**Reflexión acerca del uso del patrón Singleton y Observer en mi proyecto**

El patrón singleton al igual que el patrón oberver dentro de mi proyecto tuvieron una relevancia importante, el patrón singleton me permitió generar únicas instancias de objetos para posteriormente usar en otros lugares, esto por ejemplo para invocar métodos de una interfaz desde otra clase es perfecto ya que así se guardan las referencias de esa interfaz, o por ejemplo en la utilización de un solo stage para cargar todas las interfaces ahí, ellas debían de tener una sola referencia a la stage, es por ello que decidí hacer esta referencia singleton, así siempre se apuntaría hacia el mismo objeto y esto es algo tan relevante que sin este objeto stage, no tendría en el proyecto cambio de interfaces.

A su vez, el patron listener me permitió poder notificar durante todo el programa que interfaces usar, cuando conectarme, cuando obtener mensajes, acabar el juego entre muchas cosas, gracias a este patrón el programa tenía una secuencia lógica, el programa sabia en todo instante que ejecutar después de cierto proceso y no generaba así llamamientos infinitos a un proceso esperando un cambio o luz verde para ejecutar otro proceso, esto conocido como long polling.