

ZFCócteles: Patrón de Diseño 🍹

Patrón de diseño elegido: Builder 🏠

El Builder es un patrón de diseño creacional que permite la construcción de objetos complejos paso a paso 🧑. Es ideal cuando necesitamos crear un objeto con múltiples configuraciones, pero queremos evitar constructores sobrecargados con demasiados parámetros opcionales.

Para lograrlo, se separa el código de construcción del objeto de su clase principal, dividiéndolo en segmentos independientes. Cada uno de estos se encarga de añadir características específicas al objeto, manteniendo el proceso flexible y organizado ✅.

🔍 Justificación y contexto:

Un cóctel 🍹 puede ser un objeto de alta complejidad, ya que su preparación incluye:

- Ingredientes variables 🍋🍌🍓
- Decoración 🍹🌟
- Presencia (o no) de alcohol 🚫🍷
- Categoría (clásico, tropical, sin alcohol, etc.) 🏷️
- Recetas modificadas o preexistentes 📄

El patrón Builder es perfecto para este caso, ya que permite:

- ✅ Construir el objeto en etapas claras.
- ✅ Evitar un constructor gigante y difícil de mantener.
- ✅ Controlar el proceso de preparación, modificación y creación de manera ordenada.

💻 Implementación:

Cada cóctel no solo será una tabla en SQLite, sino un objeto con propiedades como:

- Nombre 🏷️
- Ingredientes 🍹
- Categoría 📌
- Imagen 🖼️
- Instrucciones 📄

En lugar de usar un constructor monolítico, se utilizarán métodos encadenables (setters) o pasos específicos para construir el objeto de manera modular.

🚀 Uso:

Actualmente, no se puede mostrar el código implementado, pero la estructura del proyecto ya refleja este patrón:

- Se ha creado la carpeta `renderer/builders/`, donde se alojarán las clases Builder de las entidades principales.

Esto demuestra que el proyecto sigue una arquitectura limpia y escalable, preparada para manejar la complejidad de los cócteles de manera eficiente 🎯.

