**CAPTURA DEL RESULTADO**



**EXPLICACION DEL CODIGO**

**1. AlumnoBean.java**

**Propósito: Representa una clase de objeto ("bean") para manejar datos de estudiantes ("Alumno").**

**Atributos:**

**numexpdte: Número de identificación del estudiante.**

**nombre: Nombre del estudiante.**

**ciclo: Ciclo de estudios del estudiante.**

**dni: Número de identificación del estudiante.**

**conexionA: Conexión JDBC para operaciones en la base de datos.**

**Métodos:**

**Métodos Getter: Obtienen los valores de los atributos.**

**Métodos Setter: Actualizan los valores de los atributos y realizan actualizaciones SQL correspondientes en la base de datos.**

**Métodos Find: Obtienen información de estudiantes según diferentes criterios (por ejemplo, NUMEXPDTE, ciclo, nombre).**

**Método de eliminación (delete): Elimina un registro de estudiante de la base de datos.**

**Método de inserción (getNuevoAlumno): Crea e inserta un nuevo registro de estudiante en la base de datos.**

**2. AlumnoInterface.java**

**Propósito: Define la interfaz para manejar datos de estudiantes.**

**Métodos:**

**Métodos Getter: Definen métodos para obtener información de estudiantes.**

**Métodos Setter: Definen métodos para actualizar información de estudiantes.**

**Métodos Find: Definen métodos para obtener estudiantes según diferentes criterios.**

**Método de inserción (getNuevoAlumno): Define un método para crear una nueva instancia de estudiante.**

**3. AplicacionAlumno.java**

**Propósito: Contiene el método main para demostrar el uso de las clases relacionadas con estudiantes.**

**Métodos:**

**Método main: Crea una instancia de AlumnoInterface a través de FactoriaAlumnos y crea un nuevo estudiante usando getNuevoAlumno.**

**4. FactoriaAlumnos.java**

**Propósito: Clase de fábrica responsable de crear instancias de AlumnoInterface.**

**Métodos:**

**Método getAlumnoDao: Devuelve una nueva instancia de AlumnoBean.**

**Resumen:**

**AlumnoBean implementa la lógica de manejo de datos para los registros de estudiantes, incluidas las interacciones con la base de datos.**

**AlumnoInterface define el contrato para manejar datos de estudiantes, garantizando una interfaz consistente.**

**AplicacionAlumno demuestra el uso de estas clases creando una nueva instancia de estudiante.**

**FactoriaAlumnos actúa como una fábrica para crear instancias de AlumnoInterface.**