CLASE 2 - Práctica -

1. Métodos de arrays

```
var productos = [
      {nombre: 'mouse', categoria: 'computadora', cantidad: 2},
      {nombre: 'teclado', categoria: 'computadora', cantidad: 0},
      {nombre: 'almohadon', categoria: 'deco', cantidad: 0},
      {nombre: 'cartera', categoria: 'moda', cantidad: 0},
      {nombre: 'iphone 8', categoria: 'celulares', cantidad: 5}
];
```

a. Imprimir los nombres de todos los productos por consola.
b. Imprimir cantidad disponible de iphone 8.
c. Aumentar 2 cantidades en todos los productos.
d. Imprimir solo los productos que pertenecen a la categoría computadora.
e. Imprimir stock total.

2. Funciones

a. Definir una función que reciba un número X e imprima los primeros X números de la sucesión Fibonacci. Los números de Fibonacci deben ser generados por la función.

0,1,1,2,3,5,8,13,21,34,55,...

b. ¿Qué creen que imprimirá por consola el siguiente código? ¡No ejecutar el código!

```
var frutas = ["Cereza", "Manzana", "Melón", "Frutilla"];
for(var i = 0; i <= frutas.length; i++){
      setTimeout( function(){
            console.log(frutas[i]);
      }, 2000);
}.
```

c. Ejecutar el código anterior. ¿Qué imprime por consola y por qué?
d. ¿Cómo harían para que imprima por consola cada elemento del array cada 2 segundos?

3. **Callback**

a. Consultar datos de un pokemón haciendo llamado ajax a pokeAPI (Por ejemplo: Pidgey) e imprimir por consola su nombre y peso.
https://pokeapi.co/api/v2/pokemon/{pokemon}

b. A partir de la consulta anterior, imprimir por consola el nombre y peso de su próxima evolución (Es decir, Pidgeotto).

c. ¿Cómo haríamos si queremos también imprimir los datos de su próxima evolución? (En nuestro ejemplo, Pidgeot). ¿Qué comportamiento se repite?

4. **Promesas**

a. Modificar el ejercicio anterior para manejar el código asíncrono con promesas.
b. Instalar la librería Axios y utilizarlo para peticiones HTTP del ejercicio anterior.

Ejercicio integrador Clase 2

Dado el siguiente array:

```
var movies = [
  {
    name: "Thor Ragnarok",
  },
  {
    name: "Back to the Future",
  },
  {
    name: "Robocop",
  },
];
```

Generar un componente (ItemsSection.js) que mediante el método map, imprima lo siguiente:

Dado este otro array:

```
var movies = [
  {
    name: "Thor Ragnarok",
    viewed: true,
  },
  {
    name: "Back to the Future",
    viewed: true,
  },
  {
    name: "Robocop",
    viewed: false,
  },
];
```

Generar un componente (ViewedItems.js) que imprima un listado de componentes “ListItem“, pero mostrando solo los que tengan la propiedad viewed en true mediante el método filter.