

**INSTITUTO TECNOLOGICO DE CHILPANCINGO**

**Cómputo En La Nube Y Grandes Datos**

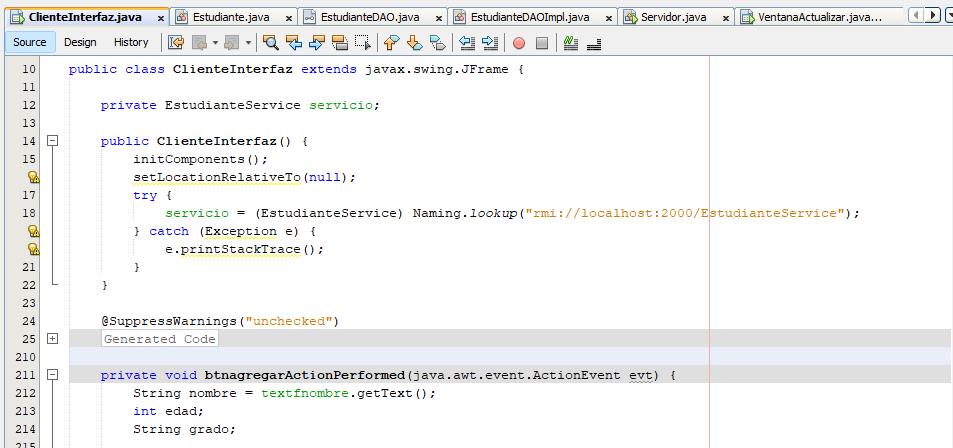
Practica RMI Con Capa

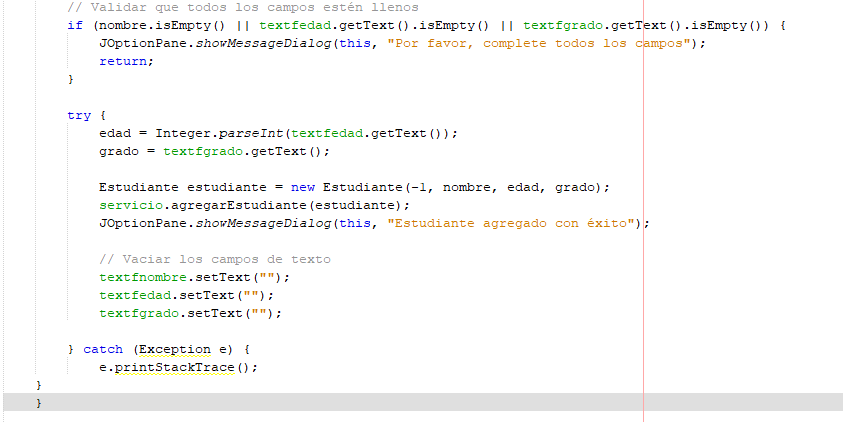
**Alumno:**

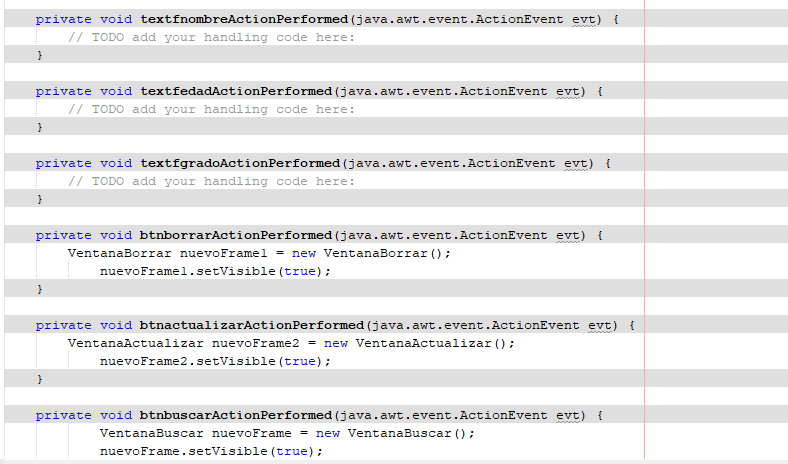
Fierro Monroy Isaac Antonio

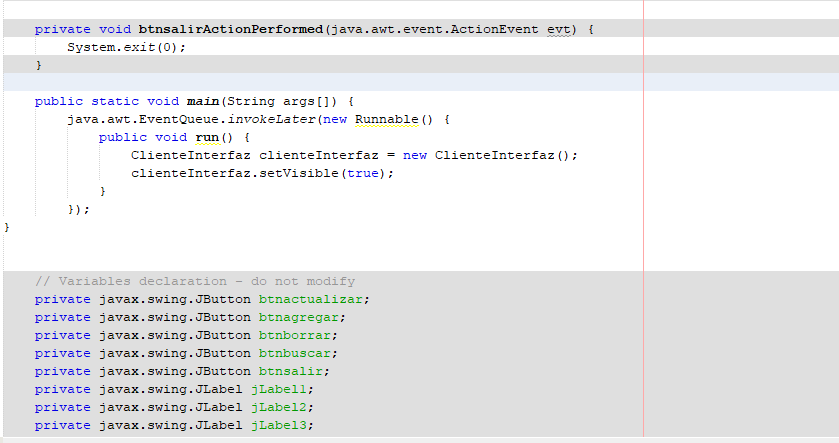
Lunes 13 de mayo de 2024

# RMI CON CAPA

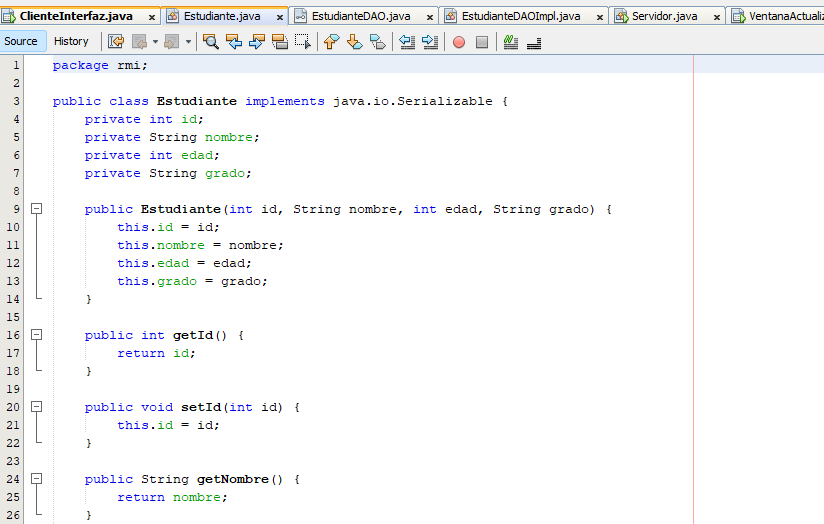


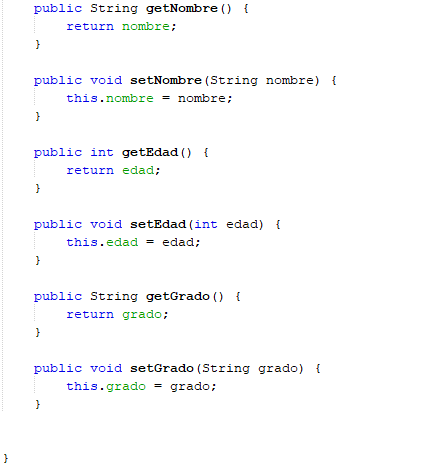




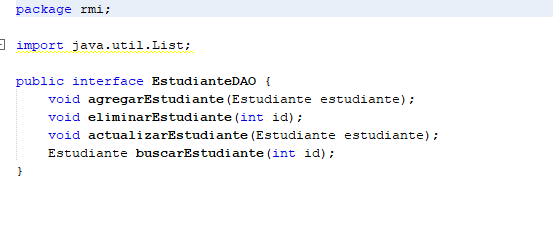


El código proporciona una interfaz de usuario intuitiva para interactuar con un servicio RMI que gestiona datos de estudiantes, facilitando la realización de operaciones CRUD de manera sencilla y eficiente.

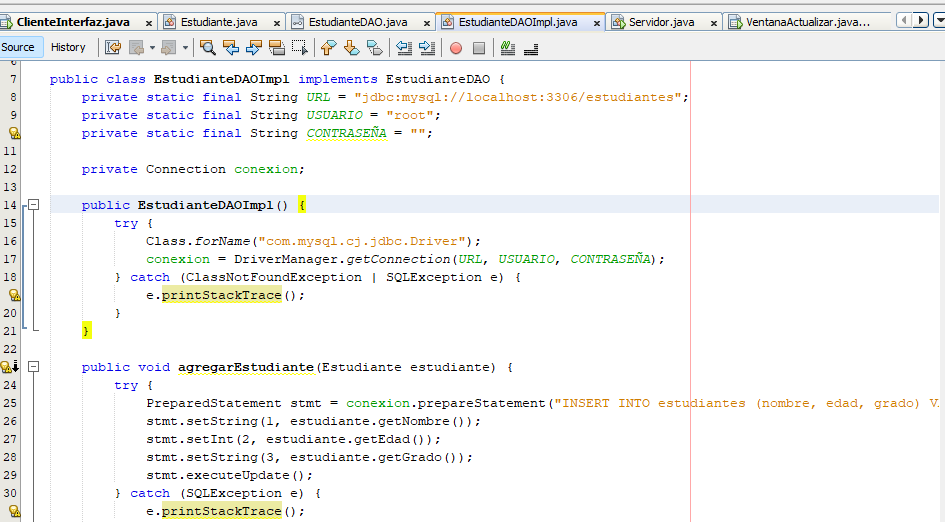


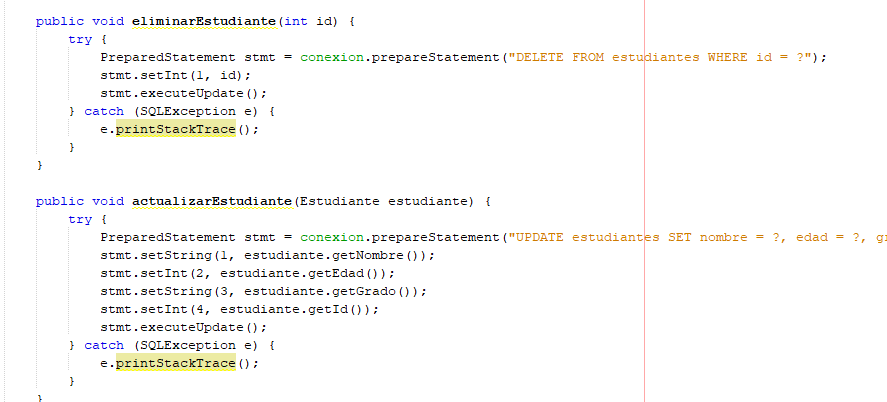


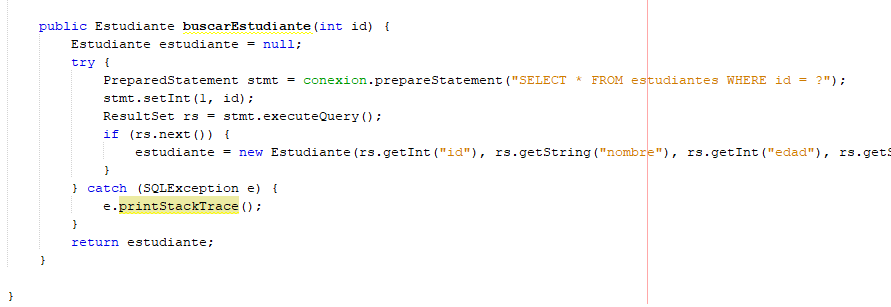
Este código define una clase Estudiante en el paquete rmi. Esta clase implementa la interfaz java.io.Serializable, lo que significa que los objetos de esta clase pueden ser serializados y desserializados, lo que es importante cuando se transfieren objetos entre procesos en una comunicación RMI.



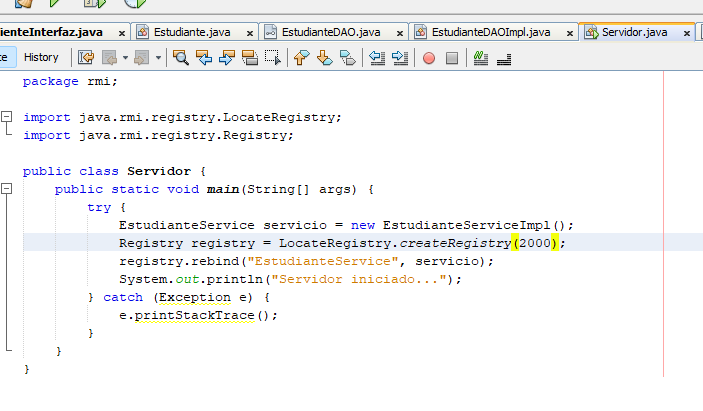
Este código define una interfaz llamada EstudianteDAO en el paquete rmi. Esta interfaz especifica las operaciones básicas que pueden ser realizadas sobre los datos de estudiantes, siguiendo el patrón de acceso a datos (DAO, por sus siglas en inglés: Data Access Object).



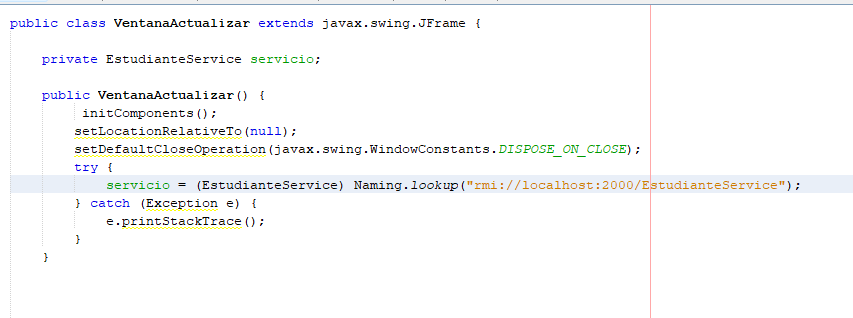


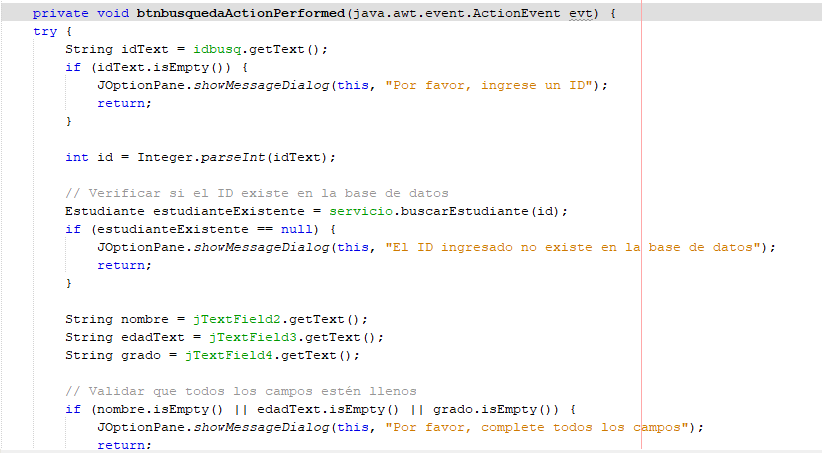


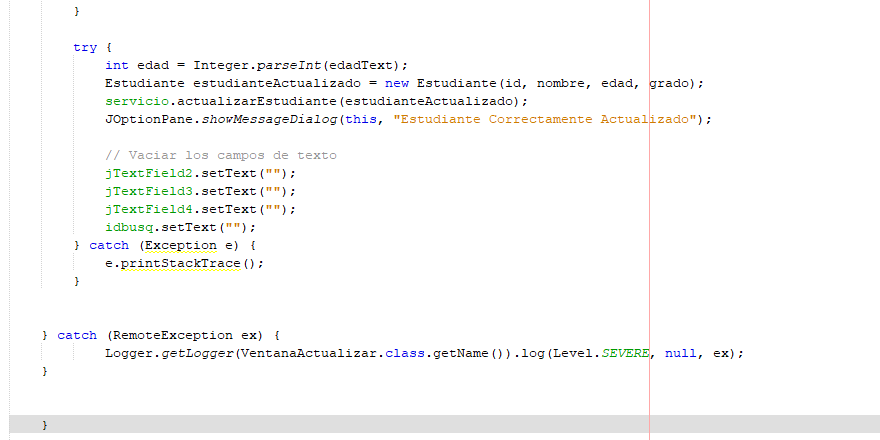
Este código proporciona una implementación concreta de la interfaz EstudianteDAO, llamada EstudianteDAOImpl, en el paquete rmi. Esta implementación utiliza JDBC (Java Database Connectivity) para interactuar con una base de datos MySQL y realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los datos de los estudiantes.

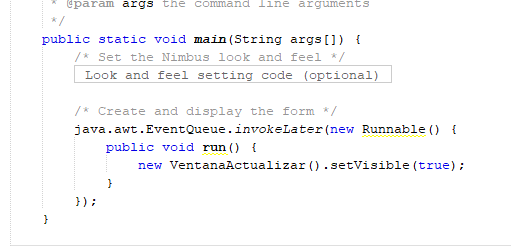


Este código implementa una clase Servidor en el paquete rmi. La clase Servidor contiene un método main que se encarga de iniciar el servidor RMI para proporcionar el servicio EstudianteService.

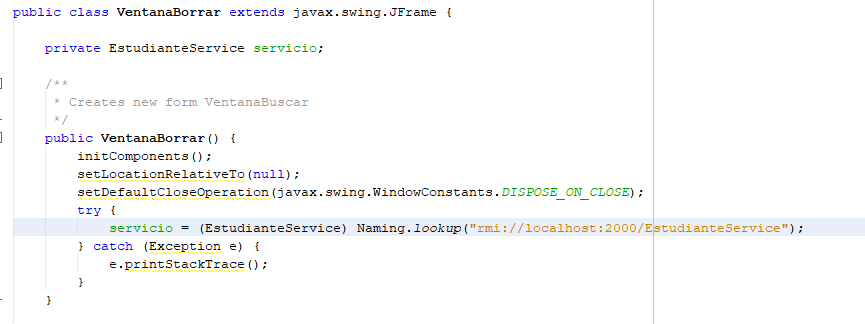


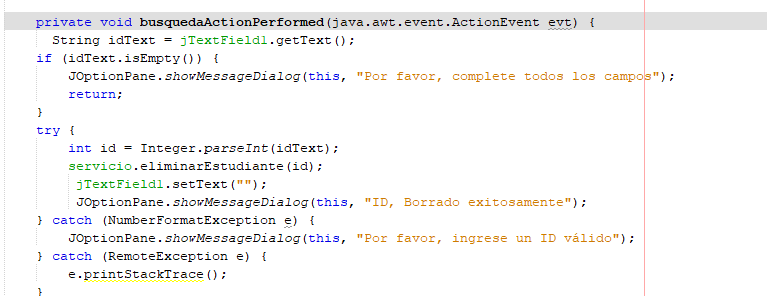




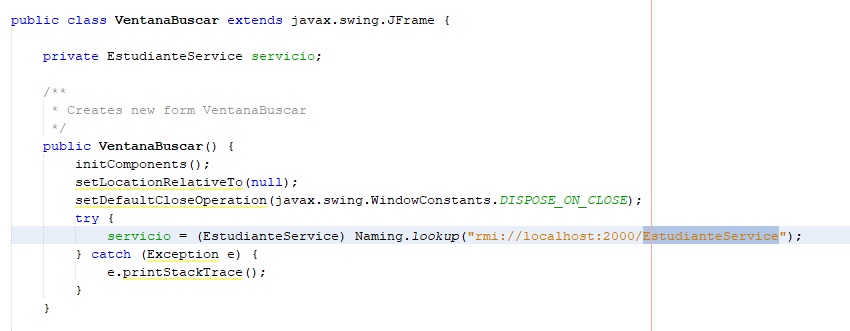


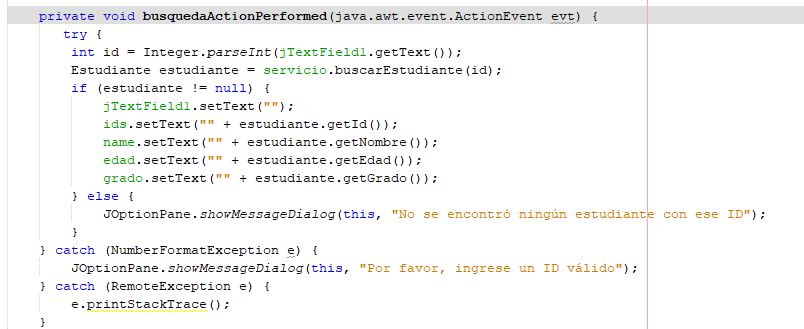
Este código implementa una ventana de actualización en la aplicación de gestión de estudiantes mediante RMI.



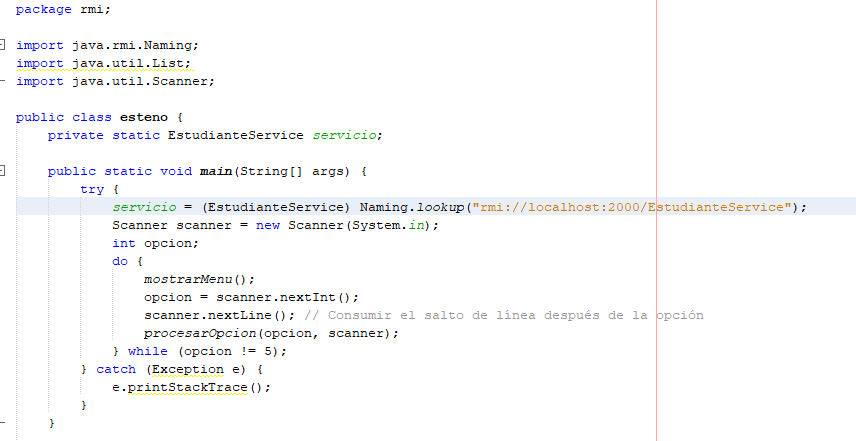


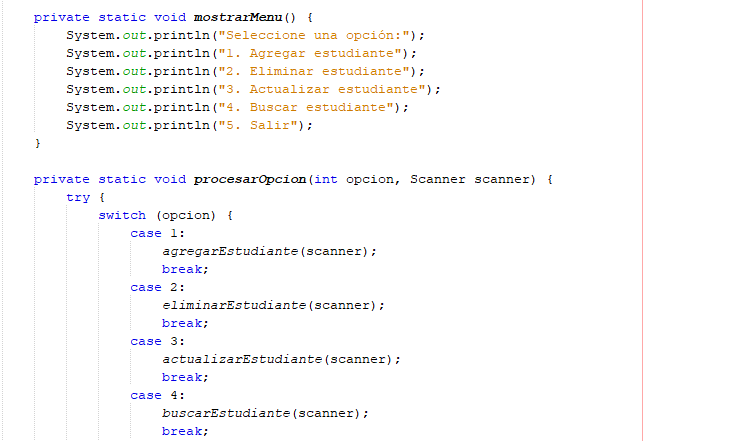
Este código implementa una ventana para eliminar un estudiante de la base de datos utilizando RMI.

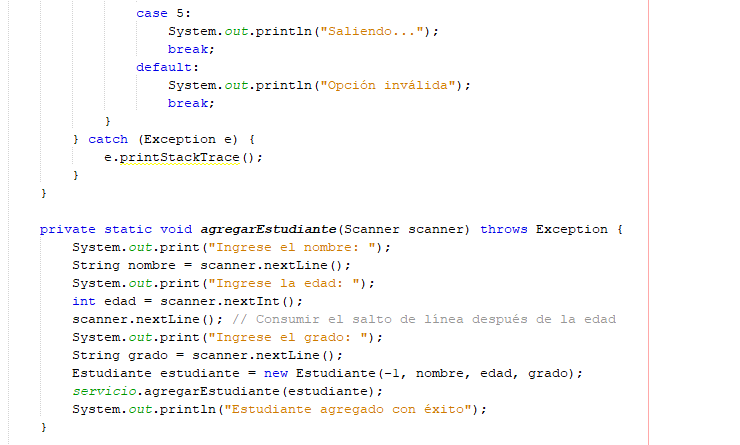


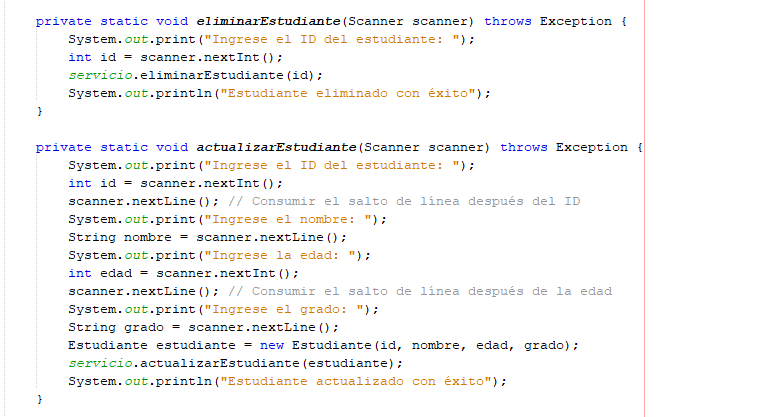


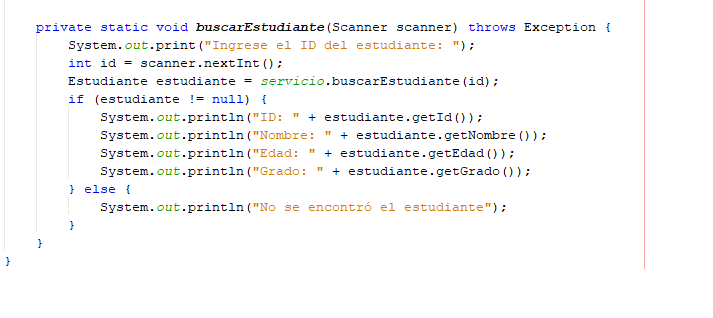
Este código implementa una ventana para buscar un estudiante en la base de datos utilizando RMI.



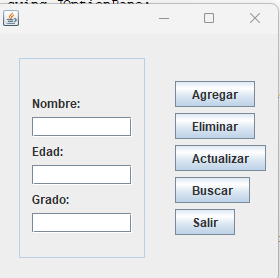


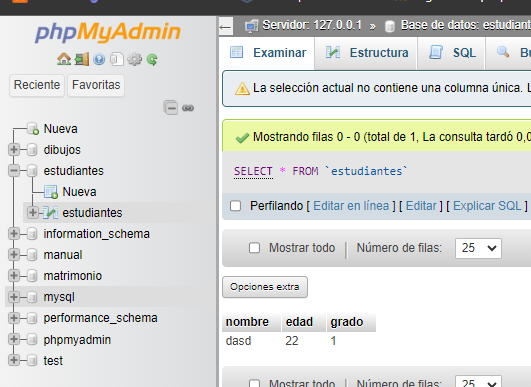




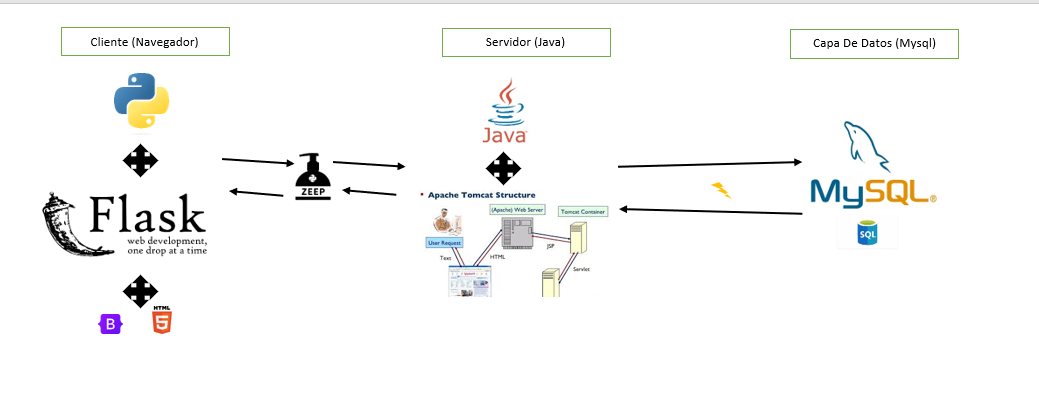


Esta aplicación proporciona una forma de interactuar con un servicio RMI que gestiona estudiantes a través de una interfaz de línea de comandos, permitiendo realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en la base de datos de estudiantes.





# Estructura de datos



# CONCLUSION

El conjunto de códigos proporcionados constituye un sistema de gestión de estudiantes implementado mediante la tecnología RMI (Remote Method Invocation). Aquí hay algunas conclusiones sobre el sistema en su conjunto:

* Arquitectura Cliente-Servidor: El sistema sigue el paradigma cliente-servidor, donde el servidor expone métodos remotos que pueden ser invocados por clientes remotos para realizar operaciones en la base de datos de estudiantes. Los clientes interactúan con el servidor a través de interfaces de usuario (GUI o CLI).
* Comunicación RMI: Se utiliza RMI para la comunicación entre el cliente y el servidor. RMI permite la invocación remota de métodos, lo que facilita la interacción entre las aplicaciones distribuidas.
* Operaciones CRUD: El sistema permite realizar las operaciones básicas de Crear, Leer, Actualizar y Eliminar (CRUD) sobre los registros de estudiantes en la base de datos. Los clientes pueden agregar, eliminar, actualizar y buscar estudiantes utilizando las interfaces proporcionadas.
* Interfaz de Usuario: Se proporcionan interfaces de usuario tanto de línea de comandos (CLI) como gráfica (GUI) para interactuar con el sistema. Esto facilita su uso y hace que sea accesible para diferentes tipos de usuarios.
* Manejo de Excepciones: Se implementa un manejo de excepciones para capturar posibles errores durante la ejecución del sistema, como la falta de conexión con el servidor RMI o errores en la entrada del usuario.