

Practica 3

Implementacion de patrones de diseño a un proyecto existente.

Implementaremos los siguientes patrones de diseño al proyecto “image-generator-ai-mvc”.

Debemos descargar el proyecto del siguiente enlace:

- <https://github.com/hxadev/image-generator-ai-mvc>

El proyecto image-generator-ai-mvc es un generador de imagenes mediante una sentencia que nosotros escribimos por consola(Prompt), este proyecto está conectado a una inteligencia artificial mediante una API.

Sin embargo el proyecto tiene algunas mejoras para poder estructurar el codigo y volverlo mucho más extensible para las futuras características.

Patrones a implementar

1. Factory Method – Abstract Factory

- Implementa el patron factory en las operaciones **generateClient()** y **generateRequest(String prompt)** que se encuentran en la clase **ImageGeneratorControllerImpl**.
- Genera un **AIGeneratorClientFactory** y su interfaz **ClientFactory** que permita construir clientes con una configuracion inicial.
- Genera un **ImageGeneratorRequestFactory** y su interfaz **RequestFactory** para poder generar objetos para la peticion mateniendo una configuracion inicial.

2. Singleton

- Implementa el patron Singleton en las clases **FilesUtil** y **PropertiesUtil** para mantener una sola instancia de esas clase durante la ejecucion.

3. Strategy

- Implementa el patron Strategy en la clase ImageGeneratorView. Se necesita manegar un Strategy por operacion que puede realizar el usuario. El programa permite 3 operaciones principales “Generar Imagen”, “Mostrar Imagenes guardadas”, “Mostrar imagenes por prompt”.
- Genera un **DisplayImagesByPromptStrategy** para mantener la funcionalidad de “Mostrar imagenes por prompt”.
- Genera un **GenerateImageStrategy** para mantener la funcionalidad de “Generar imagenes”.

Módulo 3 – Principios y patrones de diseño

- d. Genera un **DisplayImagesStrategy** para mantener la funcionalidad “Mostrar imagenes por prompt”.