**Aplicación de Patrones Creacionales y Principio Solid**

***Problema 1: Sistema de renderizado de gráficos en una aplicación de diseño***

* ***Contexto:*** Una aplicación de diseño gráfico permite a los usuarios trabajar con diferentes tipos de objetos visuales: imágenes, texto, formas vectoriales, etc. Cada uno de estos objetos puede tener múltiples representaciones (por ejemplo, en pantalla, en impresión, en miniatura).
* ***Patrón estructural aplicado:***Bridge (Puente)
* ***Justificación:*** El patrón Bridge permite separar la abstracción (el tipo de objeto gráfico) de su implementación (la forma en que se renderiza), de modo que ambos puedan evolucionar independientemente. Esto es útil cuando:
  + Hay múltiples combinaciones de objetos y formas de renderizado.
  + Se desea evitar una explosión de clases como TextoEnPantalla, TextoEnImpresión, ImagenEnPantalla, etc.
* ***Diagrama UML:***

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

***Problema 2: Sistema de gestión de archivos en la nube con estructura jerárquica***

* ***Contexto:*** Una plataforma de almacenamiento en la nube necesita representar archivos y carpetas. Las carpetas pueden contener archivos y otras carpetas, formando una estructura jerárquica.
* ***Patrón estructural aplicado:***Composite (Compuesto)
* ***Justificación:*** El patrón Composite permite tratar objetos individuales (archivos) y composiciones de objetos (carpetas) de manera uniforme. Es ideal cuando:
  + Se necesita representar estructuras jerárquicas.
  + Se desea que operaciones como mover, eliminar, o mostrar funcionen igual para archivos y carpetas.
* **Diagrama UML:**

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Link github:

<https://github.com/KevinAPerez/Ptrones_Diseno_Software/tree/main/Patrones%20Estructurales%20y%20Principio%20Solid>