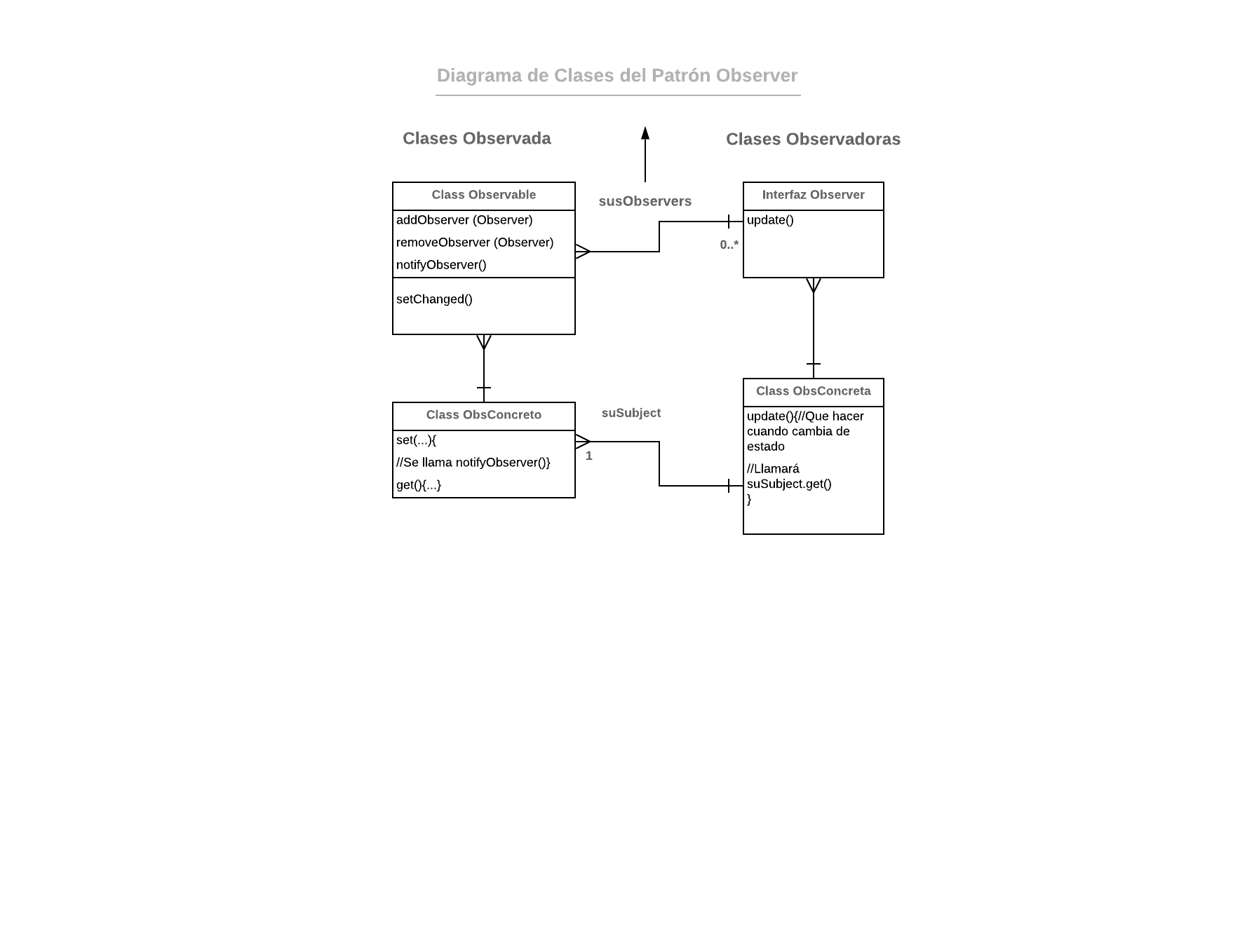
**Patrón de Diseño Observer**

El patrón de diseño **observador**permite tener una dependencia de uno a muchos objetos de una manera libre, sin estar fuertemente acoplados, consiguiendo que un objeto observado, llamado **sujeto**, al cambiar su estado, todos los demás objetos interesados en estos cambios sean notificados y actualizados automáticamente. A estos objetos interesados se les llama **observadores**y pueden recibir del sujeto datos relacionados sobre los cambios.



|  |  |
| --- | --- |
| Métodos | Significado |
| addObserver(o) | Añade un nuevo observador |
| removeObserver(o) | Elimina observador o |
| setChanged() | Marca el objeto como modificado |
| hasChanged | Revisa si el objeto esta modificado |
| notifyObserver | Notifica a todos los objetos suscritos si el objeto esta en el estado modificado |
| Update() | Ejecuta lo predicho si un objeto cambia de estado |

En el patrón de diseño Observer como se menciona anteriormente , se usan los observadores y los observados ,en el diagrama de clases se puede observar como se tiene una función de **update** la cual actualiza el estado de lo que esta sucediendo en el observado en el momento de que algún elemento se edite o cambie de estado usando el **setChanged**, lo actualiza notificando a los observados lo sucedido y para esto utiliza el método **notifyObserver** y se ejecuten las acciones correspondientes al cambio de estado, los observados se enlazan con un observador con el método **addObserver** y para eliminar ese enlace de utiliza **removeObserver**.

**Ejemplos de implementación del patrón**

Un ejemplo es un sistema de compras, varios clientes tienen su interfase como observado si uno selecciona un producto único se deberá actualizar la página y notificar el cambio de estado y ahí entra el observador que actualiza las demás interfases u observados cambiando sus estados y en este caso eliminando o reservando el producto que se acaba de comprar.

Otro ejemplo es cuando se utiliza un lector de huella digital, como un seguro de puerta, suponiendo que este es el observador, el observado vienen siendo lo que reciba la señal del seguro, cuando se pone el dedo en el lector, se actualiza como cambio de estado notificando el cambio de estado, el observado valora el estado y si coincide con la huella guardada responde abriendo la puerta.