**PATRONES DE DISEÑO**

**Singleton:**

Propósito:

* Provee una forma de acceder a las instancias.
* Se asegura de que no se crean muchas instancias de una clase.
* Sirve como un único punto de acceso a la variable.

**Abstract factory:**

Las clases concretas deben tener los mimos métodos de la clase abstracta, pero tiene parámetros diferentes.

**Facade:**

Definición:

Provee una interfase unificada para varias interfases en un sistema. La fachada define una interfase de alto nivel que hace más fácil de usar un subsistema.

Por ejemplo, para acceder a los elementos internos de una casa, por ejemplo, se hace uso de la fachada, como lo es el frente de la casa.

**Adapter:**

Convierte la interfase de una clase en la interfase que el cliente espera. Deja a las clases trabajar juntas si estas no pueden hacerlo entre sí, debido a interfases incompatibles. Ejemplo:

CELULAR 🡪 USB 🡪 ADAPTADOR 🡪 TOMA DE CORRIENTE ELECTRICA

El celular hace uso del USB, el cual tiene un adaptador que permite hacer uso de la corriente eléctrica para cargar la batería.

**Observer:**

Define una dependencia entre objetos de uno a muchos, de manera que cuando un objeto cambia de estado, todos sus dependientes son notificados y actualizados automáticamente. Ejemplo:

En “YOU TUBE” cuando alguien sube un video hay un observador que notifica a los subscriptores sobre ese nuevo video.