

# Taller de Conocimientos

1. Ejecute un proyecto de Node js con “Hola Mundo” desde la consola, bajo las 3 formas en las que se puede aplicar y ejecutar un proyecto de Node Js a TypeScript. Agregue pantallazos de los códigos de la terminal, explorer y ejecución.

2. Tenemos el siguiente código en Node JS

## App.js

```
const tiendas = [  
  {  
    id: 1,  
    nombre: "San Juan",  
    categorias: [  
      {  
        nombre: "Electrónica",  
        productos: [  
          { nombre: "TV", precio: 1599.99, stock: 10, estado: true },  
          { nombre: "Radio", precio: 49.99, stock: 25, estado: true },  
          { nombre: "Teléfono", precio: 699.99, stock: 15, estado: false },  
        ],  
      },  
      {  
        nombre: "Ropa",  
        productos: [  
          { nombre: "Camiseta", precio: 19.99, stock: 50, estado: true },  
          { nombre: "Pantalón", precio: 39.99, stock: 30, estado: true },  
          { nombre: "Zapatos", precio: 89.99, stock: 20, estado: false },  
        ],  
      },  
    ],  
  },  
  {  
    id: 2,  
    nombre: "Santa Marta",  
    categorias: [  
      {  
        nombre: "Electrónica",  
        productos: [  
          { nombre: "TV", precio: 1699.99, stock: 8, estado: true },  
          { nombre: "Radio", precio: 59.99, stock: 20, estado: true },  
          { nombre: "Teléfono", precio: 799.99, stock: 12, estado: true },  
        ],  
      },  
      {  
        nombre: "Ropa",  
        productos: [  
          { nombre: "Camiseta", precio: 14.99, stock: 60, estado: true },  
        ],  
      },  
    ],  
  },  
]
```

```

        { nombre: "Pantalón", precio: 49.99, stock: 25, estado: false },
        { nombre: "Zapatos", precio: 99.99, stock: 15, estado: true },
    ],
},
],
},
];
// Recorrer y mostrar la información
tiendas.forEach(function (tienda) {
    console.log("Tienda: ".concat(tienda.nombre, " (ID: ").concat(tienda.id, ")"));
    tienda.categorias.forEach(function (categoria) {
        console.log("  Categor\u00EDA: ".concat(categoria.nombre));
        categoria.productos.forEach(function (producto) {
            console.log("    Producto: ".concat(producto.nombre));
            console.log("    Precio: ".concat(producto.precio.toFixed(2), " Pesos"));
            console.log("    Stock: ".concat(producto.stock));
            console.log("    Estado: ".concat(producto.estado ? "Activo" : "Inactivo"));
        });
    });
});
});

```

**NOTA:** La secuencia \u00ED es un código Unicode que representa la letra í (i latina con tilde). Esto asegura que el texto sea correcto incluso en sistemas que no manejan directamente caracteres acentuados.

## Desarrollar:

1. Ejecuta el código actual en Node Js y coloca a continuación pantallas del resultado.
2. Se requiere que el código en Node Js, sea escrito de acuerdo al tipado de TypeScript, recuerda que TypeScript se debe definir las variables o constantes a utilizar.
3. Para Mostrar la lista de la información de las tiendas introdúcelo, agrégalo a una function en typescript.
4. Crea una funtion llamada buscarProducto, donde el parámetro de entrada sea el nombre del producto y muestre por pantalla, en que tienda se encuentra, cual es el precio y el stock del mismo, siempre y cuando estén activos.

## Entregables:

1. Código desarrollado
2. Foto de encarpetao o estructura de software(folder-explorer)
3. Video de ejecución
4. Trabajo a realizar máximo 3 personas.
5. Subir todo a un repositorio de GITHUB, y subir el link aquí en Q10.

## Material de Apoyo adicional, que puedes utilizar

**1. filter:** filter es un método de los arrays en JavaScript que se usa para **filtrar elementos** según una condición. Devuelve un nuevo array que contiene solo los elementos que cumplen con esa condición.

**Sintaxis:** array.filter(callback(elemento, índice, array))

- **callback:** Una función que se ejecuta en cada elemento del array.
- **elemento:** El elemento actual que se evalúa.
- **índice** (*opcional*): La posición del elemento en el array.
- **array** (*opcional*): El array original.

**Ejemplo:**

```
const numeros = [1, 2, 3, 4, 5];  
  
const pares = numeros.filter(num => num % 2 === 0);  
  
console.log(pares); // [2, 4]
```

---

**2. toLowerCase()** es un método de las cadenas de texto (strings) en JavaScript que convierte todos los caracteres de la cadena a **minúsculas**.

**Sintaxis:** cadena.toLowerCase()

**Ejemplo:**

```
const texto = "Hola Mundo";  
  
console.log(texto.toLowerCase()); // "hola mundo"
```

**Uso común:** Se utiliza a menudo para hacer comparaciones de cadenas de texto sin importar mayúsculas o minúsculas.

**Ejemplo con comparación:**

```
const palabra1 = "Hola";  
  
const palabra2 = "hola";  
  
if (palabra1.toLowerCase() === palabra2.toLowerCase()) {  
    console.log("Son iguales.");  
} else {  
    console.log("Son diferentes.");  
}
```

```
}
```

```
// Salida: "Son iguales."
```

---

### **Combinación de filter y toLowerCase()**

Se pueden usar juntos para filtrar elementos en un array de cadenas ignorando mayúsculas o minúsculas. Por ejemplo:

#### **Ejemplo práctico:**

```
const productos = ["TV", "Radio", "Teléfono", "teléfono"];
const resultado = productos.filter(producto => producto.toLowerCase() === "teléfono");
console.log(resultado); // ["Teléfono", "teléfono"]
```

#### **Explicación:**

1. **producto.toLowerCase()** convierte cada nombre de producto a minúsculas.
2. **producto.toLowerCase() === "teléfono"** compara el nombre en minúsculas con "teléfono".
3. **filter** devuelve un nuevo array con los productos que coinciden.