

## Práctica 1.

Equipo: **SyntaxError**

Integrantes:

Centeno Lopez Shani Samantha	319032680
Hernández Barrios Carlos	321297374
Mendoza Romero Fabian	321269568

Aplicamos el patrón Strategy para manejar los diferentes comportamientos de ataque y defensa de los distintos personajes, ya que, éstos necesitan cambiar dinámicamente su forma de pelear cuando adquieren nuevos poderes. Aplicar el patrón Strategy nos permitió encapsular algoritmos en clases separadas, facilitando la adición de nuevos poderes sin modificar las clases de personajes existentes. Además de que los poderes pueden ser reutilizados por diferentes personajes o podemos agregar más personajes si fuera necesario.

También, aplicamos el patrón Observer para manejar la transmisión de los combates, ya que, múltiples espectadores necesitan ser notificados de los eventos del combate y existe una relación de uno a muchos entre el combate y los espectadores, lo cual es el principio de Observer. Esto, además, nos permite que el Combate no necesite conocer los detalles internos de cada Observador, solo sabe que implementan la interfaz Espectador y tienen un método actualizar(), lo que nos permite agregar más espectadores si lo deseamos.

Para poder visualizar el combate, el usuario solo tiene que ejecutar el archivo: SimuladorBatalla y con eso podremos visualizar en consola el desarrollo de alguno de los tres casos de prueba sobre los combates, además de que se crearán simultáneamente los archivos terminación .txt, personalizadas para cada espectador.

En teoría, el archivo funciona sin problemas, esperamos que no tengan problema para ejecutarlo. En cuanto al diagrama, Álvaro nos dijo que la punta de la flecha era con ese estilo (--) porque tuvimos confusión entre los ejemplos de los tres, con cuál era la terminación correcta de la flecha, espero que no tengamos problema con ello. Finalmente, las versiones de java usadas para la elaboración de la práctica fueron: 21, 24, 25.