



Trabajo Grupal PO2: “Ciencia Participativa y Juegos”

Integrantes:

- | | | |
|---|---------------------------|-------------------------------------|
| - | Provvidenza Ivana | vanndere@gmail.com |
| - | Salas Marcos Nahuel | nahuio@gmail.com |
| - | Villarroel Julián Esteban | villarroel.julian.esteban@gmail.com |

Link al repositorio: <https://github.com/athanop/unqui-po2-tpFinal>

Decisiones de diseño:

En el siguiente documento explicaremos las decisiones de diseño tomadas al inicio del desarrollo del trabajo, así como también, los patrones utilizados y los detalles que merecen ser destacados.

Nuestro primer paso en el desarrollo del diseño fue realizar un diagrama UML con las clases principales, buscando entender sus responsabilidades así como también sus interacciones con las demás. Durante esta tarea designamos los mensajes que cada clase debía comprender, sus colaboradores internos y los patrones que se podían aplicar sobre algunas de ellas.

A continuación, se detallan aquellas clases que destacan alguna característica particular o forman parte de uno de los patrones aplicados:

Sobre el Sistema (Patrón Singleton):

El sistema es implementado como un Singleton para asegurar que la clase tenga una única instancia y brindar un punto de acceso global a esa única instancia. Este sistema se encarga de gestionar y dar de alta los proyectos y categorías para poder realizar búsquedas de proyectos utilizando distintos filtros.

Sobre las restricciones temporales en los desafíos (Patrón Composite):

Como se requiere restringir los intervalos de tiempo en dónde son válidas las admisiones de las muestras enviadas para cumplir los desafíos se incorporó al modelo el patrón Composite para poder realizar un filtrado complejo combinando los distintos criterios pedidos. Los roles quedan divididos del siguiente modo:

- RestriccionTemporal: es la interfaz Component.
- RestriccionComposite: es el objeto compuesto Composite.
- RestriccionFecha, RestriccionDiasDeSemana, RestriccionFinDeSemana: son las hojas u objetos simples.

Sobre las búsquedas de muestras en el sistema (Patrón Composite):

Como se requiere realizar búsquedas de proyectos basadas en filtros se incorporó al modelo el patrón Composite para poder realizar filtros complejos combinando los distintos criterios pedidos y las operaciones lógicas NOT, OR y AND. Los roles quedan divididos del siguiente modo:

- BusquedaComponent es la interfaz Component.
- ANDComposite, ORComposite, NOTComposite: son los objetos compuestos Composite.
- FiltroPorTexto, FiltroPorCategorialInclusion, FiltroPorCategorialExclusion: son las hojas u objetos simples.

Sobre los estados del desafío de usuario (Patrón State):

Para el desarrollo de los estados del Desafío (Incompleto - Completo) utilizamos el patrón de diseño State.

Asumimos que el estado incompleto es un desafío que fue aceptado por el usuario, por lo cual, no encontramos la necesidad de poner algún otro estado (así como por ejemplo: estado pendiente, estado aceptado).

Los roles quedan divididos del siguiente modo:

- EstadoDesafioUsuario: es el rol State.
- DesafioIncompleto: es el rol ConcreteState.
- DesafioCompleto: es el rol ConcreteState.

Sobre las recomendaciones de elemento de ludificación (Patrón Strategy):

Utilizamos el patrón Strategy para la forma de recomendación que puede escoger un usuario en base a sus preferencias.

Los roles quedan divididos del siguiente modo:

- Recomendación: esta clase abstracta es el Strategy.
- PreferenciasEnJuego: es el rol ConcreteStrategy.
- Favoritos: es el rol ConcreteStrategy.