Gerónimo Soto Leslie - 320032848
Lozano Estrada Yael - 319007095

PATRÓN DE DISEÑO: ADAPTER

PROBLEMA:

Imagina una situación en la que tienes un sistema de banca en línea que gestiona la información de las cuentas bancarias, incluyendo números de cuenta, nombres de los titulares, tasas de interés y anualidades. Este sistema de banca en línea está en funcionamiento y se utiliza en una institución financiera.

Luego, tienes un cliente, que es una aplicación o componente de software independiente, que necesita interactuar con la información de estas cuentas bancarias para realizar diversas tareas. Sin embargo, el cliente espera una interfaz específica que sea coherente con su propio diseño y requerimientos. La interfaz del sistema de banca en línea y la interfaz del cliente son incompatibles, lo que significa que no pueden comunicarse directamente.

En este contexto, el patrón de diseño Adapter se convierte en una solución valiosa. Aquí está cómo se relacionan las partes en este problema:

Cliente: Representa la aplicación o componente que necesita acceder a la información de las cuentas bancarias. Este cliente tiene una interfaz específica que espera para interactuar con los datos de la cuenta.

Banca en línea: Es el sistema existente que almacena y gestiona la información de las cuentas bancarias. Tiene su propia interfaz que no coincide con la interfaz del cliente. Puede proporcionar detalles de cuentas bancarias, como números de cuenta, nombres de titulares, tasas de interés y anualidades, pero no en el formato requerido por el cliente.

Adapter: El Adapter es una clase intermedia que actúa como un puente entre el cliente y el sistema de banca en línea. Tiene dos tareas principales: a) implementar una interfaz compatible con el cliente, y b) utilizar la interfaz del sistema de banca en línea para obtener los datos necesarios y convertirlos en un formato que el cliente pueda entender. En esencia, adapta la información proporcionada por la banca en línea para que sea coherente con las expectativas del cliente.

¿Por qué elegimos este patrón de diseño?

El patrón de diseño Adapter es una buena opción en este problema debido a que se requiere la compatibilidad entre dos interfaces incompatibles. En este caso, tienes un "Cliente" que necesita obtener información de una "Banca en línea" que tiene su

propia interfaz, pero el "Cliente" espera una interfaz diferente. El Adapter actúa como un intermediario que adapta la interfaz de la "Banca en línea" a la interfaz que el "Cliente" pueda entender.
El patrón de diseño Adapter permite que el cliente utilice la funcionalidad y los datos del sistema de banca en línea sin modificar el código fuente del sistema de banca en línea. Esto es especialmente útil en situaciones en las que no se puede cambiar el sistema de banca en línea, como cuando se trata de un sistema heredado o proporcionado por terceros.

Además, el uso del patrón Adapter facilita la separación de responsabilidades, ya que la lógica de adaptación está encapsulada en el Adapter, lo que hace que el código sea más modular y mantenible.