libraries



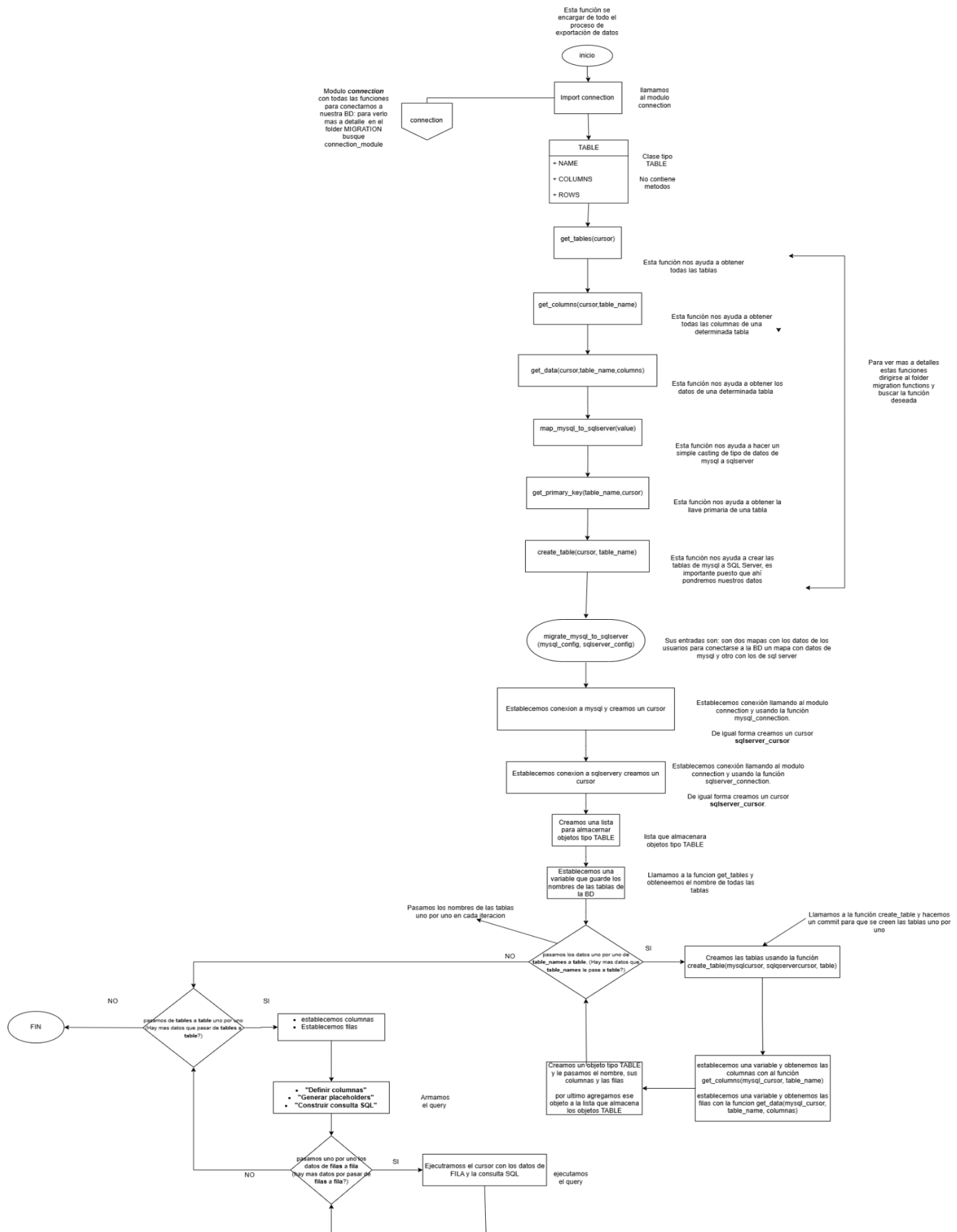
Modulo 2: Connection



Modulo 3 : FK_connections

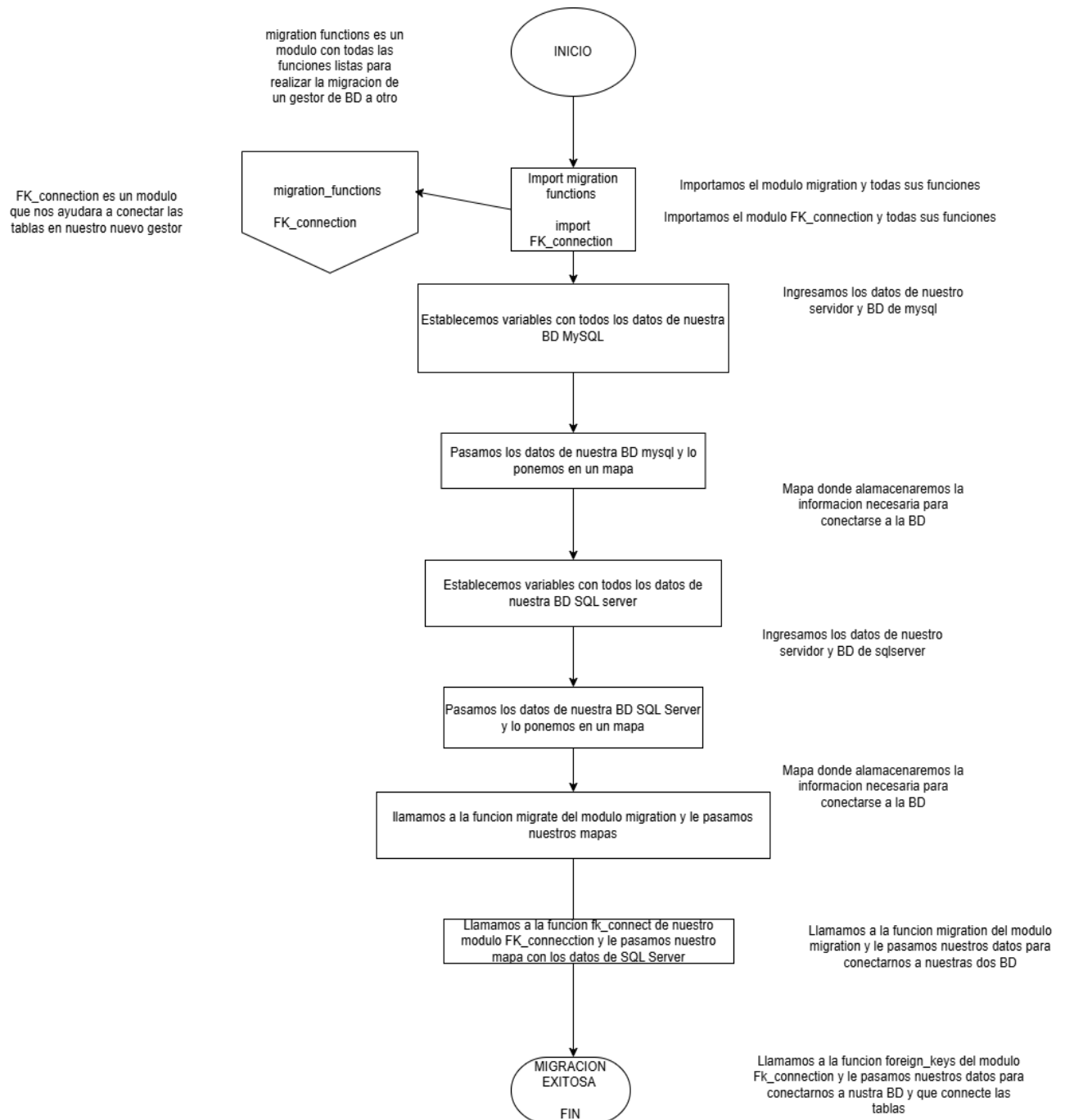


Modulo 4: Migration_functions:



nota: para ver más a detalle el módulo migration function, dirigirse al archivo Diagramas y luego a MIGRATION, ahí encontrará el diagrama y sus otras funciones más a detalle.

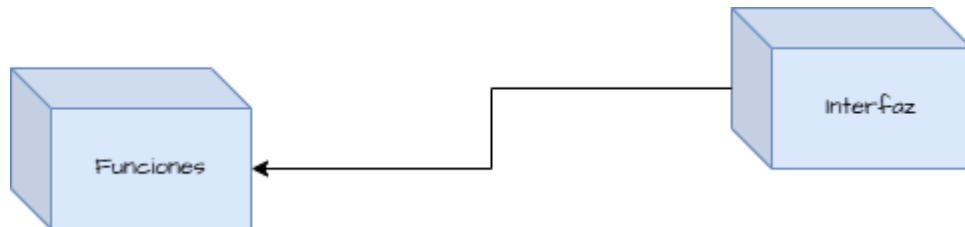
Módulo 5: main



Nota: Se pueden apreciar de mejor manera los diagramas en el archivo DOCUMENTACIÓN -> DIAGRAMAS -> MIGRATION

2.- Desarrollo de una interfaz gráfica:

Diagrama de Módulos:



Descripción individual de los módulos:

Módulo de Interfaz (Interfaz)

1. **Descripción:** Este módulo se encarga de la creación de la interfaz gráfica de usuario (GUI) usando **Tkinter** y su funcionalidad. Permite al usuario gestionar las tablas de la base de datos SQL Server de manera visual, con opciones para mostrar, agregar, modificar y eliminar registros. Utiliza el widget **Treeview** para mostrar los datos de las tablas en una estructura tipo árbol.
2. **Responsabilidad:**
 - a. Crear ventanas y controles para que el usuario interactúe con el sistema (como botones, formularios, etc.).
 - b. Mostrar los datos obtenidos de SQL Server en la interfaz gráfica.
 - c. Recoger entradas del usuario y enviarlas al módulo de funciones para su procesamiento.
 - d. Recoger entradas del usuario y enviarlas al módulo de funciones para su procesamiento.
 - e. Manejar validaciones de entrada para garantizar que los datos ingresados por el usuario sean correctos y seguros.
3. **Dependencias:**
 - a. **Tkinter:** Biblioteca para la creación de la interfaz gráfica.
 - b. **Treeview:** Widget de Tkinter para visualizar los datos en forma de tabla.

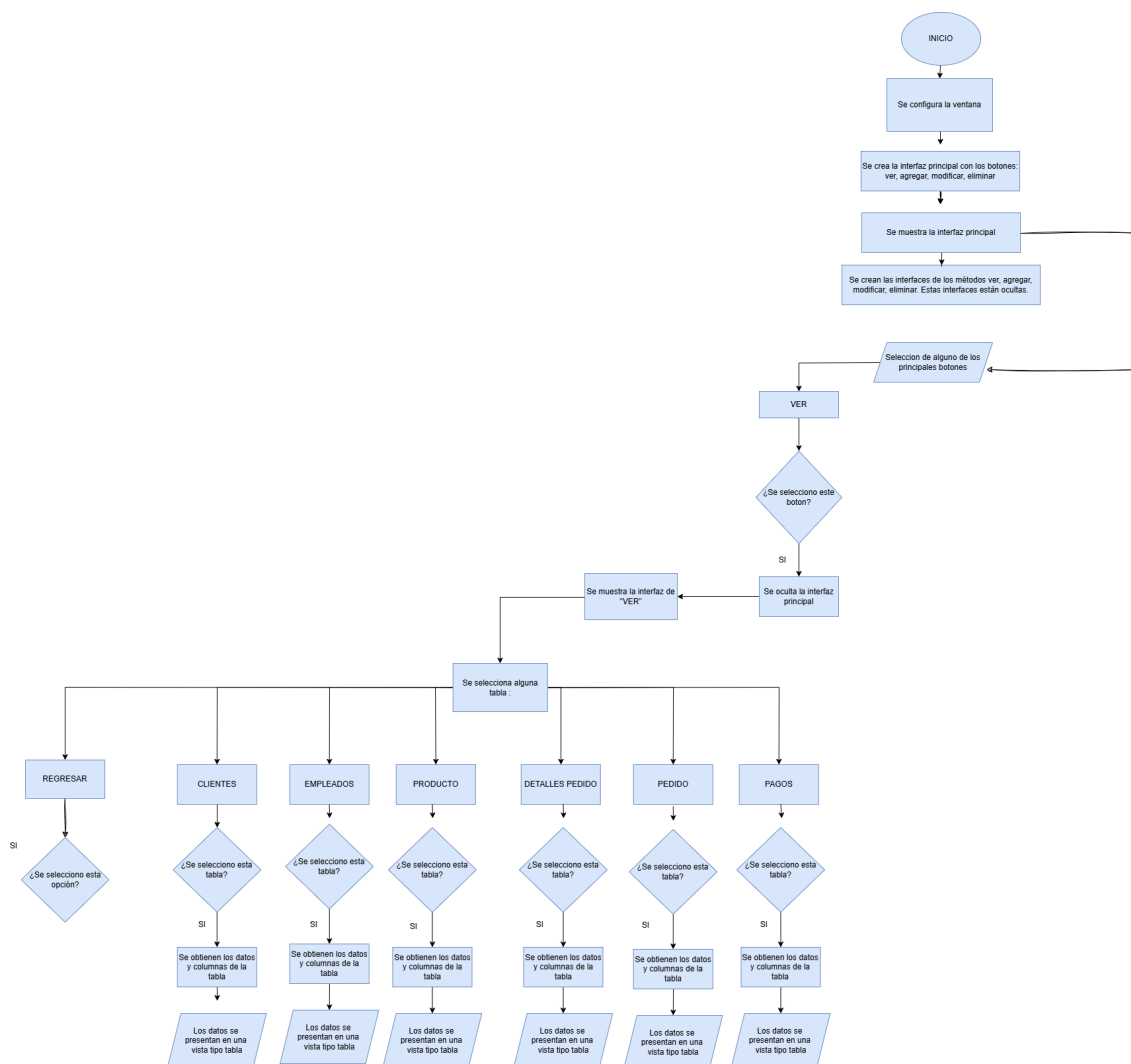
c. **Pyodbc**: Para la conexión a SQL Server y la obtención de los datos de la base de datos.

4. **Implementación**: La implementación de este módulo se basa en la creación de ventanas con **Tkinter**, donde se colocan los controles necesarios (botones, tablas, campos de texto) para que el usuario pueda interactuar con la base de datos. El acceso a los datos se realiza a través del módulo de funciones, que envía consultas SQL para mostrar la información.

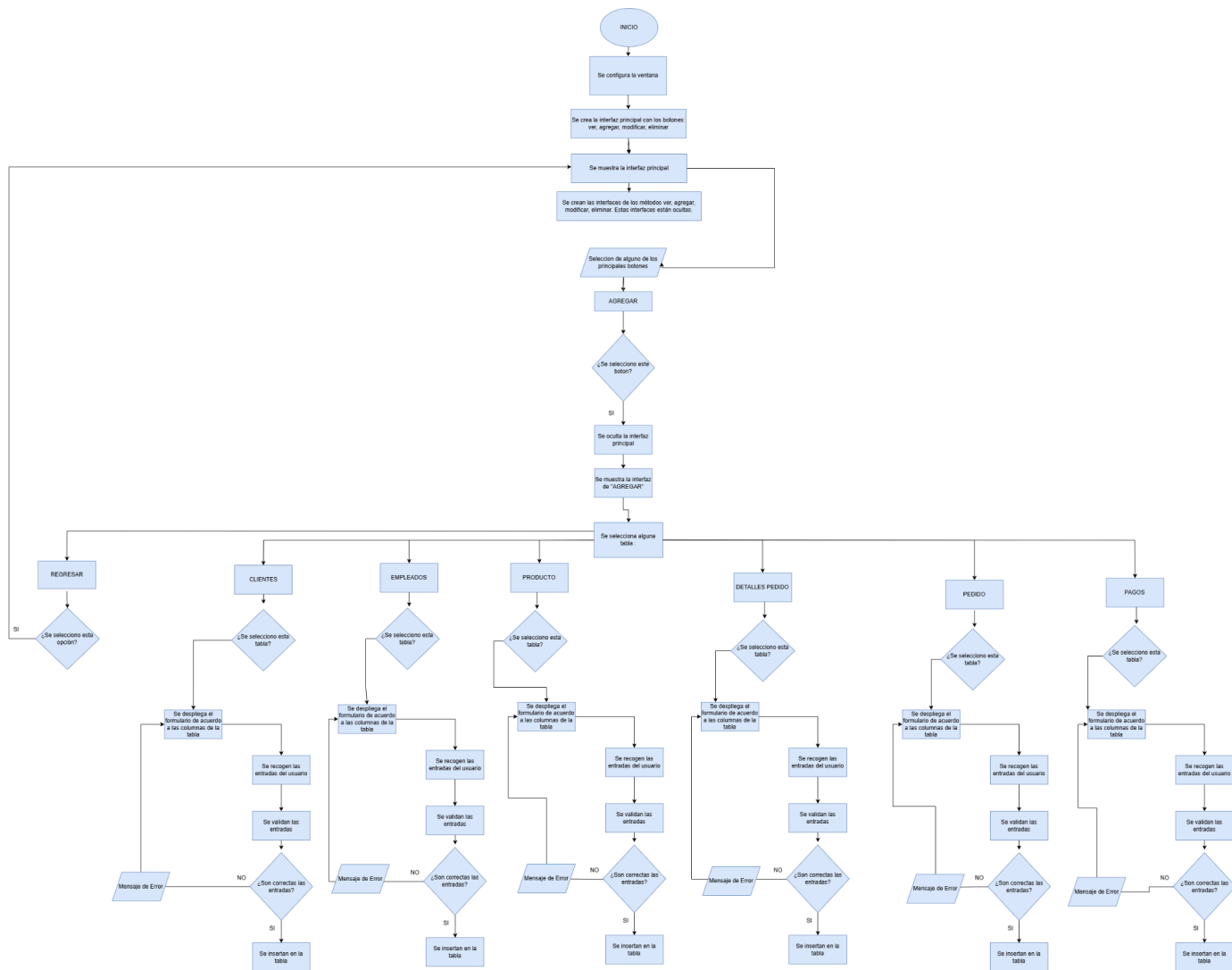
Diagrama de flujo del Módulo(Interfaz)

Partes:

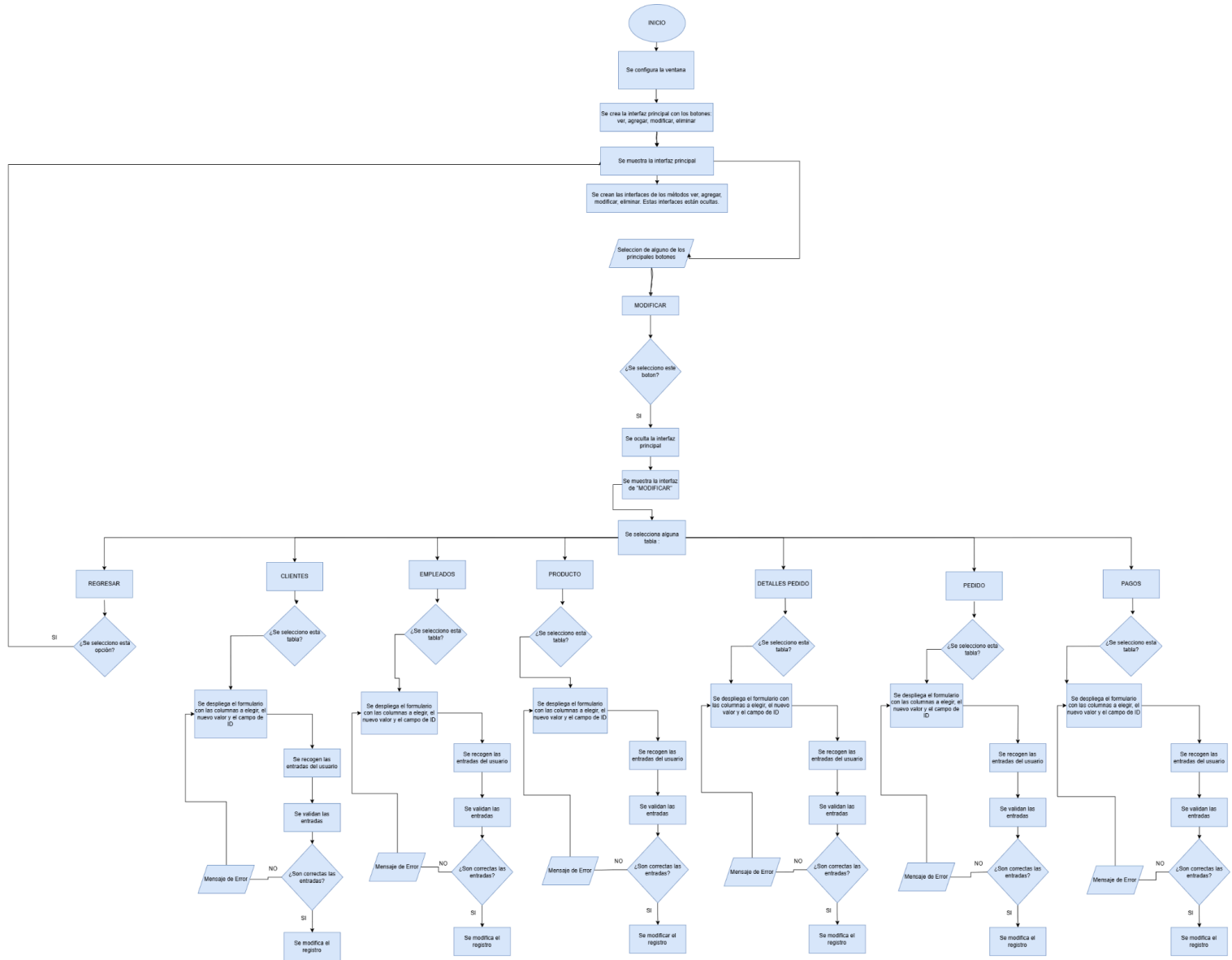
Método/Interfaz “VER”



Método/Interfaz “Agregar”



Método/Interfaz “Modificar”



Método/Interfaz “Eliminar”

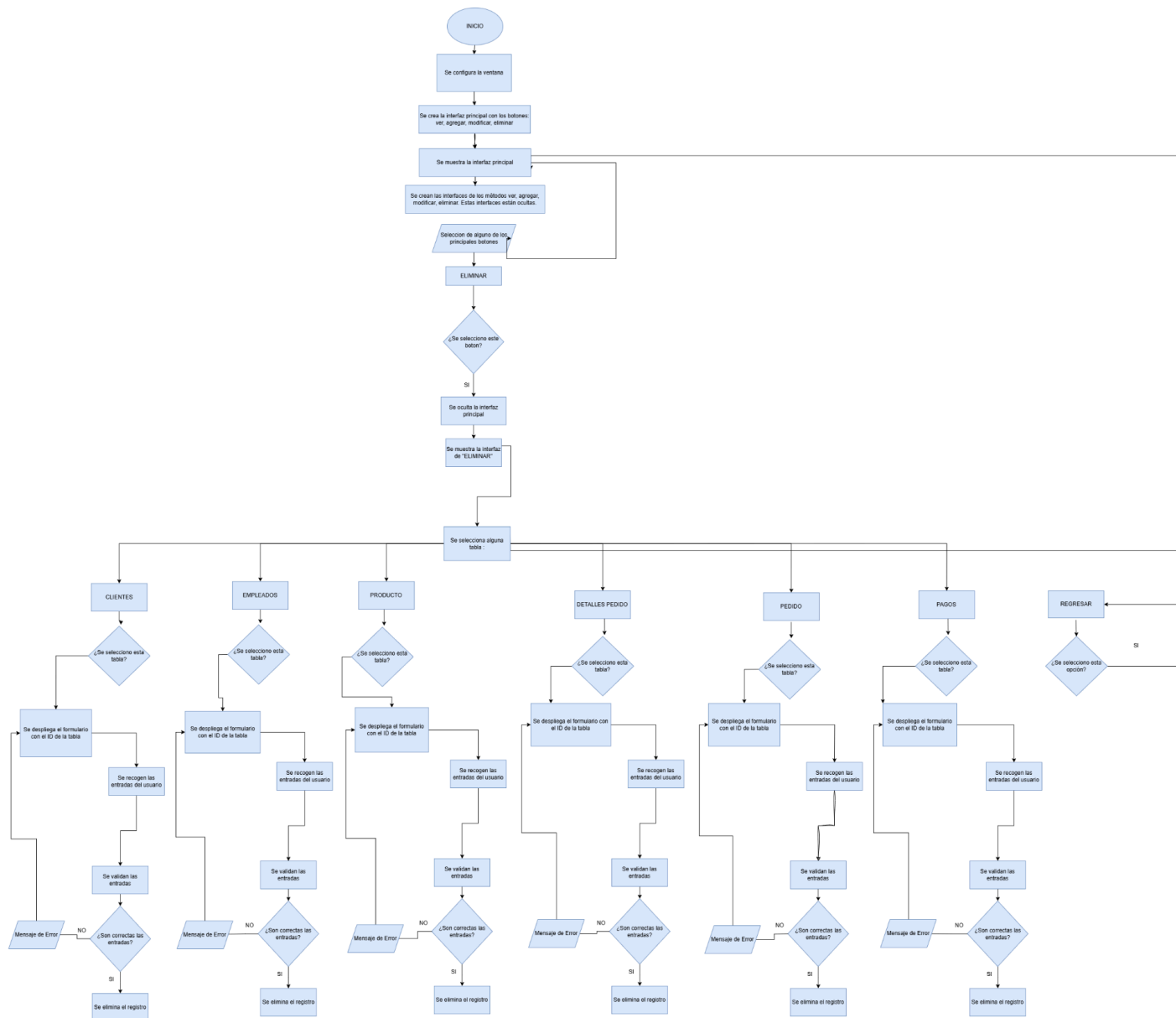


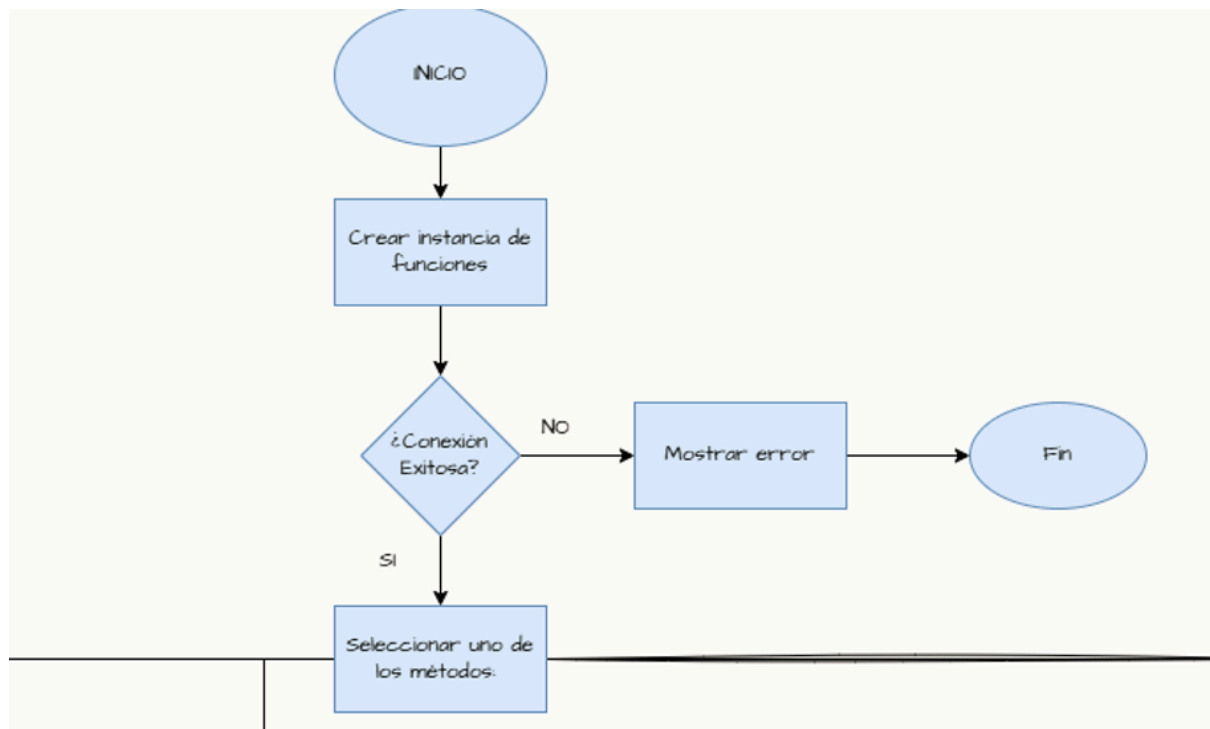
Diagrama UML de la Clase (Interfaz)

| Interfaz |
|---|
| -funciones: Funciones |
| -Bloque_Principal |
| -Interfaz_vista: Frame |
| -Interfaz_agregar: Frame |
| -Intefaz_modificar: Frame |
| -Interfaz_eliminar:Frame |
| -Entrys: list |
| -Labels: list |
| -Tabla: Treeview |
| +ocultar_mostrar() |
| +limpiar_entrys(self, *args) |
| + ver_tabla(self) |
| + metodo_agregar(self) |
| + guardar_agregar(self) |
| + metodo_modificar(self) |
| + guardar_modificar(self) |
| + metodo_eliminar(self) |
| + guardar_eliminar(self) |
| + crear_interfaz(self) |
| + objetos(self, columnas: list, editable=True) |

Módulo de Funciones (Funciones)

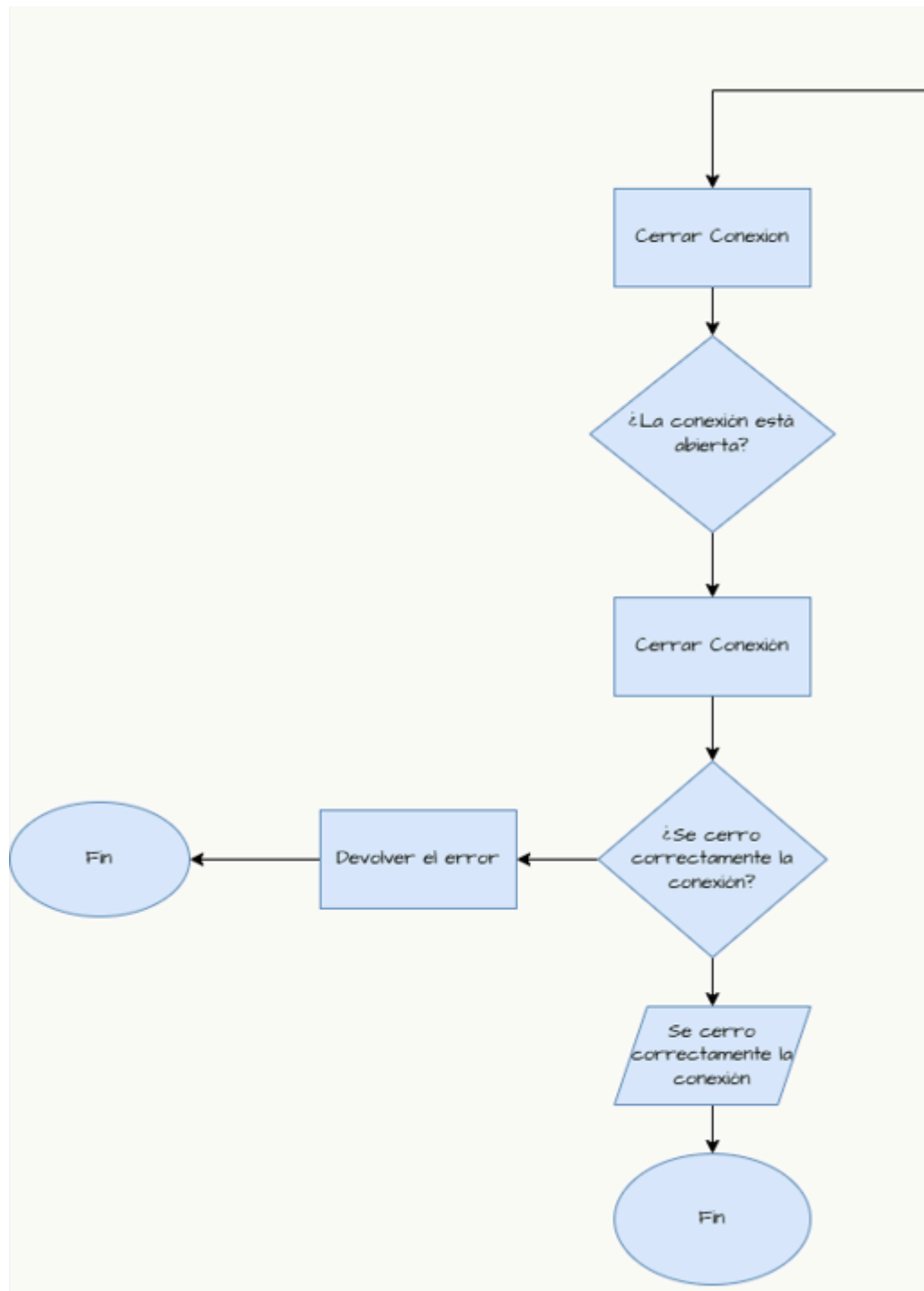
1. **Descripción:** Este módulo maneja toda la lógica del negocio que interactúa con la base de datos en **SQL Server**. Asegura la correcta ejecución de las operaciones solicitadas por el usuario en la interfaz.
2. **Responsabilidad:**
 - a. Establecer la conexión con las bases de datos en **SQL SERVER**.
 - b. Realizar las consultas SQL necesarias para obtener, insertar, actualizar y eliminar datos en SQL Server.
 - c. Gestionar la lógica de los procesos que serán ejecutados por el usuario a través de la interfaz.
3. **Dependencias:**
 - a. **Pyodbc:** Para la conexión con la base de datos SQL Server.
4. **Implementación:** El módulo de funciones realiza la conexión con las bases de datos y contiene las funciones que ejecutan las operaciones de lectura, escritura, actualización y eliminación sobre SQL Server.

Diagrama de flujo del Módulo (Funciones)

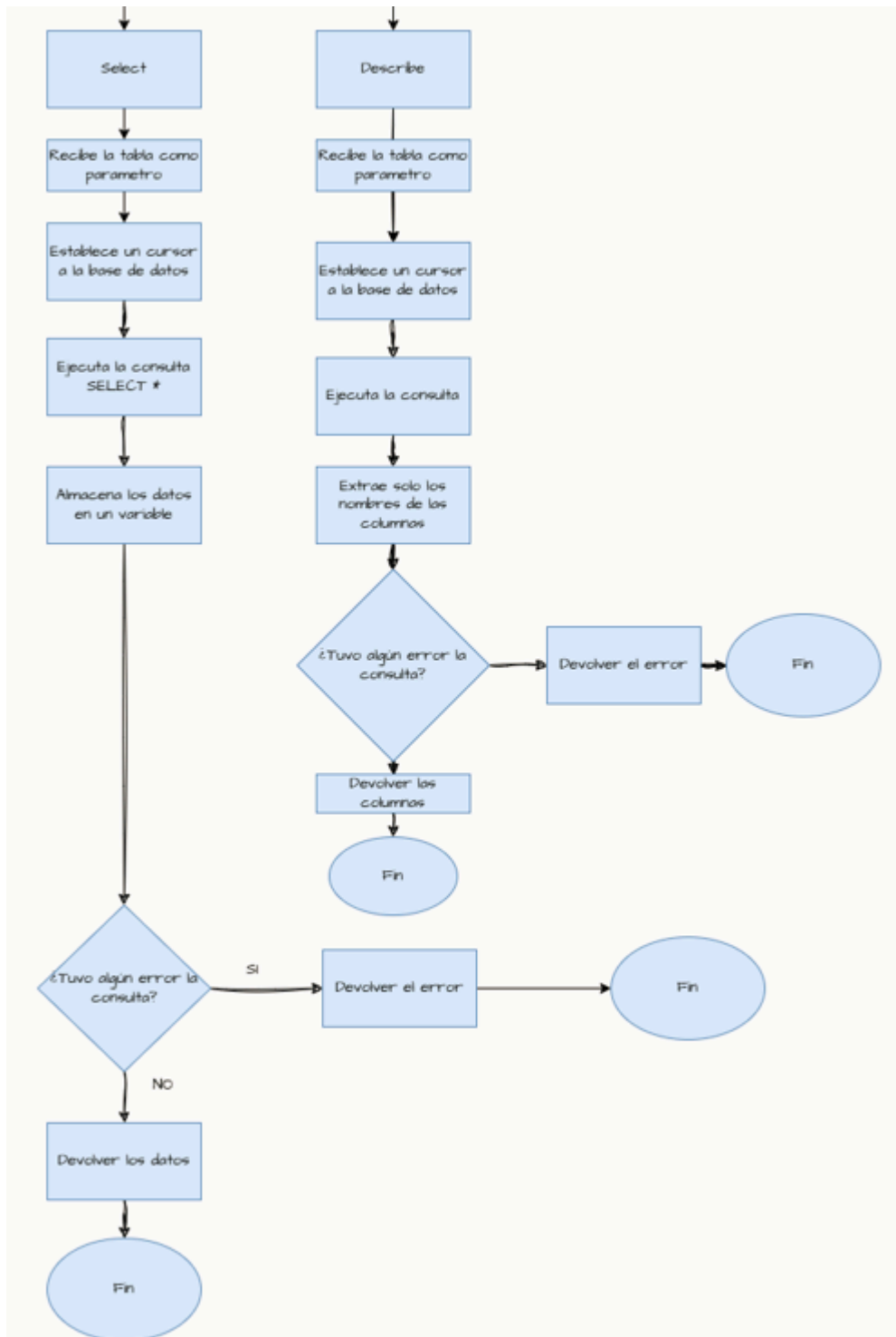


Izquierda del Diagrama de Funciones

Método Cerrar Conexión

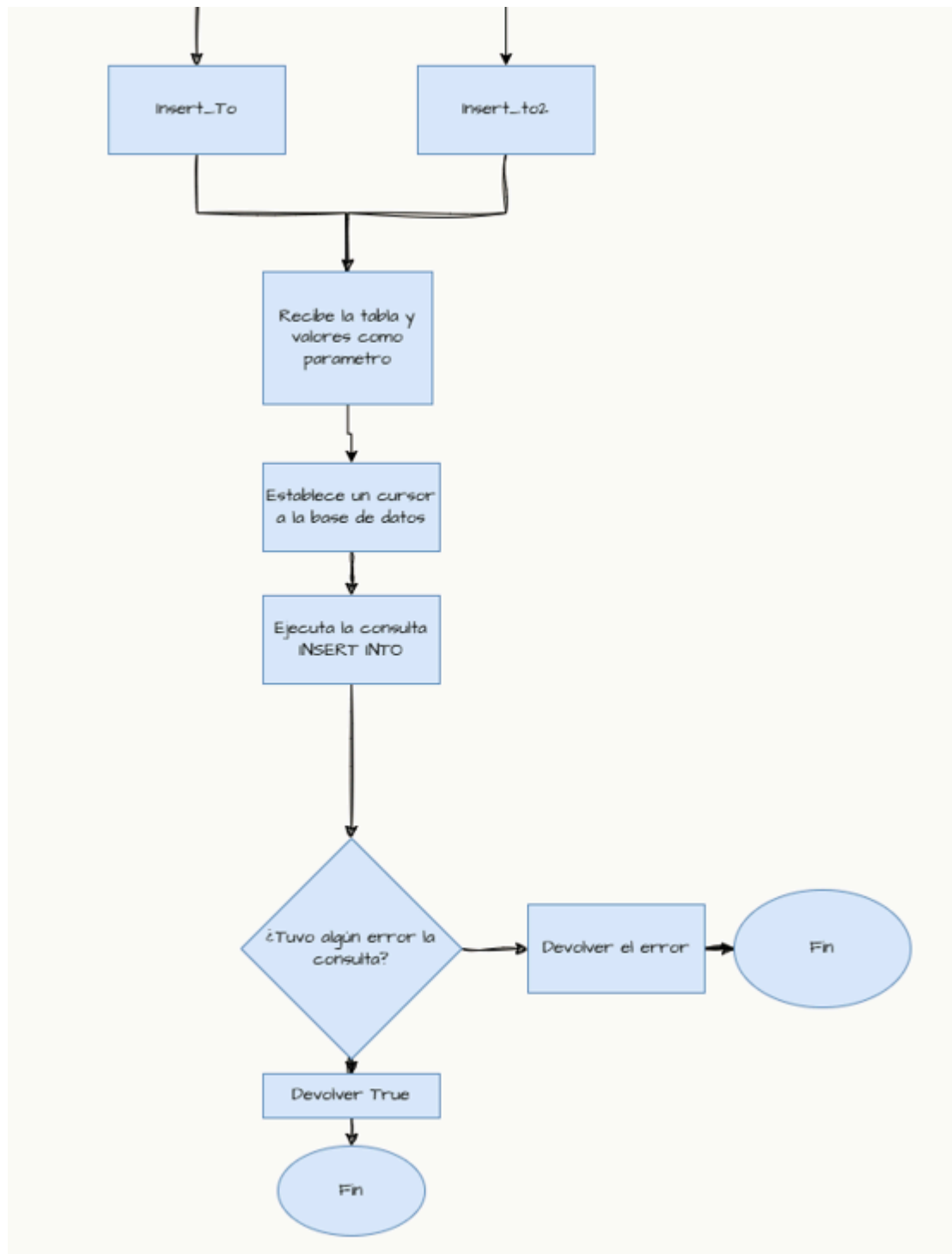


Método Select y Describe

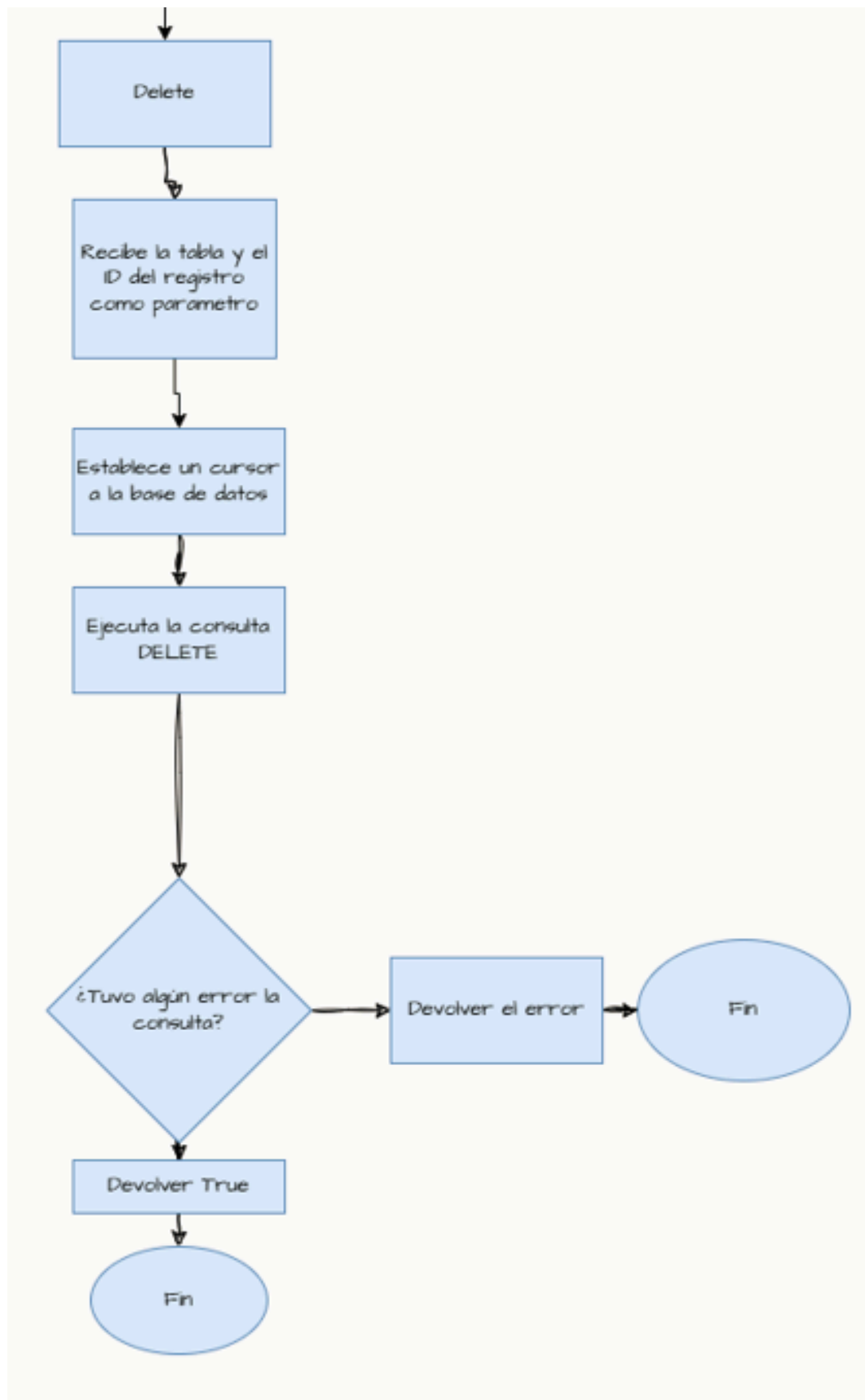


Derecha del Diagrama de Funciones

Método Insert_to



Método Delete



Método Modificar

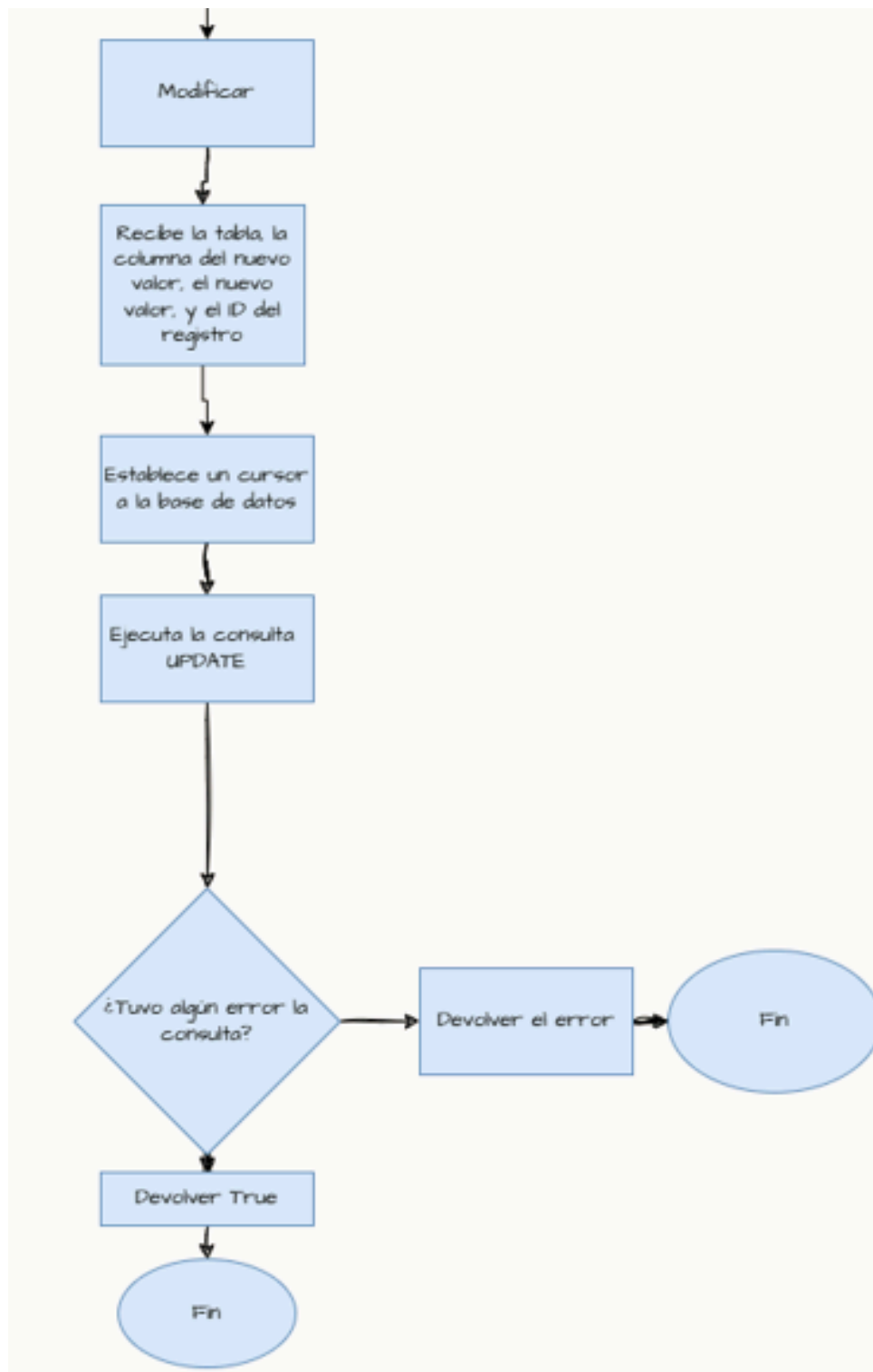
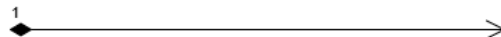


Diagrama UML de la Clase (Funciones)

| Funciones |
|---|
| + cerrar_conexion(self) |
| + select(self, query: str) |
| + describe(self, table: str) |
| + insert_to(self, table: str, values: list) |
| + modificar(self, table: str, conditions: dict) |
| + delete(self, table: str, conditions: dict) |

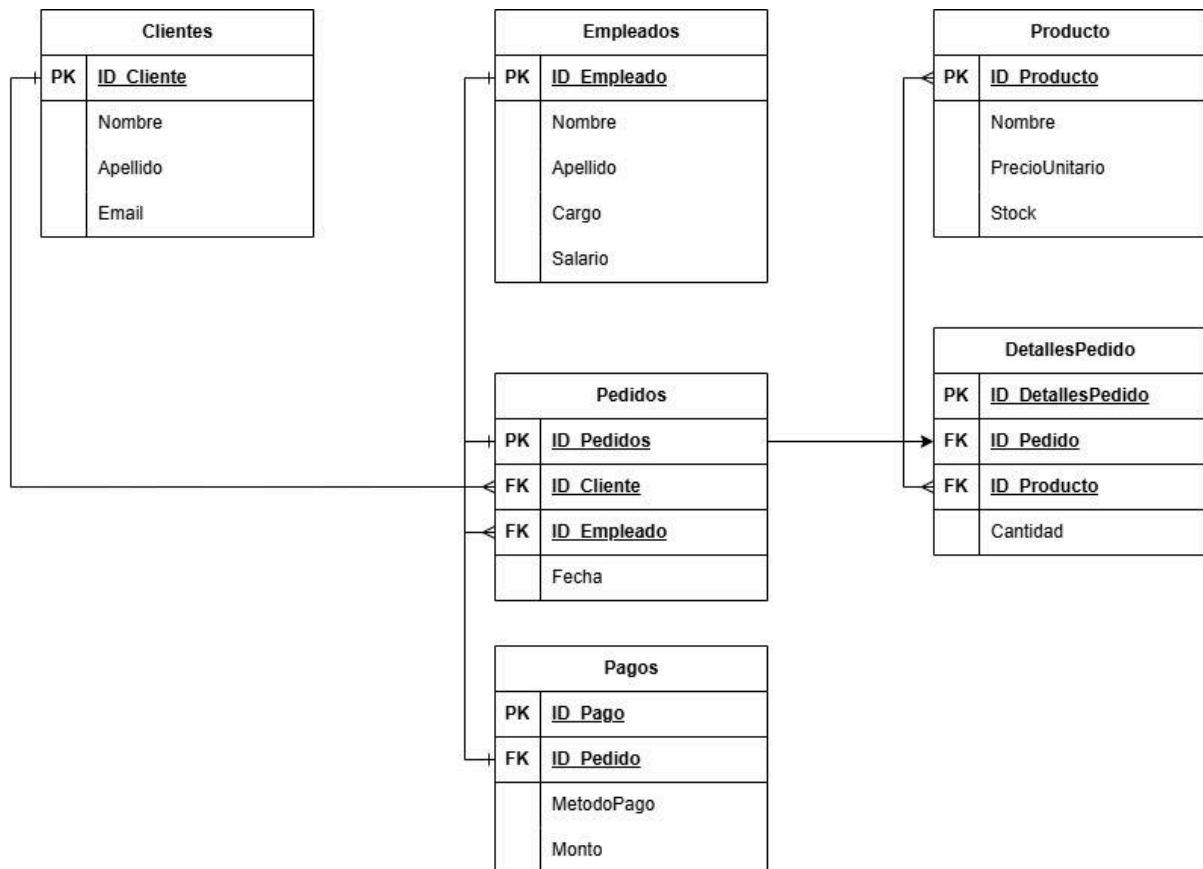
Relación entre clases:

| Interfaz |
|--|
| -funciones: Funciones |
| -Bloque_Principal |
| -Interfaz_vista: Frame |
| -Interfaz_agregar: Frame |
| -Intefaz_modificar: Frame |
| -Interfaz_eliminar:Frame |
| -Entrys: list |
| -Labels: list |
| -Tabla: Treeview |
| +ocultar_mostrar() |
| +limpiar_entrys(self, *args) |
| + ver_tabla(self) |
| + metodo_agregar(self) |
| + guardar_agregar(self) |
| + metodo_modificar(self) |
| + guardar_modificar(self) |
| + metodo_eliminar(self) |
| + guardar_eliminar(self) |
| + crear_interfaz(self) |
| + objetos(self, columnas: list, editable=True) |



| Funciones |
|---|
| + cerrar_conexion(self) |
| + select(self, query: str) |
| + describe(self, table: str) |
| + insert_to(self, table: str, values: list) |
| + modificar(self, table: str, conditions: dict) |
| + delete(self, table: str, conditions: dict) |

3.- Diseño del Modelo de Datos: Cafetería



Descripción General

La base de datos contiene tablas relacionadas con clientes, empleados, productos, pedidos, detalles de pedidos y pagos. Está estructurada para mantener información sobre las transacciones realizadas en la empresa y asegurar la trazabilidad de las ventas de una Cafetería.

Tablas y Propósito

1. Clientes:

- **Propósito:** Almacena información personal de los clientes.
- **Campos:**
 - **ID_Cliente** (PK): Identificador único del cliente.

- **Nombre:** Nombre del cliente.
- **Apellido:** Apellido del cliente.
- **Email:** Correo electrónico del cliente.
- Relación:
 - Cada cliente puede tener múltiples pedidos registrados en la tabla **Pedidos**.

2. Empleados:

- **Propósito:** Almacena información sobre los empleados que gestionan los pedidos.
- **Campos:**
 - **ID_Empleado** (PK): Identificador único del empleado.
 - **Nombre:** Nombre del empleado.
 - **Apellido:** Apellido del empleado.
 - **Cargo:** Puesto que ocupa en la empresa.
 - **Salario:** Salario del empleado.
- Relación:
 - Cada empleado puede estar asociado a varios pedidos en la tabla **Pedidos**.

3. Producto:

- **Propósito:** Almacena información sobre los productos disponibles para la venta.
- **Campos:**
 - **ID_Producto** (PK): Identificador único del producto.
 - **Nombre:** Nombre del producto.
 - **PrecioUnitario:** Precio por unidad del producto.
 - **Stock:** Cantidad disponible en inventario.
- Relación:
 - Cada producto puede aparecer en múltiples pedidos a través de la tabla **DetallesPedido**.

4. Pedidos:

- **Propósito:** Almacena los pedidos realizados por los clientes, gestionados por los empleados.
- **Campos:**

- **ID_Pedidos** (PK): Identificador único del pedido.
- **ID_Cliente** (FK): Referencia al cliente que realizó el pedido.
- **ID_Empleado** (FK): Referencia al empleado que gestionó el pedido.
- **Fecha**: Fecha en la que se realizó el pedido.
- Relación:
 - Se conecta con **DetallesPedido** para desglosar los productos incluidos en cada pedido.

5. **DetallesPedido:**

- **Propósito**: Representa la relación muchos a muchos entre **Pedidos** y **Producto**, almacenando los productos específicos y cantidades asociadas a un pedido.
- **Campos**:
 - **ID_DetallesPedido** (PK): Identificador único del detalle.
 - **ID_Pedido** (FK): Referencia al pedido asociado.
 - **ID_Producto** (FK): Referencia al producto asociado.
 - **Cantidad**: Cantidad de unidades del producto en el pedido.
- Relación:
 - Permite detallar cuántos productos y de qué tipo se incluyen en cada pedido.

6. **Pagos:**

- **Propósito**: Almacena información sobre los pagos realizados por los pedidos.
- **Campos**:
 - **ID_Pago** (PK): Identificador único del pago.
 - **ID_Pedido** (FK): Referencia al pedido asociado con el pago.
 - **MetodoPago**: Método utilizado para el pago (tarjeta, efectivo, etc.).
 - **Monto**: Cantidad pagada por el pedido.
- Relación:

- Cada pedido puede tener un pago asociado.

Relaciones

- **Clientes** → **Pedidos**: Relación 1 a muchos (un cliente puede tener varios pedidos).
- **Empleados** → **Pedidos**: Relación 1 a muchos (un empleado puede gestionar varios pedidos).
- **Pedidos** → **DetallesPedido**: Relación 1 a muchos (un pedido puede contener varios productos).
- **Producto** → **DetallesPedido**: Relación 1 a muchos (un producto puede estar en varios pedidos).
- **Pedidos** → **Pagos**: Relación 1 a 1 o 1 a muchos (cada pedido puede tener uno o más pagos asociados).

Resumen Ejecutivo

- **Nombre del proyecto**: YOUR DATA BASE (coffee shop)
- **Descripción breve**: Migramos una BD de un gestor a otro, cuidando la integridad de los datos y conexiones existentes entre tablas. Por último creamos una interfaz gráfica para manejar el nuevo gestor.
- **Estado actual**: Completado

Contribuidores

- Jose Alfredo Zambrana Cruz
- Nicolas Emanuel Oly Sánchez

"Este proyecto fue diseñado con pasión y compromiso. ¡Estamos emocionados de que formes parte de su evolución! Tu retroalimentación es invaluable para seguir mejorando."