

Primera Evaluación Programación Orientada a Objetos II.

Proyecto carrito de compras.

Crear un carrito de compras funcional en JavaScript con Node.

Clases que deben estar presentes en el proyecto son:

Producto: id, nombre, costo, cantidad, métodos get() y set() necesarios.

ProductoPerecedero: fechaCaducidad,

ProductoPerecederoRefrigeracion: tipoRefrigeracion

ProductoLineaBlanca: dimensiones,

Stock: listaProductos[], agregarProducto(), actualizarStock(), mostrarStock(),

buscarProducto(), modificarProducto()

Carrito: listaProductos[], agregarProducto(), mostrarCarrito(),

buscarProducto(), quitarProducto(), pagar()

Para su optimización los objetos de tipo producto presenta una relación de herencia donde:

Producto es el más genérico,

ProductoPerecedero es de tipo **Producto**

ProductoPerecederoRefrigeracion es de tipo **ProductoPerecedero**.

ProductoLineaBlanca es de tipo **Producto**.

OPCIONAL... una clase lista que será de la que hereden Stock y Carrito

El **Stock** estará conformado por un arreglo de productos

El **Carrito** estará conformado por un arreglo de productos que deben existir en el Stock .

El carrito de compras operará sobre el Stock (Arreglo de objetos),

Los cambios sobre el stock no se verán realizados hasta que la compra se haya confirmado.

Todas las operaciones deberán estar validadas como por ejemplo:

- no agregar más productos de los existentes en el carrito

Identificar variables y métodos privados(), utilizar polimorfismo.

Los siguientes puntos se considerarán para su evaluación.

	Excelente	Bien	Regular	Insuficiente
Diseño UML de las clases y sus relaciones a nivel de detalle de desarrollo				
La programación deberá utilizar el paradigma de la POO.				
Manejo de Arreglos				
Manejo de Callback				
Manejo de Arrow function				
Funcionalidades del carrito: Agregar producto, eliminar producto, pagar, editar producto, mostrar carrito				
Funcionalidades del stock: Crear producto, modificar producto, actualizar stock.				