Informe reentrega del trabajo Práctico

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

Ayudante que nos corrigió:

- Fabrizio Britez

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

04/12/22

Cambios realizados:

Sistema

Patron Singleton está bien explicada la intención, pero no es correcto utilizarlo para poder ser llamado desde cualquier punto del sistema. Esto va en contra de un buen diseño de conocimiento entre objetos. **Eliminamos el patrón Singleton.**

DesafioUsuario

votoDelUsuario: No delegan en el estado el comportamiento, espera la valoración o lanzar la excepción. **Se le delegó al Estado el comportamiento del voto del Usuario.**

muestrasValidas Nombre de mensaje poco cohesivo. Nombre correcto debería de indicar que devuelve un número. **Se renombró adecuadamente:**

- **cantidadDeMuestrasValidas(Usuario usuario)**

Usuario

buscarDesafios: No es necesario crear una lista y agregar los valores del resultado delatara a recomendación.seleccioDeDesafios. Devolver directamente el resultado de recomendación.seleccioDeDesafios. **Se eliminó el ArrayList que guardaba los resultados y ahora se devuelven directamente.**

aceptarDesafio. No es necesario validar que el desafío exista en la lista.
Porque no debería darse el caso de que el usuario acepte dos veces el mismo desafío. **Se eliminó la validación.**

agregarMuestra: Debería de tener algún tipo de interacción con los desafíos de usuario para que se actualicen y sepa si el desafío está cumplido o no. No hay ningún objeto que llame al mensaje actualizarDesafio por lo que nunca se haría el pasaje de estados. Esa lógica debería de ejecutarse al recibir la muestra número N que cumple con la condición de aceptación. **Se implementó que al agregar una muestra se le notifique a los desafíos que se actualicen las muestras tenidas en cuenta en los desafíos donde está participando el usuario.**

Es necesario tener un mensaje en promedioDeCompletitudGeneral que devuelve el porcentaje de completitud que tiene el usuario sobre todos los desafíos terminados y en curso. Delegar en el Desafío el porcentaje y hacer los cálculos correspondientes.
Creamos el mensaje que devuelve el porcentaje de completitud general de los desafíos del usuario.

Los usuarios no están interesados en conocer a qué proyectos aportan. En todo caso el proyecto está interesado en los usuarios. Deberían de re-diseñar esta parte.
La manera que abordamos, luego de conversar en clase al respecto, fue eliminar la responsabilidad que tenía el usuario de conocer los proyectos y hacer que el sistema devuelva los proyectos donde está participando un usuario. Los que nos llevó a modificar el Strategy de recomendaciones el cual ahora obtiene los proyectos del usuario desde el sistema.

- **getProyectosDeUsuario(Usuario usuario)**