

Resolución Práctico 2 B

Grupo 9. Comisión 2

Alumnos: Antequera Gabriel, Bonahora Ignacio, Bosch Gaspar, Boyero Juan, Farias Alexis

Repositorio donde se desarollo la implementación del proyecto a realizar:

https://github.com/JuanBoyero/is1_2025_eti

Análisis del código fuente y patrones de diseño

¿Qué patrón identificaron?

El patrón de diseño que encontramos implementado es un patrón Singleton.

¿Dónde y cómo se aplica en el código?

Este patrón se aplica en la clase implementada: DBConfigSingleton.java .

Donde se establece los métodos correspondientes del patrón: el método public static getInstance() y el creador como private DBConfigSingleton().

Luego en la clase principal App.java se crea un objeto de la clase que DBConfigSingleton.java, este objeto genera la conexión entre una base datos y la funcionalidad de nuestro sistema para poder almacenar los datos de los usuarios y los que se generan a través de las funciones de App.java. (Esto ocurre la línea 41 de App.java)

¿Qué problema resuelve este patrón en ese contexto?

Es claro el uso de este patrón dado que lo que necesitamos es generar una única conexión a nuestra base de datos: dev.db, ya que el patrón identificado tiene la característica de ser un patrón de tipo creacional para asegurar la creación de una única instancia y este objeto ser utilizado de manera global.

Es inteligente y correcto utilizar este patrón para hacer la conexión a la base de datos, ya que nos interesa que esa conexión sea única y que al iniciar el programa nuevamente no ocurra algún error al hacer una corrección en paralelo y/o una inconsistencia en los datos almacenados.

Implementación de Historias de Usuario (HU)

ID de HU	001
Título	Alta de profesor al sistema
Declaración	Como administrador del sistema, quiero registrar un nuevo profesor ingresando su información personal, para poder asignarlo a las asignaturas correspondientes dentro de una carrera.
Descripción Detallada	Cuando deseamos agregar a un nuevo profesor se debe crear una nueva página para realizar la carga de un nuevo profesor, si el profesor ya está cargado en el sistema pero se desea agregar nuevas materias asociadas, no se debe crear un nuevo profesor solo se le deben agregar las materias asociadas.
Criterios de Validación (Criterios de Aceptación)	<ul style="list-style-type: none"> ● Flujo exitoso: Al completar todos los campos obligatorios (nombre, apellido, correo, DNI) con datos válidos y guardar, el sistema muestra un mensaje de éxito. ● Validaciones de Datos: El sistema debe impedir el registro si: <ul style="list-style-type: none"> ○ Faltan campos obligatorios. ○ Si el formato del correo electrónico no es válido. ○ El correo electrónico o el DNI ya existen en la base de datos. ● Manejo de Errores: Si alguna validación falla, el sistema debe mostrar un mensaje de error claro, sin permitir que se guarde el formulario. ● Acción de Cancelar: El formulario debe incluir un botón "Cancelar" que elimine todos los datos ingresados y devuelva al usuario a la pantalla anterior.
Tareas Asociadas a la Implementación	<p>Desarrollar la interfaz de usuario Crear las tablas correspondientes de la base de datos(profesor y curso) Implementar y desarrollar la funcionalidad de la operacion</p>