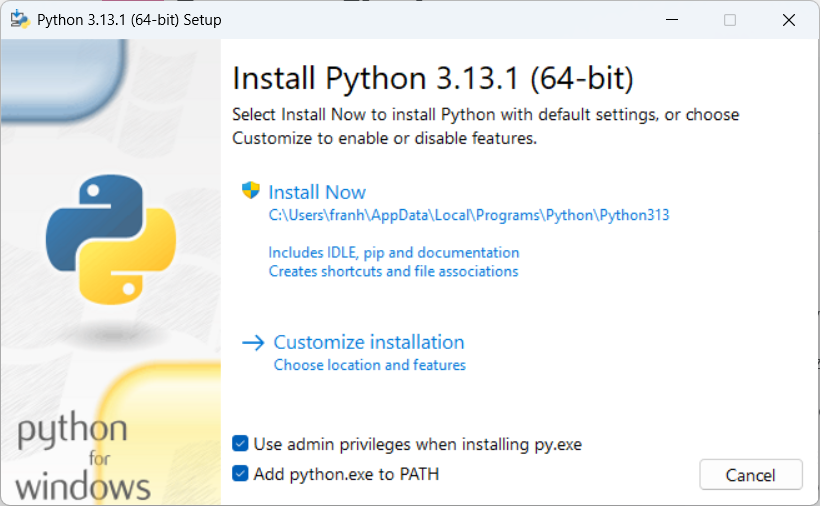
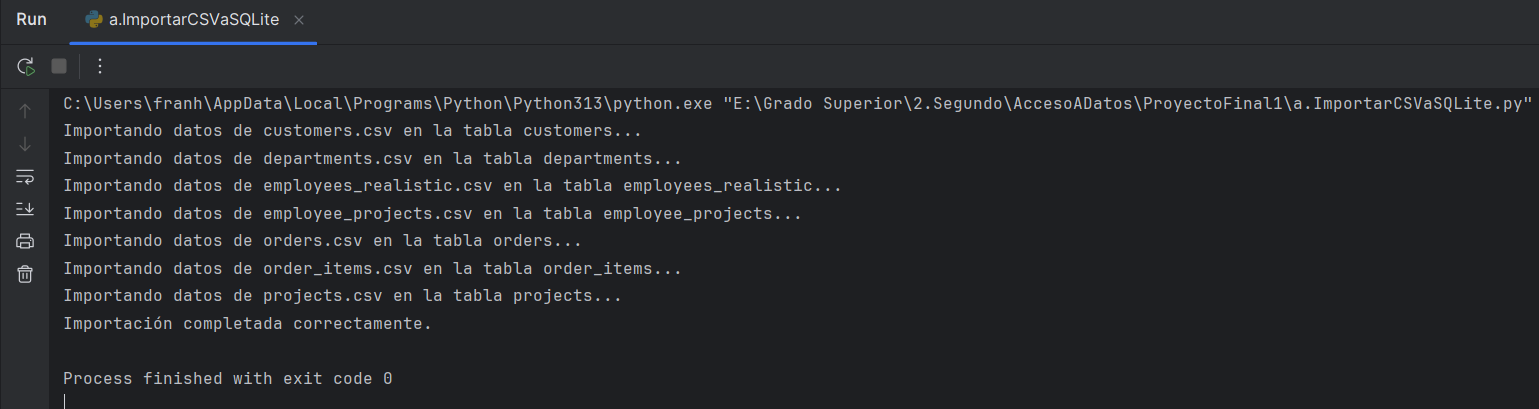
**Funcionalidad:**

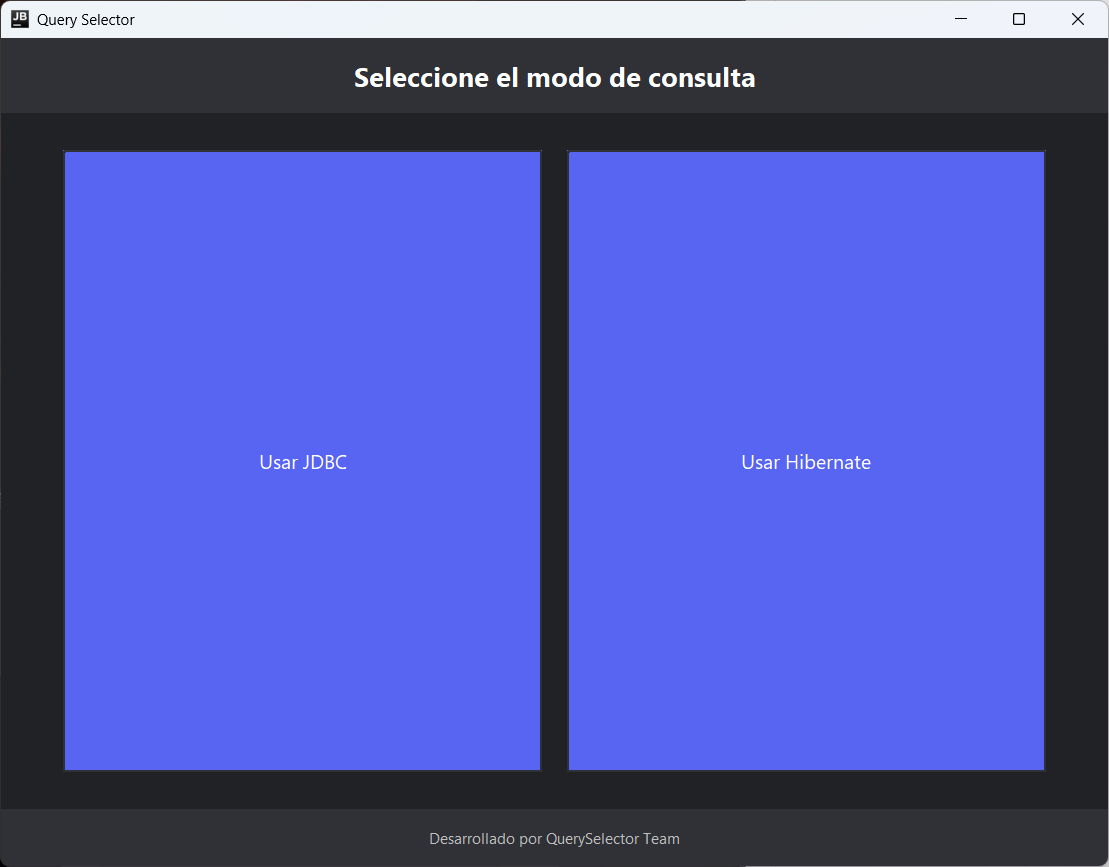
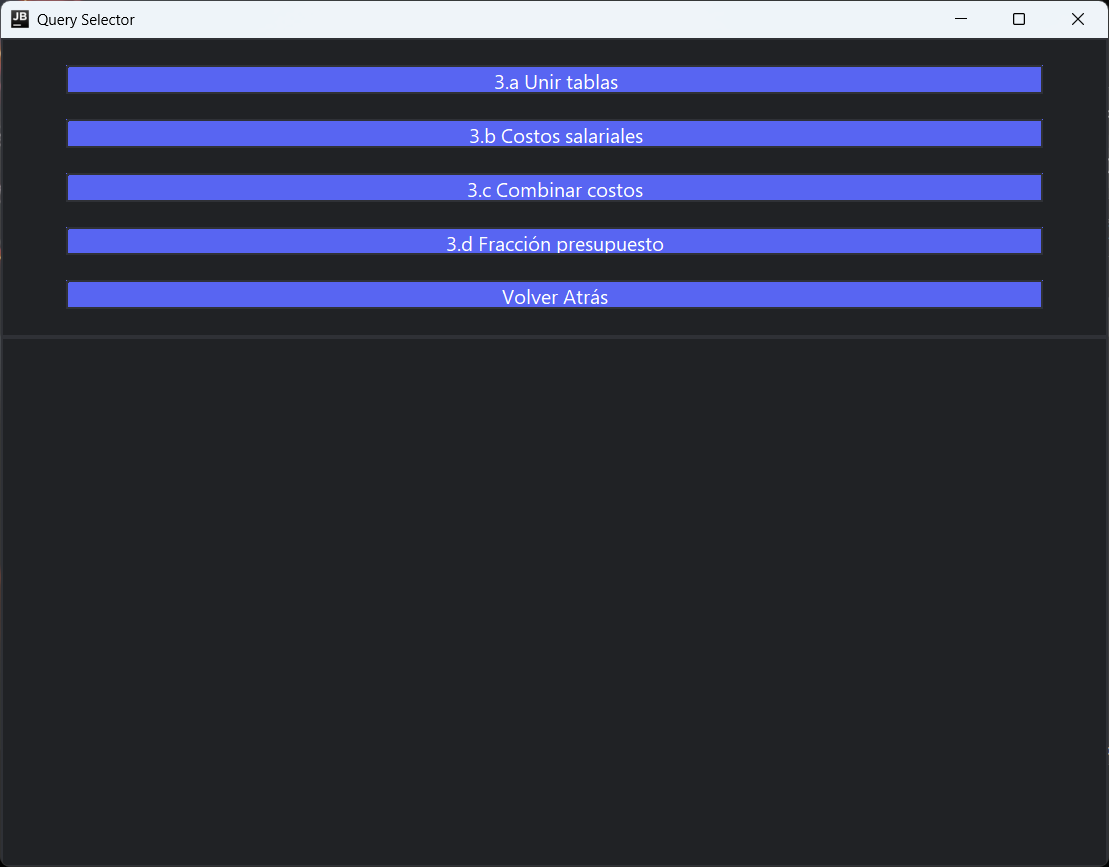
1. Asumimos que tenemos instalado Python (y está añadido a PATH) y un IDE de Java como IntelliJ IDEA o NetBeans



1. Ejecutamos el archivo “a.ImportarCSVaSQLite.py” ubicado en src/main/csv para crear la base de datos SQLite (company\_database.db, se creará en la ruta del proyecto) e importar los archivos .csv a la base de datos. En caso de que no estén todos los archivos ("customers.csv", "departments.csv", "employees\_realistic.csv", "employee\_projects.csv", "orders.csv", "order\_items.csv", "projects.csv") la importación no se efectuará.



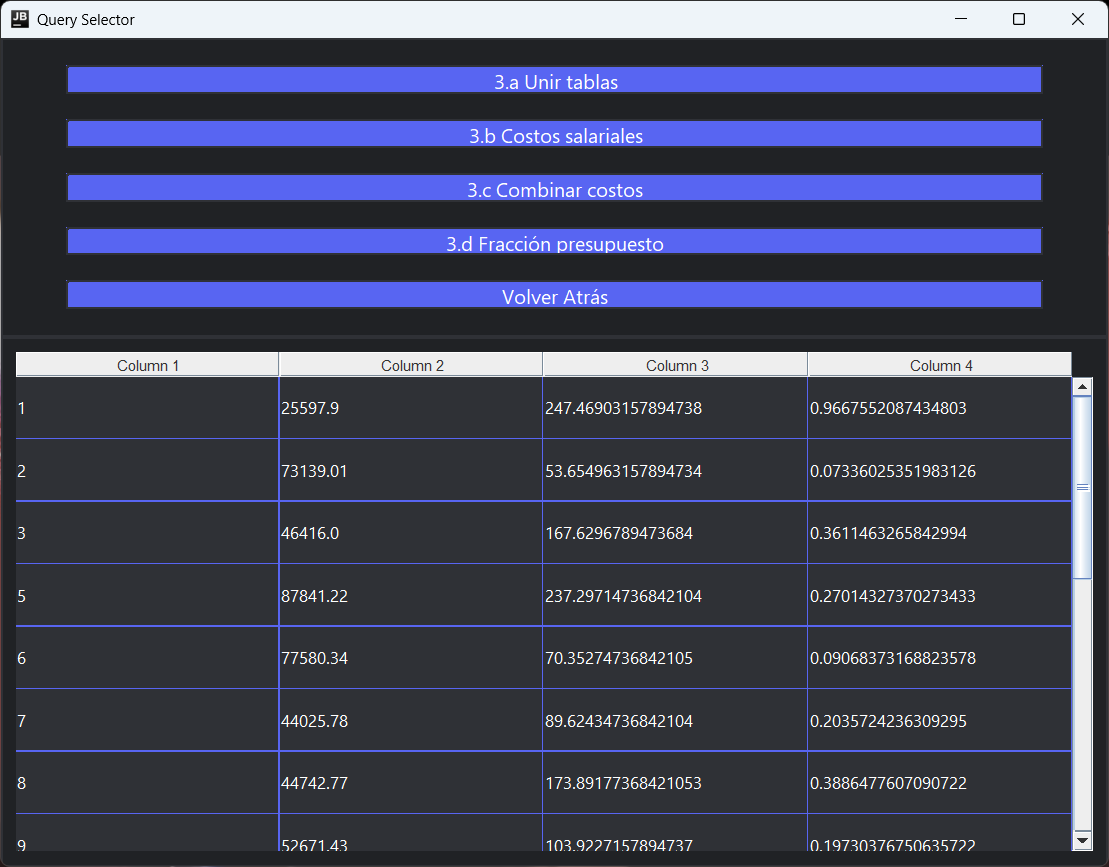
En caso de fallo porque Python no funcione, podemos ejecutar el archivo “ImportCSVtoSQLite.java” y ocurrirá lo mismo.

1. Las consultas pedidas se han realizado en archivos .sql (“b.ConsultarInformacionProyecto.sql”, “CalcularCostosSalarialesPorProyecto.sql”, “CombinarCostosConPresupuesto.sql” y “e.CalcularFraccionPresupuestoCostosSalariales.sql”).
2. En el archivo “SQLiteJDBCFullQueries.java” se encuentran las consultas mencionadas previamente que se ejecutan utilizando JBDC.  
   Lo ejecutamos y obtenemos lo siguiente:  
   
3. En el archivo “HibernateSQLiteFullQueries.java” se encuentran las consultas mencionadas previamente que se ejecutan utilizando Hibernate.  
   Lo ejecutamos y obtenemos lo siguiente:  
   
4. Para mejorar la visualización de los datos, contamos con la clase “QuerySelectorGUI.java”, que, al ejecutarla, nos ofrece una GUI de Java Swing donde:
   1. Podemos seleccionar si usar JDBC o Hibernate para la consulta a realizar:  
      
   2. Independientemente de cuál seleccionemos, nos mostrará la siguiente lista de consultas disponibles:  
      
   3. Seleccionaremos la 3.d: Primero en JDBC, y luego en Hibernate

En JDBC se ven los nombres de las columnas:



En Hibernate no se ven los nombres de las columnas, puesto que hibernate trabaja con objetos y no tiene un método de acceder al nombre de las columnas:

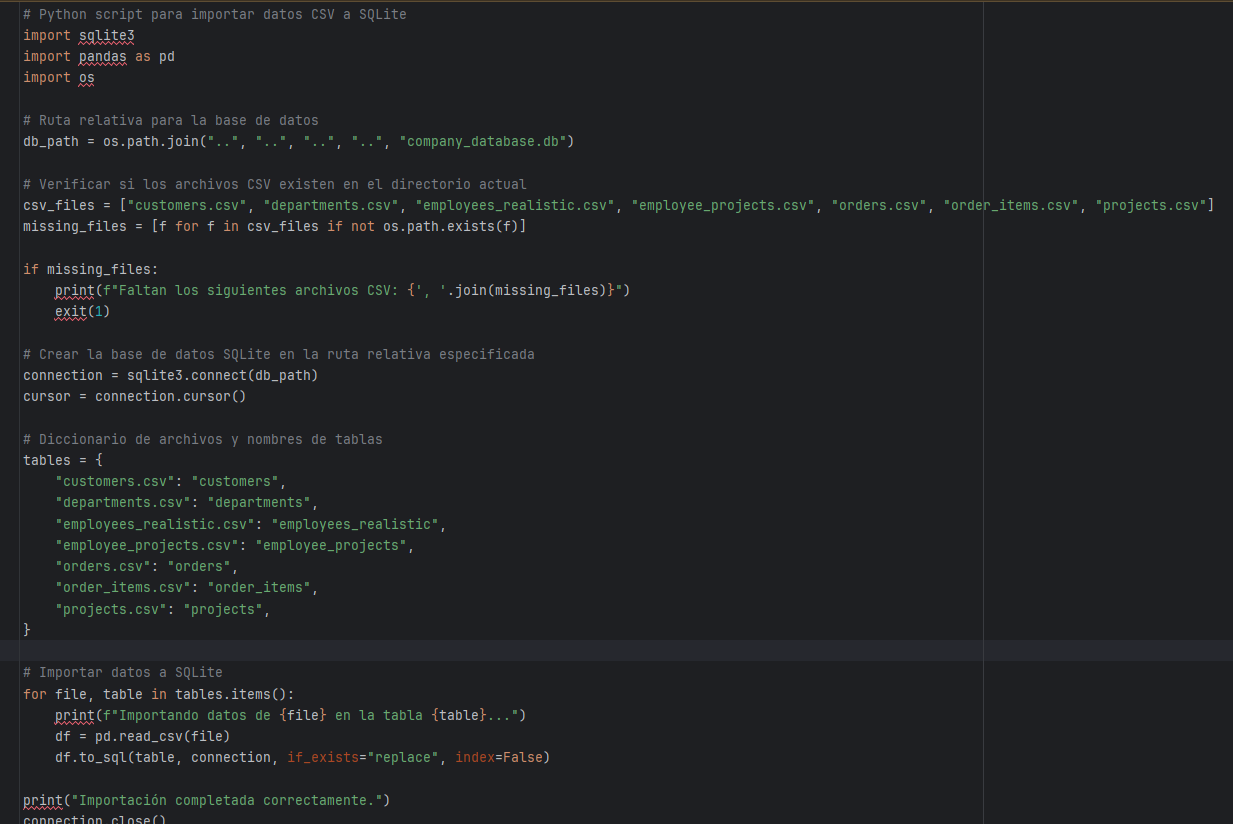


**Documentación del código**

**Importación de Datos**

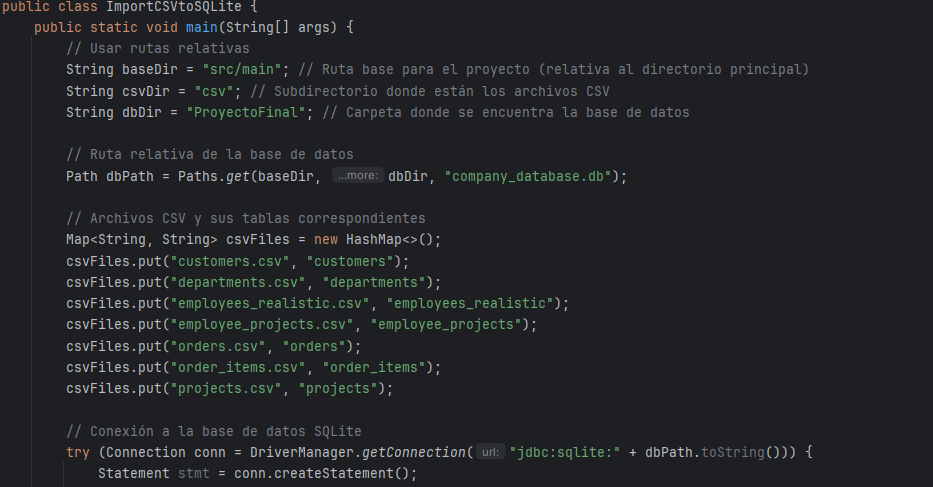
**a. ImportarCSVaSQLite.py**

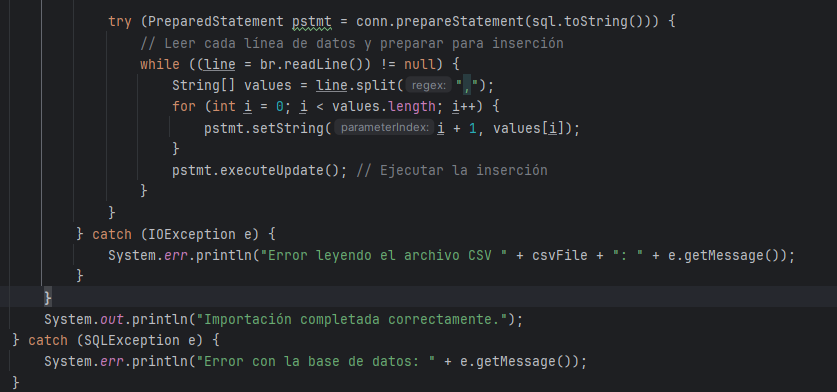
Este script en Python utiliza pandas para leer archivos CSV y sqlite3 para insertar los datos en una base de datos SQLite.

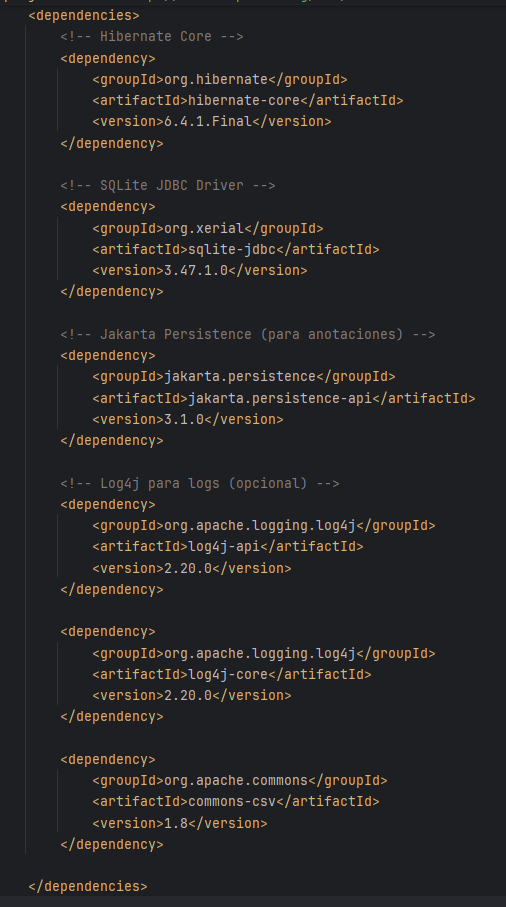


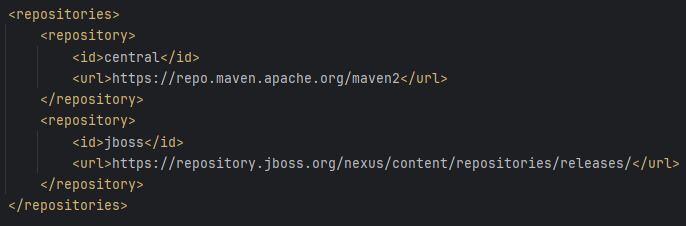
**b. ImportCSVtoSQLite.java**

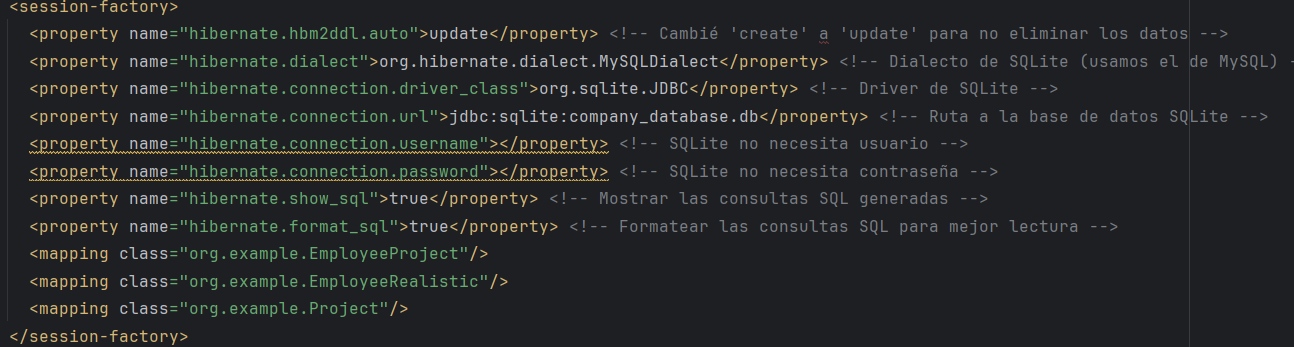
Esta clase Java lee archivos CSV desde un directorio específico y los inserta en la base de datos SQLite utilizando JDBC.  
Captura del main:

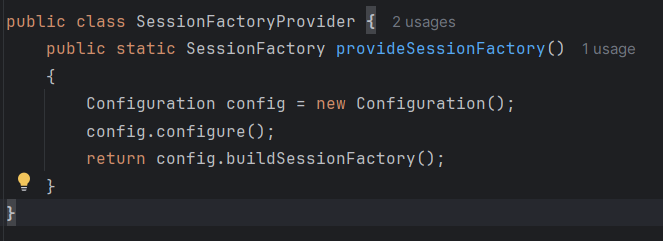


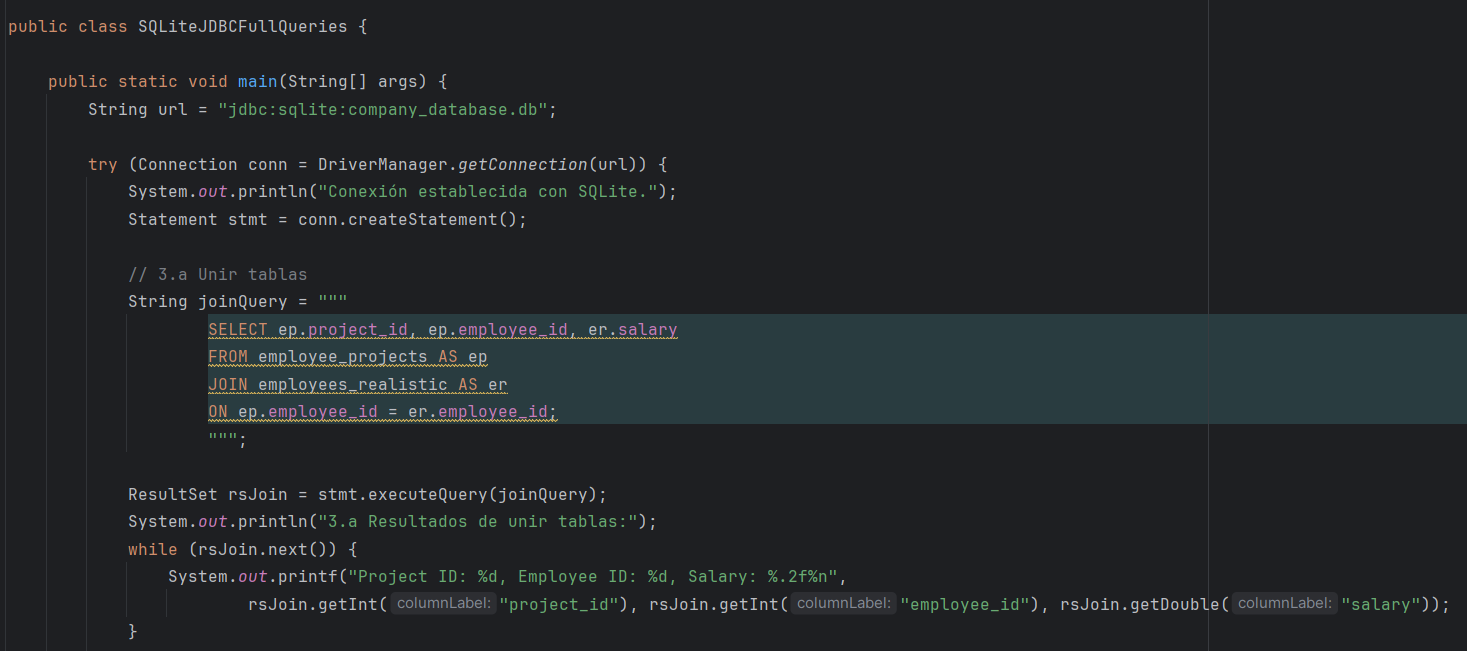
Captura de la sentencia SQL para inserción:  


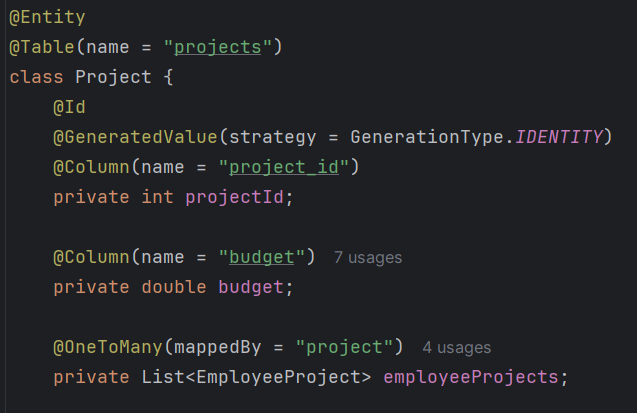
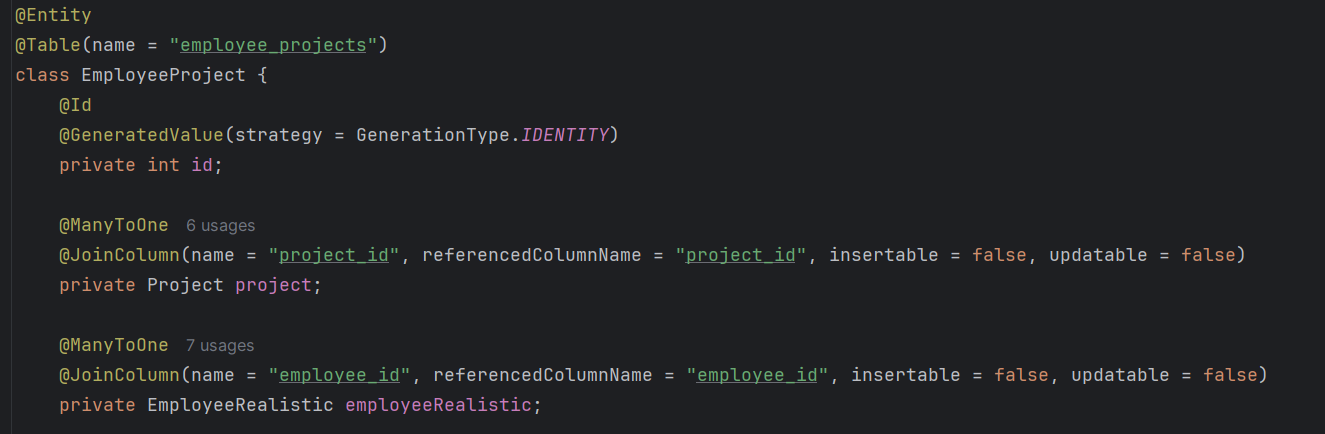
**Configuración de Maven  
pom.xml**  
Este archivo contiene las configuraciones y dependencias necesarias para el proyecto, incluyendo Hibernate y SQLite JDBC Driver.  
Captura del bloque <dependencies> de pom.xml:  


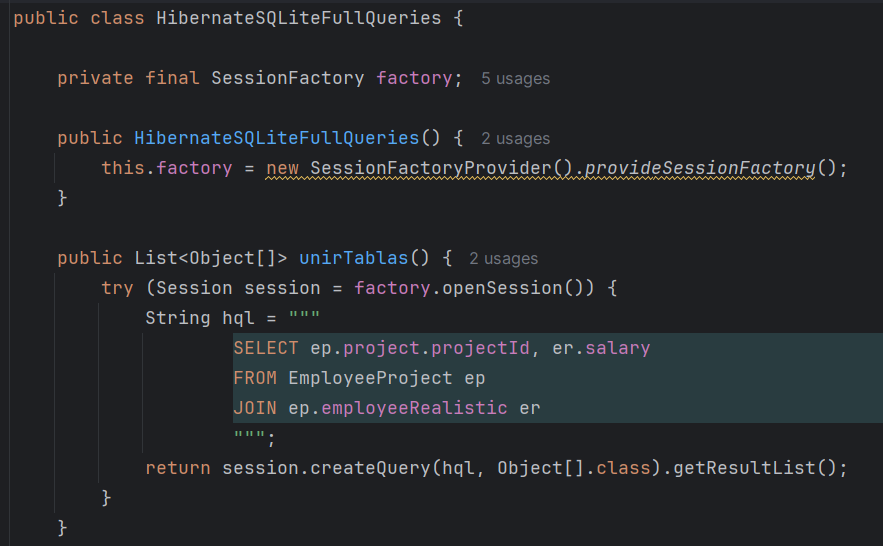
Captura del bloque <repositories> de pom.xml:  


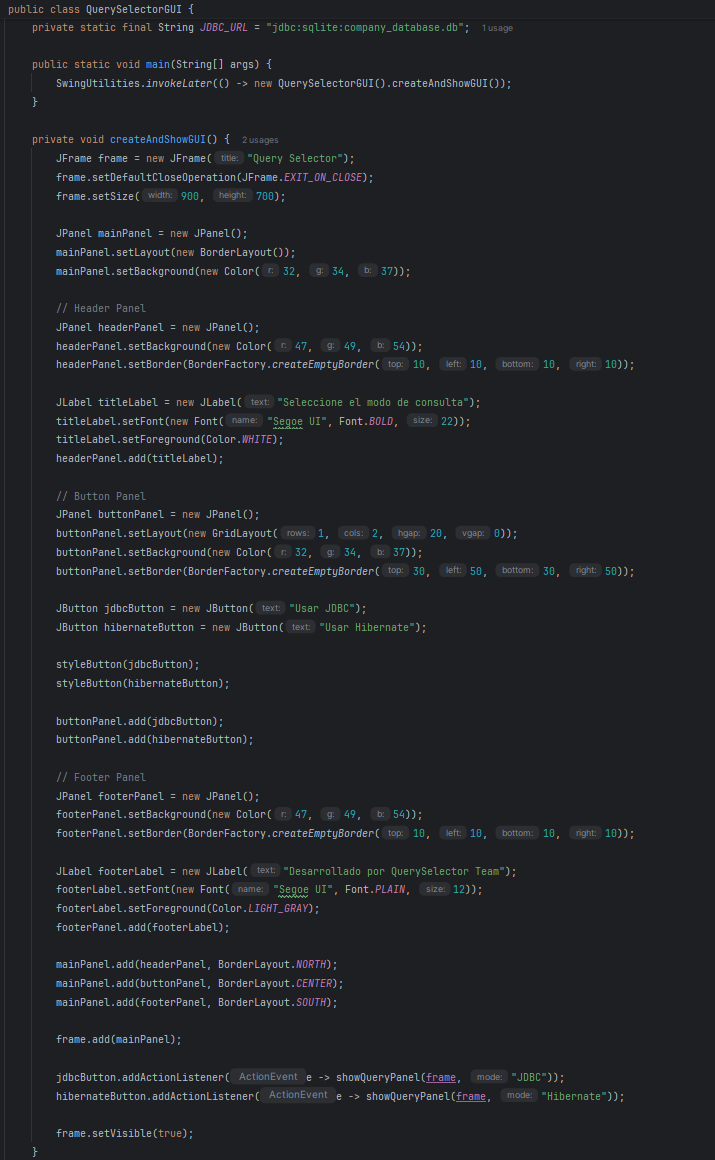
**Configuración de Hibernate:  
hibernate.cfg.xml**Este archivo XML configura Hibernate para conectar con la base de datos SQLite y mapear las clases en entidades/objetos.  
Captura del bloque <session-factory> de hibernate.cfg.xml: y captura de las etiquetas <mapping class="..."/>:  


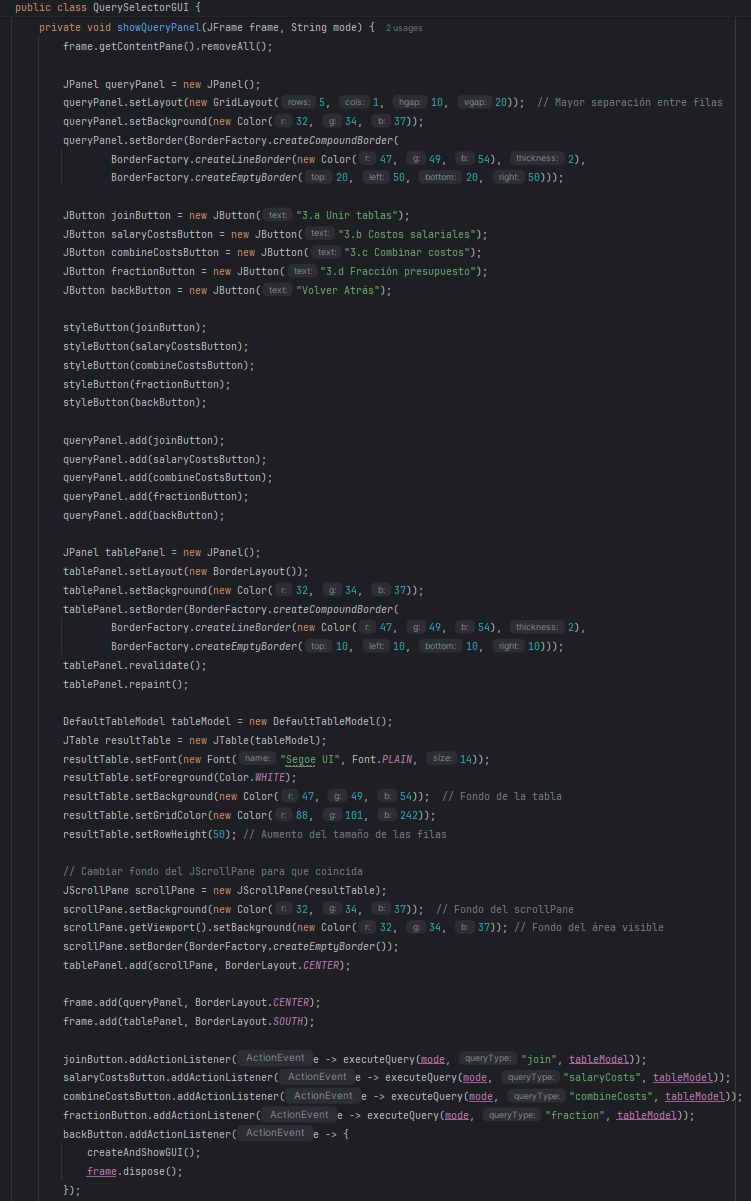
**Clases de Hibernate  
SessionFactoryProvider.java**  
Esta clase proporciona una instancia única de SessionFactory.  
Captura del método provideSessionFactory() de SessionFactoryProvider.java:  


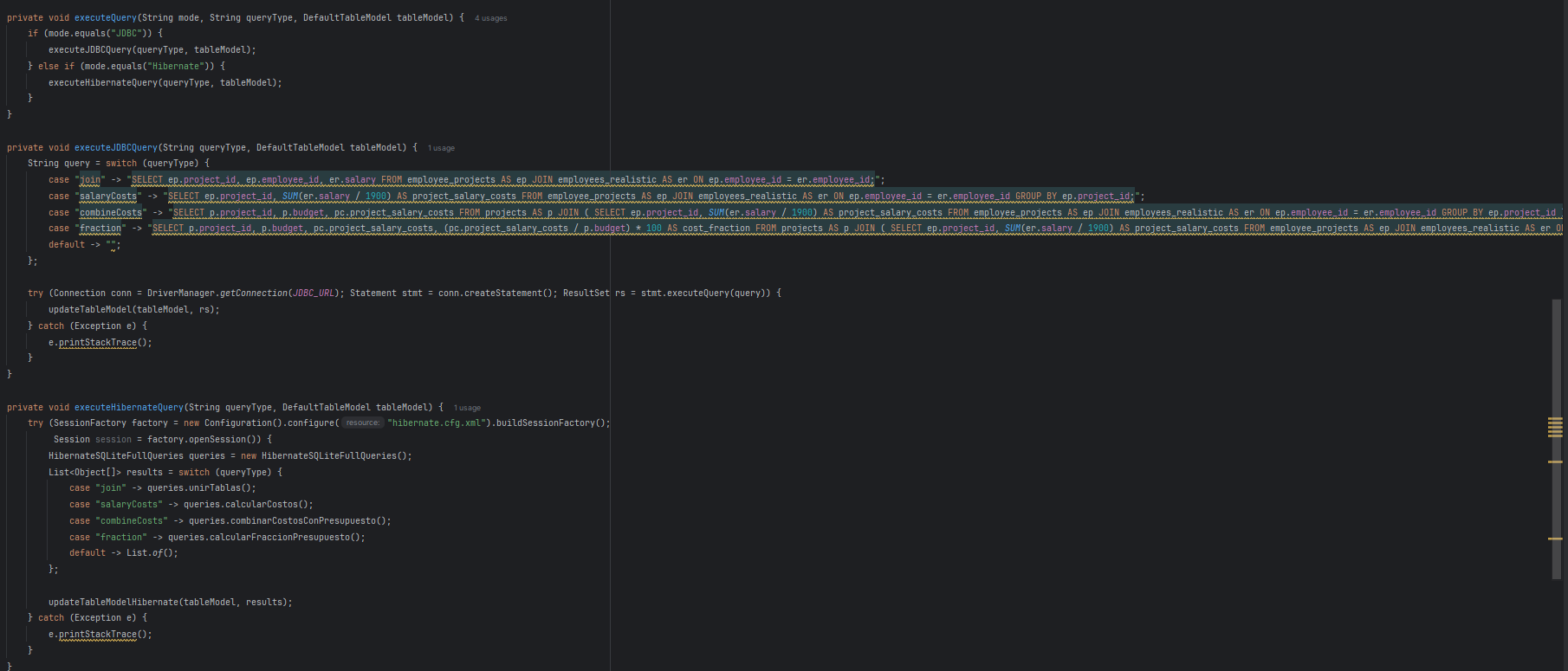
**Consultas con JDBC y Hibernate  
SQLiteJDBCFullQueries.java**Este script Java contiene consultas SQL directas a la base de datos SQLite para unir tablas, calcular costos salariales, combinar costos con presupuesto y calcular fracciones del presupuesto.Captura del bloque try donde se ejecutan las consultas (de una consulta específica joinQuery):  


**HibernateSQLiteFullQueries.java  
Esta clase Java utiliza Hibernate para realizar consultas similares a las realizadas con JDBC.**Captura de la definición de las entidades (EmployeeProject, EmployeeRealistic, Project):  


Captura del método unirTablas(), que realiza la primera consulta (3.a) utilizando Hibernate:  


**Interfaz Gráfica (GUI)  
QuerySelectorGUI.java**Esta clase Java crea una interfaz gráfica utilizando Java Swing que permite al usuario seleccionar el modo de consulta (JDBC o Hibernate) y ejecutar diferentes consultas.Captura del método createAndShowGUI():  


Captura del método showQueryPanel(JFrame frame, String mode):  


Captura de los métodos para ejecutar Query (con JDBC o Hibernate):