TRABAJO PRÁCTICO FINAL

PROGRAMACIÓN ORIENTADA A OBJETOS II -
2° CUATRIMESTRE 2022

PROFESORES: Diego Cano - Matias Butti - Diego Torres

ALUMNOS:

BARON,ELIAS

eliasnbaron@gmail.com

PALAVECINO, FLORENCIA

fpalavecino743@gmail.com

Decisiones de diseño:

Procedimos a leer varias veces el enunciado y las consignas del trabajo.
Luego diseñamos un UML intentando hacerlo lo mejor y detallado posible, ya que esa fue nuestra base para comenzar con la implementación del código en Java.
Al momento de iniciar la implementación, creamos las clases y luego fuimos desarrollando cada una de ellas, dejando los test para el final.
Con los cuales, nos fuimos dando cuenta de los errores, y corrigiendo a medida que los íbamos desarrollando.

Detalle de implementación:

- Decidimos hacer los DesafioUsuario como una especie de observadores del Usuario.

- En el Usuario agregamos setDesafiosUsuario(List<DesafioUsuario>), el cual utilizamos exclusivamente para los tests.

- En algunos tests(restricciones temporales),los dejamos implementados sin Mockito, ya que fue una herramienta que aprendimos a utilizar en el transcurso del trabajo.

- En Usuario, en algunos métodos, por ejemplo CalificarDesafio(DesafioUsuario) tuvimos la duda de si por parámetro debiamos pasarle un Desafio o un DesafioUsuario,finalmente nos

decidimos por pasarle un DesafioUsuario para abrir la posibilidad que el usuario pueda aceptar un desafío varias veces y pueda tener distintos progresos.

Patrones de diseños utilizados:

En la implementación nos encontramos con los siguientes patrones:

- En la clase Filtro, aplicamos el Patrón Composite. Las clases And, Not y Or son los Composite y las clases BusquedaPorTitulo,ExclusionPorCategoria y BusquedaPorCategoria son las Leaf. La clase Filtro es el Component.

- En Usuario con TipoRecomendacion, utilizamos el Patrón Strategy.

Usuario es el Context, TipoRecomendacion es la Strategy y PreferenciaJuego y Favorito son las ConcreteStrategies.

- En DesafioUsuario, utilizamos el Patrón State. El DesafioUsuario es el Context, el EstadoDesafioUsuario es el State y DesafioCompleto, DesafioAceptado y DesafioNoAceptado son las ConcreteStates.

- DesafioUsuario lo planteamos como un Observer, por ende el Usuario es el Observado y DesafioUsuario es el Observador.