**EVALUACIÓN 2**

**ASIGNATURA: CSY4111 – Sección 2**

**Calidad de Software**

**DOCENTE: Marcela Ulloa Zamora**

|  |  |  |
| --- | --- | --- |
| **PUNTAJE MÁXIMO** | **Teórico 44 puntos (35%)** | **NOTA** |
| **PUNTAJE OBTENIDO** |  |
| **FECHA** | **/Octubre/2021** |
| **DURACIÓN** | **15 minutos por grupo** |

**INDICACIONES GENERALES**

1. La prueba será evaluada con una escala de 1 a 7.
2. Corresponde al 35% de la nota final

Diagrama

Descripción generada automáticamente

**Enunciado**

El supermercado “Number One”, está automatizando sus procesos administrativos. Dentro de ellos se encuentra la venta de sus productos: carnes, frutas y verduras. Por tanto, se solicita su participación dentro del equipo de desarrollo.

A continuación, se anexa la descripción de cada clase, para ser implementada en Java.

Además, debe considerar las siguientes reglas de negocio:

* Valores de montos de dinero debe ser mayor o igual a $0.
* Cantidad de productos, debe ser mayor a o igual a 0.
* Categoría debe ser ‘A’, ‘B’, ‘C’
* El porcentaje de descuento debe ser para categoría A ->10%, B ->20%, C -> 30%. ---
* SE INGRESA MANUALMENTE
* Tipo de Fruta o Verdura: ‘FR’-> Frutos, ‘BL’-> Bulbos, ‘HT’-> Hojas y Tallos verdes, ‘LG-> Legumbres, ‘RC-> Raíces. – TIPO DE FRUTAS DE INGRESA MANUALMENTE Y NO PARAMETRICO

**Métodos solicitados:**

1. Calcular Monto Neto: Método encargado de calcular el monto neto que se pagará por un producto considerando que el monto se calcular multiplicando la cantidad del producto por el precio unitario.

Nota: En la carne se debe considerar el descuento indicado en el atributo de la clase.

1. Calcular Monto IVA: Método encargado de calcular el monto del IVA considerando un 19% por sobre el monto neto del producto. – IVA EN LAS FRTUAS ESTA PUESTO MANUALMENTE
2. Calcular Total Pago: Método encargado de calcular el total a pagar el que se calcular sumando el monto neto más monto IVA. – NO IMPRIME NADA, NO HAY FUNCION QUE UTILIZE EL RESULTADO DE SU CALCULO
3. Agregar Producto: Método encargado de agregar un producto al carrito de compra validando que ningún producto se repita a partir del nombre.
4. Quitar Producto: Método encargado de quitar un producto del carrito de compra por medio del nombre.
5. Listar Productos: Método encargado de listar los productos del carrito de compra.
6. Precio Total Compra: Método encargado de calcular el monto total a pagar de los productos

**Se requieren los siguientes métodos en las clases:**

1. constructores con parámetros para todas las clases. -- FRUTAS NO TIENE CONSTRUCTORES
2. accesadores para cada atributo de cada clase. – FRUTAS NO TIENE ACCESADORES
3. mutadores para cada atributo de cada clase considerando las reglas de negocio. – MUTADORES DE CATEGORIA DE FRUTAS Y DESCUENTO NO OBEDECEN REGLAS DE NEGOCIO
4. imprimir: para todas las clases. Retornará todos los atributos. – Imprimir de frutas solo retorna el tipo que también esta erroneo

**Desde la clase Principal (main()):**

* Construir un objeto lomo y manzana, agregar cantidad y precios unitarios. – SE ESTA AGREGANDO EL DESCUENTO MANUALMENTE AL IGUAL QUE UNA FORMA ERRONEA DEL TIPO
* Agregar productos al carrito de compra.
* Listar productos del carrito de compra.
* Imprimir precio total de compra.—NO EXISTE FORMA DE IMPRIMIR TODO EL CARRITO
* Eliminar un producto del carrito de compra. --

**Determinar las pruebas para el código**

Ejecutar pruebas formales a un software sencillo de manera que se pueda detectar los casos de prueba defectuosos y reportar los defectos encontrados.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **DESCRIPCIÓN** | **DATOS** | **RESULTADO ESPERADO** | **RESULTADO OBTENIDO** | **Resultado: Exitoso - Defectuoso** |
| **CP01** | Crear objeto fruta, con el tipo ingresado erróneamente | Cantidad nombre precio tipo | El objeto no se crea y muestra un error | El objeto se creo | Defectuoso |
| **CP02** | Imprimir fruta | Cantidad nombre precio tipo | Se imprime la totalidad de los atributos | Se imprime solamente el tipo | Defectuoso |
| **CP03** | Crear objeto carne, con descuento ingresado de manera manual | Cantidad nombre precio descuento | El objeto no se crea y muestra un error | El objeto se creo | Defectuoso |
| **CP04** | Calcular monto neto en objeto carne | Cantidad precio y descuento por categoría | Los cálculos se realizan correctamente, aplicando el descuento respectivo en base a la categoría | El precio pagado es 20 veces mayor al que debería ser | Defectuoso |
| **CP05** | Agregar productos al carrito, repitiendo un producto | Objetos carne y fruta | No se registra el producto repetido | Los objetos se agregaron sin repetirse | Exitoso |
| **CP06** | Listar productos del carrito | Objetos dentro de carrito | Se listan todos los productos con sus atributos | No se muestran los atributos de cada producto correctamente | Defectuoso |
| **CP07** | Eliminar productos de carrito | Objeto producto dentro de carrito | Se elimina el objeto indicado en base a su nombre | El objeto se eliminó correctamente | Exitoso |
| **CP08** | Calcular precio total del carrito | Todos los objetos dentro de carrito | El precio se calcula y se muestra en pantalla | El precio solo se calcula, pero no se muestra en pantalla | Defectuoso |
| **CP09** | Calcular monto neto en objeto fruta | Cantidad y precio | El precio se calcula correctamente | El cálculo es correcto | Exitoso |
| **CP10** | Ingresar cantidad negativa de productos | Cantidad | El sistema no permite la creación del objeto | El objeto se creó correctamente | Defectuoso |
| **CP11** | Ingresar una categoría no correspondiente | Categoría | El sistema no permite la creación del objeto | El objeto se creó correctamente | Defectuoso |
| **CP12** | Ingresar objeto con valor negativo | Precio | El sistema no permite la creación del objeto | El objeto se creó correctamente | Defectuoso |