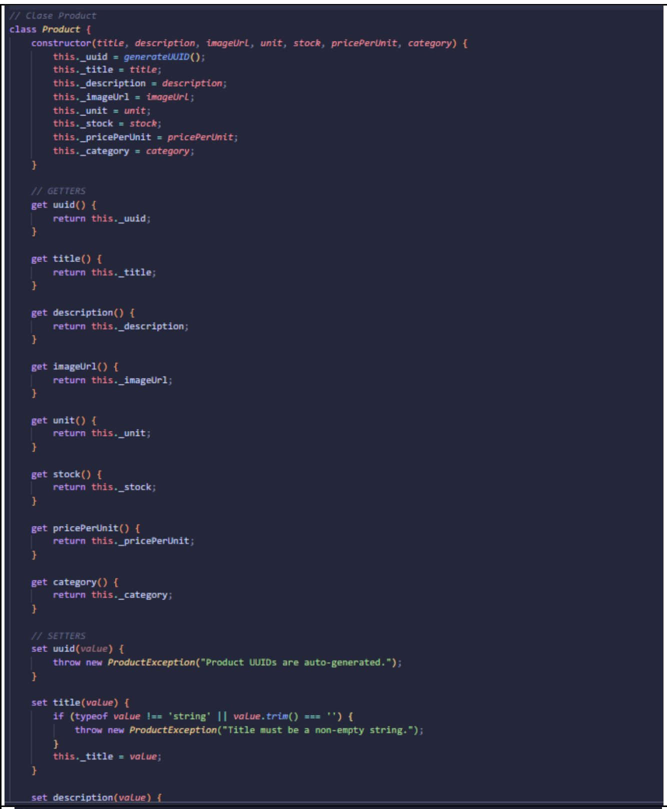
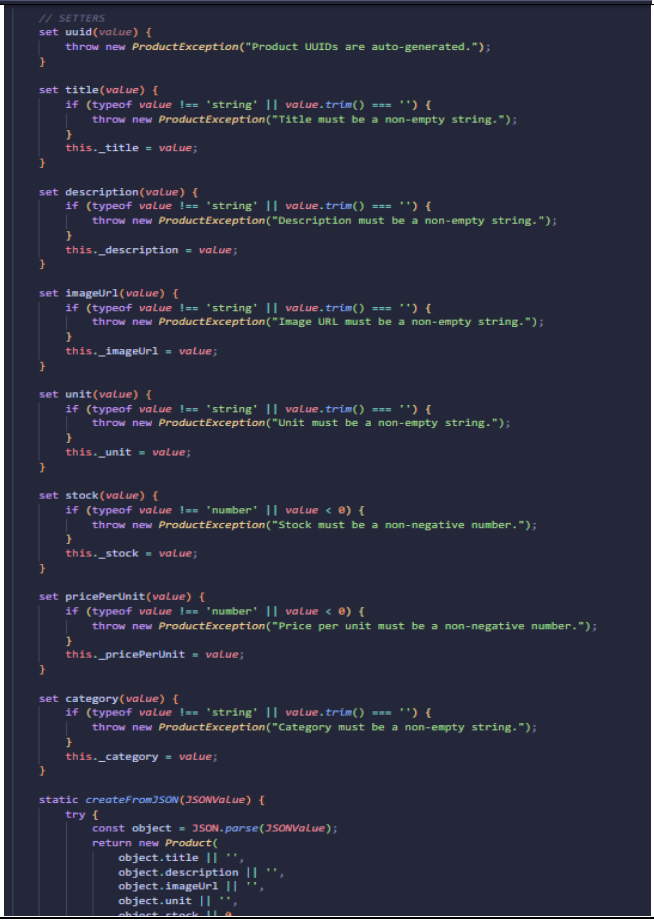
CLASE PRODUCT

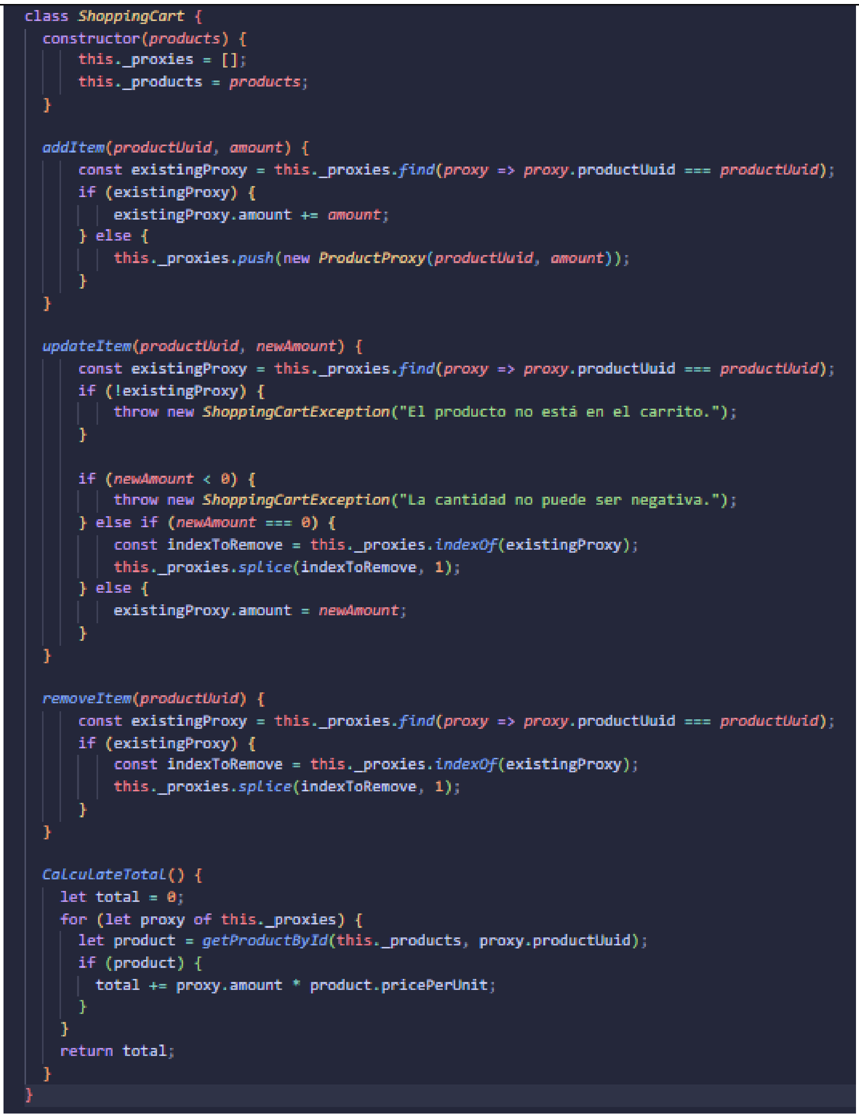


```
// Clase Product
class Product {
    constructor(title, description, imageUrl, unit, stock, pricePerUnit, category) {
        this._uuid = generateUUID();
        this._title = title;
        this._description = description;
        this._imageUrl = imageUrl;
        this._unit = unit;
        this._stock = stock;
        this._pricePerUnit = pricePerUnit;
        this._category = category;
    }

    // GETTERS
    get uuid() {
        return this._uuid;
    }

    get title() {
        return this._title;
    }

    get description() {
        return this._description;
    }

    get imageUrl() {
        return this._imageUrl;
    }

    get unit() {
        return this._unit;
    }

    get stock() {
        return this._stock;
    }

    get pricePerUnit() {
        return this._pricePerUnit;
    }

    get category() {
        return this._category;
    }

    // SETTERS
    set uuid(value) {
        throw new ProductException("Product UUIDs are auto-generated.");
    }

    set title(value) {
        if (typeof value !== 'string' || value.trim() === '') {
            throw new ProductException("Title must be a non-empty string.");
        }
        this._title = value;
    }

    set description(value) {
```



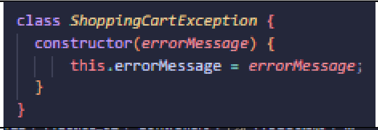
```
    // SETTERS
    set uuid(value) {
        throw new ProductException("Product UUIDs are auto-generated.");
    }

    set title(value) {
        if (typeof value !== 'string' || value.trim() === '') {
            throw new ProductException("Title must be a non-empty string.");
        }
        this._title = value;
    }

    set description(value) {
        if (typeof value !== 'string' || value.trim() === '') {
            throw new ProductException("Description must be a non-empty string.");
        }
        this._description = value;
    }

    set imageUrl(value) {
        if (typeof value !== 'string' || value.trim() === '') {
            throw new ProductException("Image URL must be a non-empty string.");
        }
        this._imageUrl = value;
    }

    set unit(value) {
        if (typeof value !== 'string' || value.trim() === '') {
            throw new ProductException("Unit must be a non-empty string.");
        }
        this._unit = value;
    }

    set stock(value) {
        if (typeof value !== 'number' || value < 0) {
            throw new ProductException("Stock must be a non-negative number.");
        }
        this._stock = value;
    }

    set pricePerUnit(value) {
        if (typeof value !== 'number' || value < 0) {
            throw new ProductException("Price per unit must be a non-negative number.");
        }
        this._pricePerUnit = value;
    }

    set category(value) {
        if (typeof value !== 'string' || value.trim() === '') {
            throw new ProductException("Category must be a non-empty string.");
        }
        this._category = value;
    }

    static createFromJSON(JSONValue) {
        try {
            const object = JSON.parse(JSONValue);
            return new Product(
                object.title || '',
                object.description || '',
                object.imageUrl || '',
                object.unit || '',
                object.stock || 0
```

CLASE SHOPPINGCART

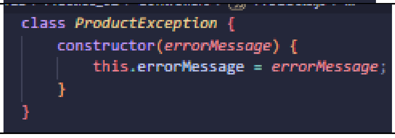


```
class ShoppingCart {
  constructor(products) {
      this._proxies = [];
      this._products = products;
  }

  addItem(productUuid, amount) {
      const existingProxy = this._proxies.find(proxy => proxy.productUuid === productUuid);
      if (existingProxy) {
          existingProxy.amount += amount;
      } else {
          this._proxies.push(new ProductProxy(productUuid, amount));
      }
  }

  updateItem(productUuid, newAmount) {
      const existingProxy = this._proxies.find(proxy => proxy.productUuid === productUuid);
      if (!existingProxy) {
          throw new ShoppingCartException("El producto no está en el carrito.");
      }

      if (newAmount < 0) {
          throw new ShoppingCartException("La cantidad no puede ser negativa.");
      } else if (newAmount === 0) {
          const indexToRemove = this._proxies.indexOf(existingProxy);
          this._proxies.splice(indexToRemove, 1);
      } else {
          existingProxy.amount = newAmount;
      }
  }

  removeItem(productUuid) {
      const existingProxy = this._proxies.find(proxy => proxy.productUuid === productUuid);
      if (existingProxy) {
          const indexToRemove = this._proxies.indexOf(existingProxy);
          this._proxies.splice(indexToRemove, 1);
      }
  }

  CalculateTotal() {
    let total = 0;
    for (let proxy of this._proxies) {
      let product = getProductById(this._products, proxy.productUuid);
      if (product) {
        total += proxy.amount * product.pricePerUnit;
      }
    }
    return total;
  }
}
```

EXCEPCIONES Y VALIDACIONES

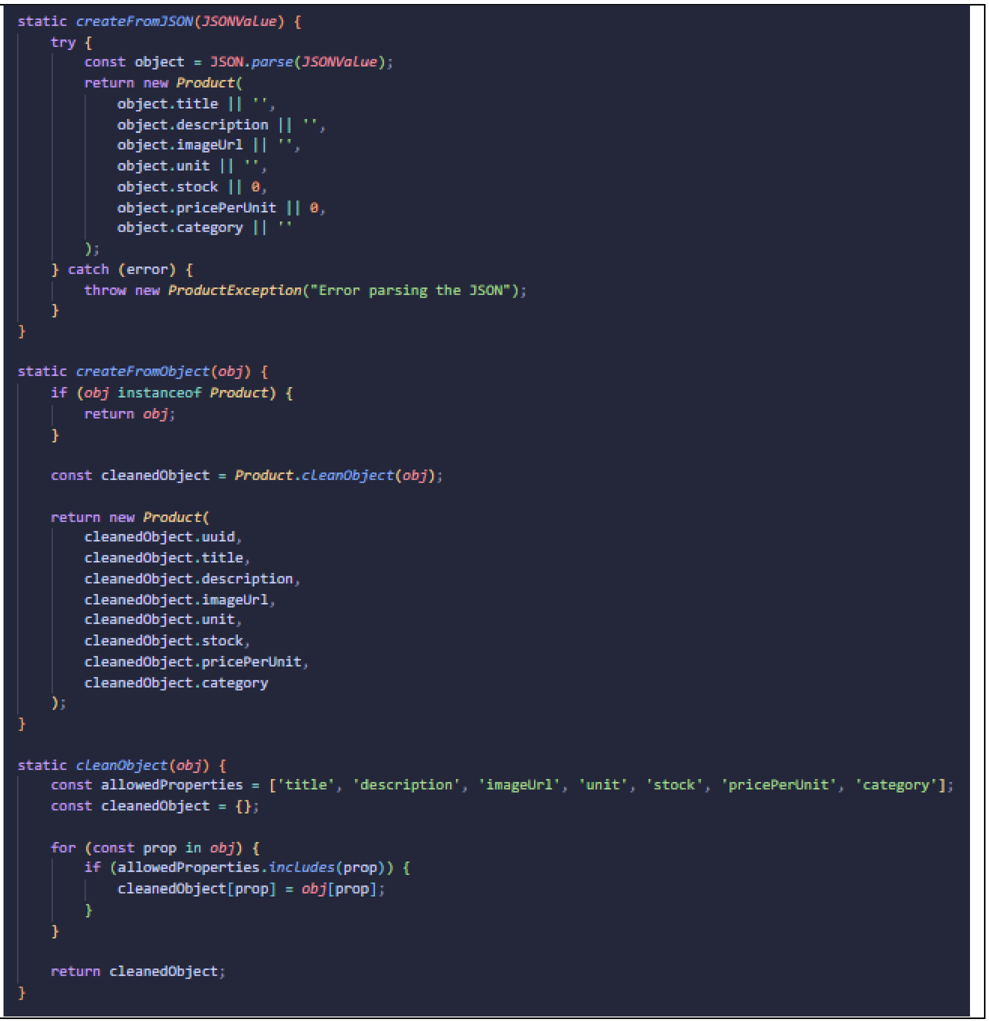


```
class ShoppingCartException {
  constructor(errorMessage) {
      this.errorMessage = errorMessage;
  }
}
```



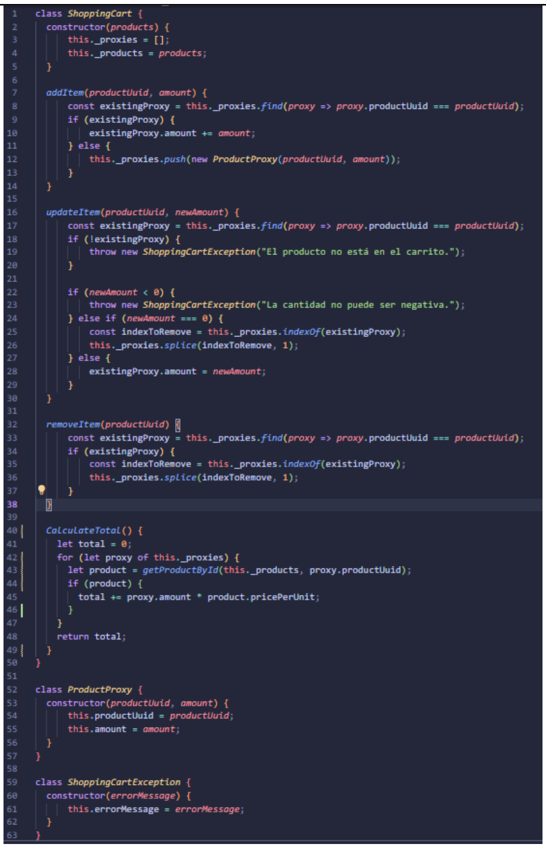
```
class ProductException {
    constructor(errorMessage) {
        this.errorMessage = errorMessage;
    }
}
```

FUNCIONES ESTÁTICAS DE PRODUCT



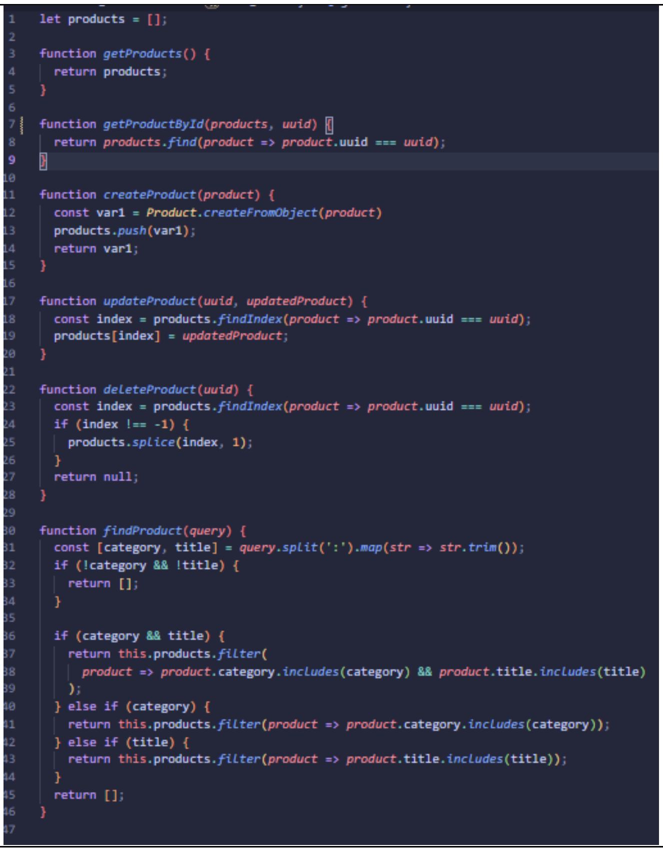
```
static createFromJSON(JSONValue) {
    try {
        const object = JSON.parse(JSONValue);
        return new Product(
            object.title || '',
            object.description || '',
            object.imageUrl || '',
            object.unit || '',
            object.stock || 0,
            object.pricePerUnit || 0,
            object.category || ''
        );
    } catch (error) {
        throw new ProductException("Error parsing the JSON");
    }
}

static createFromObject(obj) {
    if (obj instanceof Product) {
        return obj;
    }

    const cleanedObject = Product.cleanObject(obj);

    return new Product(
        cleanedObject.uuid,
        cleanedObject.title,
        cleanedObject.description,
        cleanedObject.imageUrl,
        cleanedObject.unit,
        cleanedObject.stock,
        cleanedObject.pricePerUnit,
        cleanedObject.category
    );
}

static cleanObject(obj) {
    const allowedProperties = ['title', 'description', 'imageUrl', 'unit', 'stock', 'pricePerUnit', 'category'];
    const cleanedObject = {};

    for (const prop in obj) {
        if (allowedProperties.includes(prop)) {
            cleanedObject[prop] = obj[prop];
        }
    }

    return cleanedObject;
}
```

CRUD DEL CARRITO DE COMPRAS



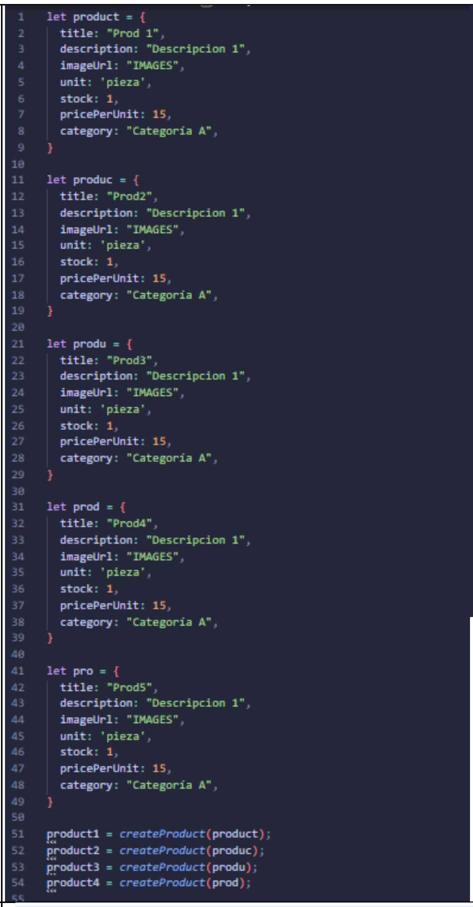
```javascript
class ShoppingCart {
  constructor(products) {
    this._proxies = [];
    this._products = products;
  }

  addItem(productUuid, amount) {
    const existingProxy = this._proxies.find(proxy => proxy.productUuid === productUuid);
    if (existingProxy) {
      existingProxy.amount += amount;
    } else {
      this._proxies.push(new ProductProxy(productUuid, amount));
    }
  }

  updateItem(productUuid, newAmount) {
    const existingProxy = this._proxies.find(proxy => proxy.productUuid === productUuid);
    if (!existingProxy) {
      throw new ShoppingCartException("El producto no está en el carrito.");
    }

    if (newAmount < 0) {
      throw new ShoppingCartException("La cantidad no puede ser negativa.");
    } else if (newAmount === 0) {
      const indexToRemove = this._proxies.indexOf(existingProxy);
      this._proxies.splice(indexToRemove, 1);
    } else {
      existingProxy.amount = newAmount;
    }
  }

  removeItem(productUuid) {
    const existingProxy = this._proxies.find(proxy => proxy.productUuid === productUuid);
    if (existingProxy) {
      const indexToRemove = this._proxies.indexOf(existingProxy);
      this._proxies.splice(indexToRemove, 1);
    }
  }

  CalculateTotal() {
   let total = 0;
   for (let proxy of this._proxies) {
    let product = getProductById(this._products, proxy.productUuid);
    if (product) {
     total += proxy.amount * product.pricePerUnit;
    }
   }
   return total;
  }
}

class ProductProxy {
  constructor(productUuid, amount) {
    this.productUuid = productUuid;
    this.amount = amount;
  }
}

class ShoppingCartException {
  constructor(errorMessage) {
    this.errorMessage = errorMessage;
  }
}
```

CRUD DEL DATA HANDLER

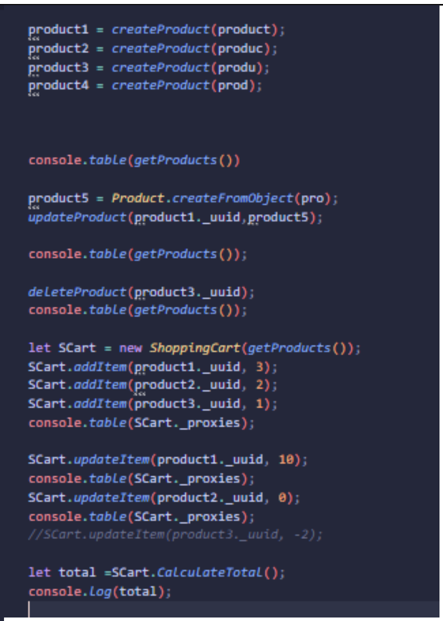


```javascript
let products = [];

function getProducts() {
  return products;
}

function getProductById(products, uuid) {
  return products.find(product => product.uuid === uuid);
}

function createProduct(product) {
  const var1 = Product.createFromObject(product)
  products.push(var1);
  return var1;
}

function updateProduct(uuid, updatedProduct) {
  const index = products.findIndex(product => product.uuid === uuid);
  products[index] = updatedProduct;
}

function deleteProduct(uuid) {
  const index = products.findIndex(product => product.uuid === uuid);
  if (index !== -1) {
    products.splice(index, 1);
  }
  return null;
}

function findProduct(query) {
  const [category, title] = query.split(':').map(str => str.trim());
  if (!category && !title) {
    return [];
  }

  if (category && title) {
    return this.products.filter(
      product => product.category.includes(category) && product.title.includes(title)
    );
  } else if (category) {
    return this.products.filter(product => product.category.includes(category));
  } else if (title) {
    return this.products.filter(product => product.title.includes(title));
  }
  return [];
}
```

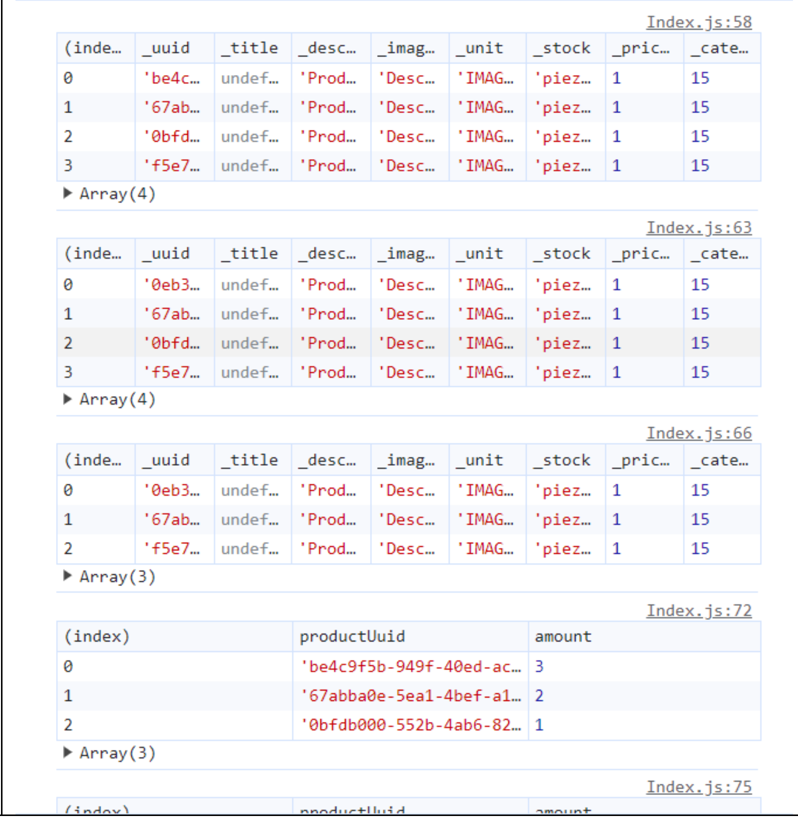
PRUEBAS Y EJEMPLOS



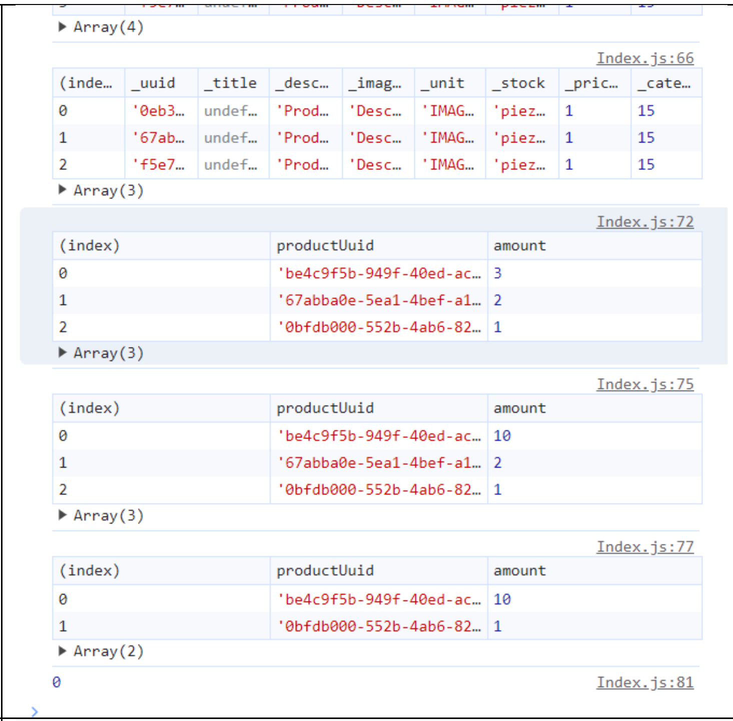
```
let product = {
  title: "Prod 1",
  description: "Descripcion 1",
  imageUrl: "IMAGES",
  unit: 'pieza',
  stock: 1,
  pricePerUnit: 15,
  category: "Categoria A",
}

let produc = {
  title: "Prod2",
  description: "Descripcion 1",
  imageUrl: "IMAGES",
  unit: 'pieza',
  stock: 1,
  pricePerUnit: 15,
  category: "Categoria A",
}

let produ = {
  title: "Prod3",
  description: "Descripcion 1",
  imageUrl: "IMAGES",
  unit: 'pieza',
  stock: 1,
  pricePerUnit: 15,
  category: "Categoria A",
}

let prod = {
  title: "Prod4",
  description: "Descripcion 1",
  imageUrl: "IMAGES",
  unit: 'pieza',
  stock: 1,
  pricePerUnit: 15,
  category: "Categoria A",
}

let pro = {
  title: "Prod5",
  description: "Descripcion 1",
  imageUrl: "IMAGES",
  unit: 'pieza',
  stock: 1,
  pricePerUnit: 15,
  category: "Categoria A",
}

product1 = createProduct(product);
product2 = createProduct(produc);
product3 = createProduct(produ);
product4 = createProduct(prod);
```



```
product1 = createProduct(product);
product2 = createProduct(produc);
product3 = createProduct(produ);
product4 = createProduct(prod);



console.table(getProducts())

product5 = Product.createFromObject(pro);
updateProduct(product1._uuid,product5);

console.table(getProducts());

deleteProduct(product3._uuid);
console.table(getProducts());

let SCart = new ShoppingCart(getProducts());
SCart.addItem(product1._uuid, 3);
SCart.addItem(product2._uuid, 2);
SCart.addItem(product3._uuid, 1);
console.table(SCart._proxies);

SCart.updateItem(product1._uuid, 10);
console.table(SCart._proxies);
SCart.updateItem(product2._uuid, 0);
console.table(SCart._proxies);
//SCart.updateItem(product3._uuid, -2);

let total =SCart.CalculateTotal();
console.log(total);
```



```
                                                                     Index.js:58
(inde…  _uuid   _title  _desc…  _imag…  _unit   _stock  _pric…  _cate…
0       'be4c…  undef…  'Prod…  'Desc…  'IMAG…  'piez…  1       15
1       '67ab…  undef…  'Prod…  'Desc…  'IMAG…  'piez…  1       15
2       '0bfd…  undef…  'Prod…  'Desc…  'IMAG…  'piez…  1       15
3       'f5e7…  undef…  'Prod…  'Desc…  'IMAG…  'piez…  1       15
▶ Array(4)
                                                                     Index.js:63
(inde…  _uuid   _title  _desc…  _imag…  _unit   _stock  _pric…  _cate…
0       '0eb3…  undef…  'Prod…  'Desc…  'IMAG…  'piez…  1       15
1       '67ab…  undef…  'Prod…  'Desc…  'IMAG…  'piez…  1       15
2       '0bfd…  undef…  'Prod…  'Desc…  'IMAG…  'piez…  1       15
3       'f5e7…  undef…  'Prod…  'Desc…  'IMAG…  'piez…  1       15
▶ Array(4)
                                                                     Index.js:66
(inde…  _uuid   _title  _desc…  _imag…  _unit   _stock  _pric…  _cate…
0       '0eb3…  undef…  'Prod…  'Desc…  'IMAG…  'piez…  1       15
1       '67ab…  undef…  'Prod…  'Desc…  'IMAG…  'piez…  1       15
2       'f5e7…  undef…  'Prod…  'Desc…  'IMAG…  'piez…  1       15
▶ Array(3)
                                                                     Index.js:72
(index)                 productUuid                 amount
0                       'be4c9f5b-949f-40ed-ac…     3
1                       '67abba0e-5ea1-4bef-a1…     2
2                       '0bfdb000-552b-4ab6-82…     1
▶ Array(3)
                                                                     Index.js:75
(index)                 productUuid                 amount
```



```
▶ Array(4)
                                                              Index.js:66
(inde… _uuid   _title  _desc…  _imag…  _unit   _stock  _pric…  _cate…
0      '0eb3…  undef…  'Prod…  'Desc…  'IMAG…  'piez…  1       15
1      '67ab…  undef…  'Prod…  'Desc…  'IMAG…  'piez…  1       15
2      'f5e7…  undef…  'Prod…  'Desc…  'IMAG…  'piez…  1       15
▶ Array(3)
                                                              Index.js:72
(index)                      productUuid                   amount
0                            'be4c9f5b-949f-40ed-ac…       3
1                            '67abba0e-5ea1-4bef-a1…       2
2                            '0bfdb000-552b-4ab6-82…       1
▶ Array(3)
                                                              Index.js:75
(index)                      productUuid                   amount
0                            'be4c9f5b-949f-40ed-ac…       10
1                            '67abba0e-5ea1-4bef-a1…       2
2                            '0bfdb000-552b-4ab6-82…       1
▶ Array(3)
                                                              Index.js:77
(index)                      productUuid                   amount
0                            'be4c9f5b-949f-40ed-ac…       10
1                            '0bfdb000-552b-4ab6-82…       1
▶ Array(2)
0                                                             Index.js:81
>
```

CONCLUSIÓN	En este proyecto, creamos un sistema para manejar productos y carritos de compra usando JavaScript. Organice nuestro código en clases y usamos excepciones para manejar errores. Aprendí sobre la importancia de probar nuestro código, aunque tuve algunos problemas con el cálculo del total en el carrito de compras. En el futuro, podría mejorar mi código agregando más pruebas y comentarios, y probándola en la parte del frontend de la página. En general, este proyecto fue un buen ejercicio para aprender sobre programación web backend y resolución de problemas que se puedan presentar en este tipo de páginas.