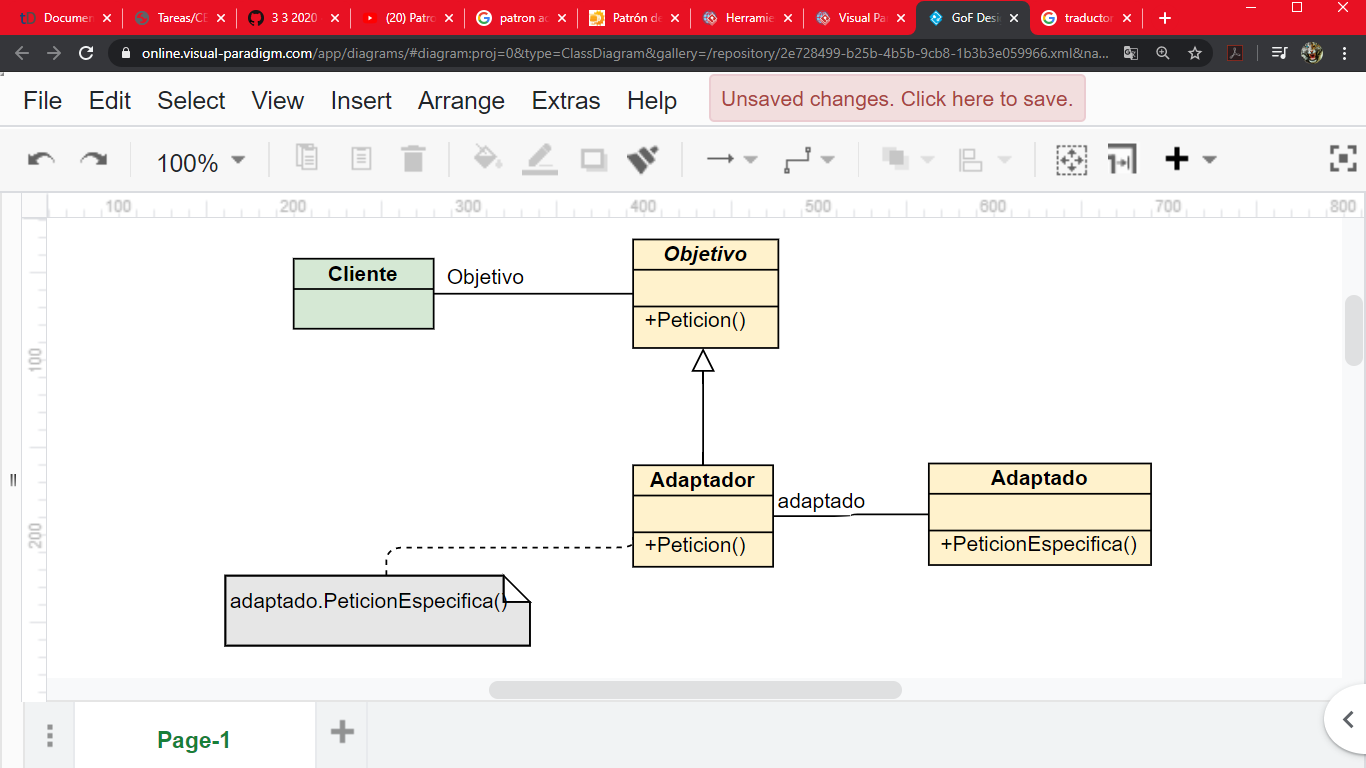
**Adapter:**

**Propósito:**

Convertir o transformar la interfaz de una clase en otra interfaz que el cliente espera. El adapter permite a las clases trabajar juntas, lo que de otra manera no podría hacerse debido a sus interfaces incompatibles.

**Estructura:**

**<<Interface>>**



Use

Extends

Contains

En el diagrama, como se puede observar, tenemos un cliente, el cual es el actor que interactúa con el adaptador. Un objetivo que es la interface que nos permitirá trabajar de forma general con las interfaces incompatibles, la cual es utilizada para crear los adaptadores. Luego el adaptador, el cual representa la implementación del Objetivo, el cual es el encargado de mediar entre el Cliente y el adaptado además, es el que se encarga de adaptar la interfaz del Adaptado a la interfaz Objetivo. Y por último, el adaptado el cual representa la clase con interface incompatible.

**Contexto:**

* Este patrón es muy útil cuando se desea usar una clase existente, y su interfaz no sea igual a la necesitada.
* También, otro caso es cuando se desea crear una clase reutilizable que coopere con clases que no están relacionadas entre ellas. Es decir, que las clases no tienen necesariamente interfaces compatibles.