**Título del Proyecto:**

**Sistema de Gestión de Hogwarts Store**

**1. Diseño de la Base de Datos**

**Estructura de Tablas y Relaciones :**

El diseño de la base de datos incluye las siguientes tablas:

1. **Clientes:**
   * Campos: id\_cliente, nombre, cedula, direccion, telefono, correo, historial\_compras.
   * Relación: id\_cliente se utiliza como clave foránea en la tabla ventas.
2. **Empleados:**
   * Campos: id\_empleado, nombre, rol, horario, salario, telefono.
   * Relación: id\_empleado se utiliza como clave foránea en la tabla ventas.
3. **Productos:**
   * Campos: id\_producto, nombre, descripcion, precio, categoria, existencias, id\_proveedor.
   * Relación: id\_proveedor se relaciona con la tabla proveedores. id\_producto se relaciona con las tablas inventarios y ventas.
4. **Inventarios:**
   * Campos: id\_inventario, id\_producto, cantidad\_vendida, fecha\_reposicion, existencias\_actuales.
   * Relación: id\_producto es clave foránea hacia la tabla productos.
5. **Proveedores:**
   * Campos: id\_proveedor, nombre, contacto, productos\_suministrados, fecha\_entrega.
   * Relación: id\_proveedor es clave primaria y está relacionada con productos.
6. **Ventas:**
   * Campos: id\_venta, id\_producto, id\_cliente, fecha\_hora, metodo\_pago, id\_empleado.
   * Relaciones: id\_producto, id\_cliente y id\_empleado son claves foráneas.

**Uso Correcto de Claves Primarias y Foráneas :**

* Cada tabla tiene una clave primaria definida (id\_cliente, id\_empleado, id\_producto, etc.).
* Las claves foráneas están correctamente relacionadas para mantener la integridad referencial:
  + Ejemplo: id\_producto en inventarios y ventas apunta a productos.

**Normalización y Eliminación de Redundancias :**

* El diseño está normalizado en tercera forma normal:
  + Cada tabla almacena datos únicos relacionados con su entidad.
  + Los datos redundantes se han eliminado al establecer relaciones entre tablas.

**Restricciones y Validaciones Implementadas :**

* Restricciones de integridad:
  + Claves primarias (PRIMARY KEY) en todas las tablas.
  + Claves foráneas (FOREIGN KEY) con restricciones de referencia.
* Validaciones:
  + Uso de NOT NULL en campos esenciales como nombre, precio, y correo.
  + Disparadores:
    - Actualización automática de inventarios después de cada venta mediante el disparador actualizar\_inventario\_al\_vender.

**2. Implementación de la Base de Datos**

**Creación Correcta de Tablas y Relaciones :**

* Las tablas están creadas con las relaciones necesarias para garantizar integridad referencial.

**Inserción Inicial de Datos Relevantes :**

* Se han insertado datos iniciales en las tablas:
  + Ejemplo: clientes incluye nombres como Harry Potter y Luna Lovegood.
  + productos tiene elementos como la Varita de Saúco y Chocolate de Rana.

**Scripts SQL Correctamente Escritos :**

* Los scripts incluyen la creación de tablas, relaciones y disparadores.
* Los datos iniciales están correctamente insertados usando INSERT INTO.

**Uso Adecuado del Motor de Bases de Datos Seleccionado :**

* Se utilizó **MySQL**, un motor robusto y eficiente.
* La base de datos se diseñó con un formato compatible con **MariaDB** y es fácilmente portable.

**3. Desarrollo de la Aplicación**

**Interfaz Gráfica de Usuario Intuitiva y Atractiva :**

* La aplicación utiliza **JavaFX** con pestañas para cada módulo (Clientes, Empleados, Productos, etc.).
* Diseño claro y bien organizado:
  + Tablas dinámicas para visualizar datos.
  + Botones para realizar operaciones CRUD.

**Funcionalidades Completas para Operaciones CRUD :**

* Se implementaron operaciones CRUD para todas las entidades:
  + Crear, leer, actualizar y eliminar clientes, empleados, productos, inventarios, proveedores y ventas.

**Comunicación Fluida con la Base de Datos :**

* La conexión con la base de datos se maneja mediante la clase DatabaseConnection.
* Consultas SQL dinámicas para interactuar con las tablas.
* Disparadores y validaciones aseguran la consistencia de datos.

**4. Documentación y Presentación**

**Descripción Clara del Diseño de la Base de Datos :**

* La base de datos se diseñó en tercera forma normal con relaciones bien definidas.
* Claves primarias y foráneas aseguran la integridad referencial.
* Disparadores como actualizar\_inventario\_al\_vender automatizan procesos clave.

**Explicación Detallada de la Estructura y Funcionalidades de la Aplicación:**

* La aplicación se organiza en pestañas:
  + **Clientes:** Gestión de clientes, historial de compras.
  + **Empleados:** CRUD para empleados.
  + **Productos:** Gestión de inventarios, categorías, y proveedores.
  + **Inventarios:** Actualización automática de existencias y ventas realizadas.
  + **Ventas:** Registro de transacciones, generación de reportes.
  + **Proveedores:** Gestión de contactos y suministros.

**5. Puntos Extra: Generación de Reportes**

* Se implementaron reportes en formato CSV:
  + **Reporte de ventas:** Genera informes diarios, semanales y mensuales.
  + **Clientes frecuentes:** Muestra los clientes con más compras, útil para promociones.
* Los reportes incluyen datos clave como nombres, fechas y totales.

**Conclusión**

El sistema desarrollado cumple con todos los requisitos establecidos en el proyecto, destacándose por su diseño eficiente de base de datos, una interfaz gráfica atractiva y funcionalidades avanzadas como generación de reportes.