

Segunda Refactorización

¿Se implementa un patrón de diseño?

- Al evaluar el código no se puede encontrar similitudes con el patrón strategy ya que el uso de interfaces no está presente en la aplicación.
- En el archivo donde se encuentra la entidad (Tarjeta de crédito) se puede observar que no existen diversos constructores lo que permite poder implementar el patrón builder, por lo que no lo tiene, además de que no implementa una clase padre por lo que el factory tampoco está presente.

Por lo que esta aplicación no se pudo identificar un patrón de diseño.

¿Está el código altamente acoplado?

Al tener varias clases con un propósito único, se puede decir que el código no está altamente acoplado.

¿Es testeable?

Parcialmente, ya que el formato y nombre que tienen los archivos ayuda a poder identificar su propósito y existen ciertos comentarios en ellos donde también se identifica que es lo que se realiza.

¿El código se puede leer?

El nombre de las variables se entiende que es lo que almacenan y los nombres de los métodos también son entendibles.

¿El código está duplicado?

Si, principalmente en los archivos que están en números de tarjetas donde se verifica, el código se repite en su mayoría a diferencia de una línea en donde se asigna qué archivo que se evaluara (numeroNacional, numeroInternacional, numeroRegional)

¿El código es muy largo para entenderlo?

No, la mayoría de clases no exceden las 100 líneas a diferencia de las clases de la UI, se podría decir que los métodos donde la intención es realizar el reporte html son un poco largas pero es por la sintaxis que lleva el reporte.