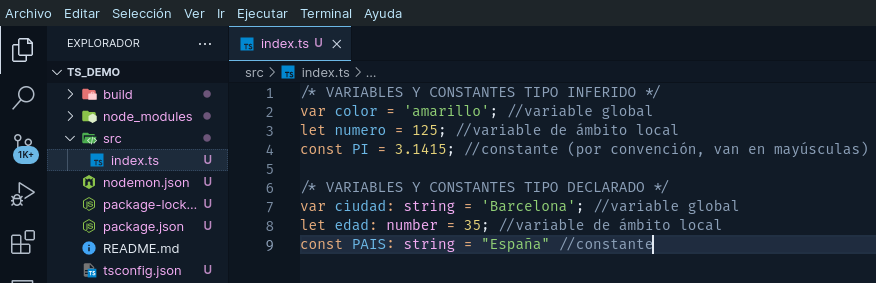
I – SINTAXIS Y VARIABLES

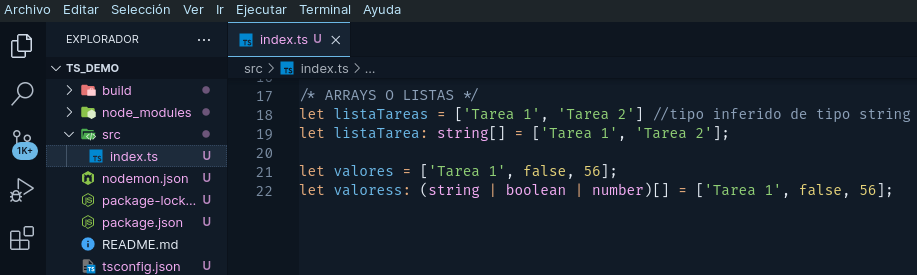
En TypeScript utilizaremos la sintaxis de **ECMASCRIPT** (en el apartado anterior hemos definido **ECMASCRIPT6**). TypeScript es de tipo inferido, por lo que las variables pueden declararse sin especificar el tipo, aunque es recomendable especificarlo:



La concatenación se puede realizar de tres formas diferentes, la más recomendada es la que, a continuación, resaltamos:

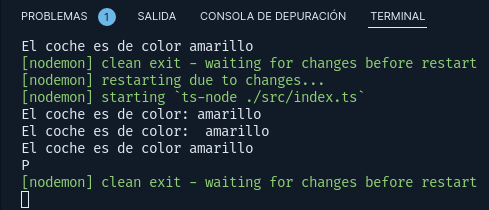


En los arrays ocurre lo mismo (son de tipo inferido), pueden ser del mismo tipo o de tipos combinados:

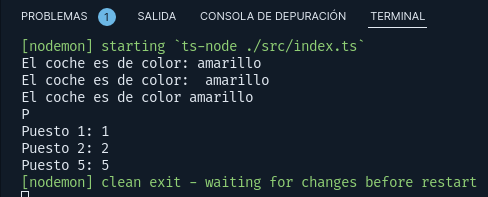


Las listas o arrays pertenecen al grupo de tipos complejos, siendo los tipos simples los llamados **tipos primitivos** (string, number, boolean, void, null y undefined).

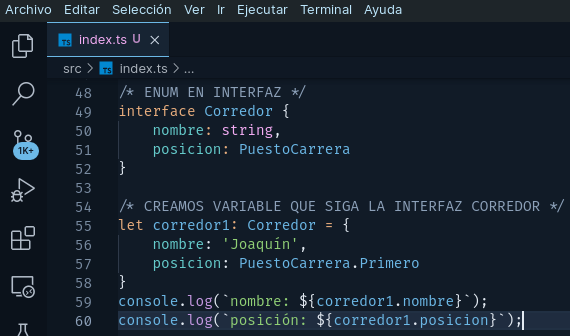
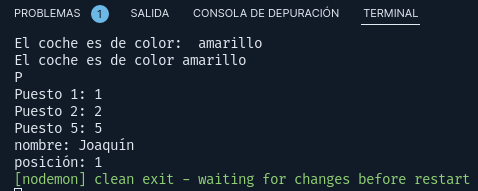
También podemos crear variables con tipos complejos, para ello, utilizaremos los **enumerados** (enum). En el siguiente ejemplo, podemos comprobar como la variable **estado** recoge el valor de la key **Pendiente**, siendo ésta una variable de tipo **Estados**:



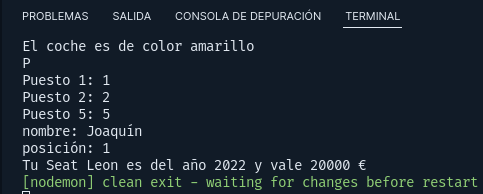
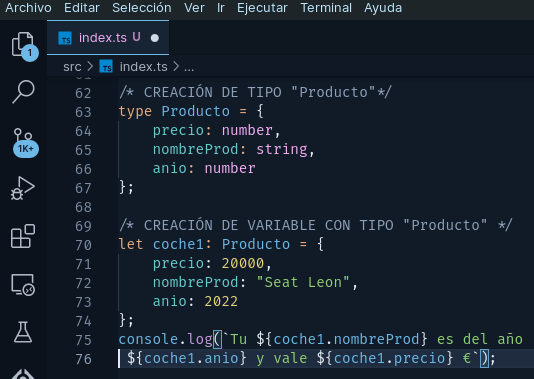
Al crear un enum cuyos valores sean números, al indicar el valor inicial en la **primera key**, si omitimos el valor en las siguientes, el valor sería valor inicial + 1:



Podemos utilizar **enum** en las **Interfaces**, de esta forma, podemos crear **variables** que, a su vez, sigan **Interfaces** definidas con **enum**:



Por último, podemos crear tipos de datos complejos a través del objeto **type**. De esta manera, podremos crear una variable, una interfaz... Que trabaje con este tipo complejo:



II – ESTRUCTURAS DE CONTROL