



Taller CRUD de un estudiante

Mateo LLumigusin, David Asmal, Richard Gualotuña

Departamento de Ciencias de la Computación, Universidad de las Fuerzas Armadas ESPE

NRC: 23305

TUTORA Ing. Jenny Alexandra Ruíz Robalino

Fecha: 11/06/2025

1. Objetivo del taller.

Desarrollar en un entorno IDE de Java una aplicación que implemente un CRUD (Crear, Leer, Actualizar, Eliminar) para gestionar datos de estudiantes (orden, nombre y edad), utilizando el patrón de arquitectura en 3 capas. Se busca demostrar cómo interactúan las capas entre sí y cómo se ejecuta la lógica del sistema desde la clase principal (Main.java)

2. Arquitectura (3 capas)

Proyecto Estudiantes

|— Main.java/ → Clase principal

|— datos/

| |— model/Estudiante.java → Modelo de datos

| |— repository/EstudianteRepository.java → Acceso a datos

|— logica_negocio/

| |— EstudianteService.java → Logica de negocio

|— presentacion/

 |— EstudianteUI.java → Interfaz de usuario por consola

- Capas y componentes:

• Capa de datos:

- **Estudiante.java:** Aquí defines la *estructura del datos*, la clase Estudiante con sus atributos (id, nombre, edad). Solo representa la entidad y puede tener métodos simples como getters y setters.

- **EstudianteRepository.java:** Esta capa es la encargada de *gestionar el almacenamiento* de los estudiantes. Aquí se maneja la persistencia, ya sea en memoria (como un HashMap), archivo, o base de datos. Provee métodos para crear, leer, actualizar y eliminar estudiantes (CRUD).
- **Capa de Lógica de negocio**
 - **EstudianteService.java:** La capa de lógica de negocio o servicio. Aquí defines las reglas de negocio y validaciones que aplican a los estudiantes. Esta capa usa el repository para obtener o modificar datos, y expone métodos que la UI puede llamar. // También puede manejar excepciones o flujos lógicos más complejos.
- **Capa de presentación:**
 - **EstudianteUI.java:** La capa de presentación o interfaz de usuario. Aquí está toda la parte visual, en este caso una ventana gráfica con botones, tablas y formularios. Recibe las acciones del usuario y llama a la capa de servicio para procesar datos. Muestra resultados o mensajes basados en lo que pasa en la lógica de negocio.
 - **Main.java:** Punto de entrada del programa. Aquí se inicializa la aplicación y se conecta la UI con el servicio. Controla el flujo principal, arrancando la interfaz gráfica.

3. Conclusión:

El desarrollo del proyecto CRUD de Estudiantes utilizando arquitectura en 3 capas permitió aplicar de forma práctica los principios fundamentales de la programación estructurada y orientada a objetos en Java. Al dividir la aplicación en las capas presentación, lógica de negocio y acceso a datos, se logró una organización clara y modular del código, facilitando su mantenimiento, escalabilidad y comprensión.