

Programación con Objetos 2

Trabajo Final

Ciencia participativa y juegos

- Agustin Campos -
Email : agustin.campos17@outlook.com
- Ezequiel Gonzalez -
Email: ezegonzalez912@gmail.com
- Federico Hernan Sanchez -
Email : federicosanchezunq@gmail.com

Patrones de diseño

State -

Se utilizó el patrón state en la clase Recomendación, la cuál es extendida por PreferenciasEnJuego y Favorito, ya que estos estados

responden a su manera a la forma de recomendar desafíos para el usuario

Decisiones de diseño

Usuario

La clase Usuario posee una lista de proyectos, en los cuales está , una lista de muestras que recolectó , su perfil de usuario con sus diferentes atributos que sirven para poder comparar con los desafíos y calcular la coincidencia de ese desafío y un atributo recomendación que sirve para indicar en qué estado se encuentra el usuario actualmente.

Desafío

En la clase Desafío se determinó que contenga un atributo de tipo RestriccionTemporal, de la cual extiende a tres clases (EntreFechas, DuranteSemana, DuranteFinDeSemana), así cada clase puede determinar su manera de restricción.

Proyecto

Para la lógica del proyecto, decidimos crear una clase del mismo nombre, del cual sabemos su nombre, descripción, categorías, desafíos y sus usuarios activos. Dentro de lo más significativo de la clase tenemos un patrón **observer** los proyectos tienen usuarios suscriptos que son notificados por cada desafío agregado en el proyecto.

Filtros

Para crear la lógica de Filtros, decidimos implementar el patrón **Composite** el cual nos permite agrupar de forma uniforme objetos primitivos (Leaf) y complejos (Composite). Llegamos a la conclusión de que deberíamos usar este patrón porque aparte de tener filtros básicos, también existen filtros compuestos por las operaciones lógicas AND y OR.

Participantes Del Patrón.

- **IFiltro**: Componente, es la interfaz que declara el protocolo común.
- **FiltroIncluirTexto**, **FiltroIncluirCategorias**, **FiltroExcluirCategorias**: son los objetos primitivos del patrón (Leaf's).
- **FiltroCompuesto**: Composite, este es la clase abstracta que abstrae el comportamiento común de los filtros compuestos, de este extienden **FiltroAND** y **FiltroOR**.