# **INFORME PATRONES**

Para la implementación del Pokémon se han utilizado tres patrones diferentes, el Observer, ya mencionado en la entrega anterior, y otros dos nuevos.

Por un lado tenemos el Factory, este patrón se utiliza para crear objetos de manera dinámica y flexible en situaciones en las que se necesitan crear muchos objetos que tienen implementaciones ligeramente diferentes o que dependen de la configuración o el contexto de la aplicación. Más adelante explicaré como lo hemos implementado en nuestro proyecto.

Y por otro lado tenemos el patrón State, este patrón State se utiliza para modelar comportamientos que cambian en función del estado interno de un objeto. Este patrón se utiliza comúnmente en situaciones en las que se necesitan implementar comportamientos que dependen del estado de un objeto, como en las máquinas de estado finito, los sistemas de control de procesos y los protocolos de comunicación.

## Patrón Factory

## 

## Patrón State

