Herencias typescript

Herencias:

Typescript nos permite crear las clases que queramos tener para nuestros proyectos, en ocasiones debemos usar campos específicos dependiendo del tipo de información que manejemos, al crear una clase, y utilizarla en un index, estamos dando al elemento del proyecto en cuestión el tipo de esa clase que hemos diseñado.

Por otro lado existen las herencias, lo cual está estrechamente relacionada con dichas clases, ya que podemos crear clases que deriven de otras, por ejemplo, en este ejercicio existen tres clases, la primera que es “Relojes” y dos derivadas de ella:

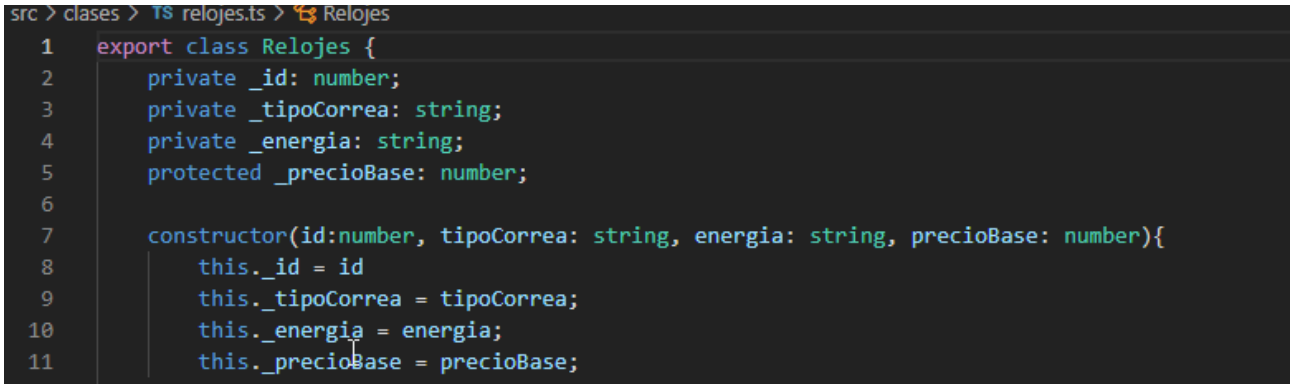
-Digitales

-Mecánicos

Polimorfismo:
Sucede cuando se pueden obtener diversos resultados dependiendo de la clase a la que nos refiramos, los métodos pueden ser sobrecargados cuando existe funciones con el mismo nombre y con funcionalidad similar, pero pertenecientes a clases independientes.

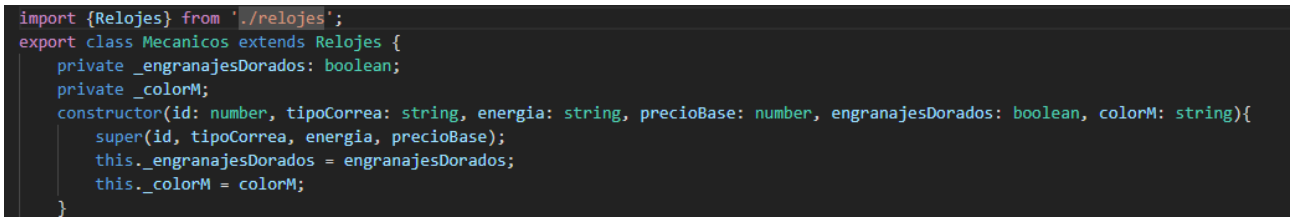
Ejemplos:

Estas tres clases comparten algunos campos pero las clases digitales y mecánicos se diferencian en un par de campos exclusivos de ellos:



Como podemos observar en la imagen, se definen los campos de la clase Relojes, y se ponen en privado, salvo en el caso de precioBase que esta en estado “protected” lo cual significa que las clases que heredan todos estos campos, pueden ver y usar los campos privados en sus propios campos, pero no modificar la información original.

Para heredar una clase, debemos crear una clase que será la “hija” y en ella importamos el contenido de Relojes, a la vez que exportamos nuestra clase como una extensión de la clase Relojes.



Y en el index tendremos que importar todos los elementos que vamos a usar.