

Desestructurando Argumentos

```
interface Producto {
  descripcion: string;
  precio: number;
}

const telefono: Producto = {
  descripcion: 'iPhone 15 Pro',
  precio: 1000
};

const tablet: Producto = {
  descripcion: 'iPad Air 3',
  precio: 500
};

interface CalculaImpuestoOpciones {
  impuesto: number;
  productos: Producto[];
}

/*
function calculaImpuesto (opciones: CalculaImpuestoOpciones):
number[] {
  let total = 0;
  opciones.productos.forEach(producto => {
    total += producto.precio;
  });
  return [total, total * opciones.impuesto];
}
*/
// He cambiado para que reciba parámetros desestructurando
function calculaImpuesto({ impuesto, productos }:
CalculaImpuestoOpciones): number[] {
  let total = 0;
  productos.forEach(({ precio }) => {
    total += precio;
  });
  return [total, total * impuesto];
}
```

```

const carritoCompra = [telefono, tablet];
const impuesto = 0.21;

/*
const resultado = calculaImpuesto({
  productos: carritoCompra,
  impuesto: impuesto
  // también podemos poner por la línea anterior, ya que se
  llaman iguales:
  // impuesto
})
*/
// Desestructuración al acceder a un dato en una estructura de
control
const [total, totalImpuesto] = calculaImpuesto({productos:
carritoCompra, impuesto});

console.log('Total', total);
console.log('Impuesto', totalImpuesto);

// He cambiado para que se acceda al dato desestructurando
for (const { descripcion, precio } of carritoCompra) {
  console.log(`Producto: ${descripcion}, Precio: ${precio}`);
}

export {};

```

Resultado

Total 1500	06-desest-funciones.ts:53
Impuesto 315	06-desest-funciones.ts:54
Producto: iPhone 15 Pro, Precio: 1000	06-desest-funciones.ts:58
Producto: iPad Air 3, Precio: 500	06-desest-funciones.ts:58