

## **Entrega #2 trabajo práctico**

Juan Sebastian Garcia Perez, Nicolas Tobón Rodriguez y Andres Felipe Gonzalez Puerta

Facultad de ingenierías, Universidad de Medellín

Ing. Sistemas, Algoritmos y Programación Orientada a Objetos grupo 61

Jesús Andrés Hincapié Londoño

Medellín, Colombia

18/05/2023

## **Sistema de facturación e inventario para Kalle Rosa Jeans**

Kalle Rosa Jeans es una empresa dedicada a la venta de ropa al por mayor y al detal con más de 4 años de actividad, sin embargo, últimamente está presentando pérdidas de dinero y mercancía debido al aumento de clientes, esto debido a la desorganización y el poco control sobre la facturación y la gestión del inventario, lo que ha causado pérdidas de facturas, pedidos mal entregados, facturas con errores en el valor, exceso o escasez de inventario, entre otros problemas. Para dar solución al problema, requieren un sistema que les permita manejar organizadamente la facturación y el inventario de la empresa.

La estructura de la empresa está conformada actualmente por 2 vendedores, un encargado del inventario y el administrador, pero se tiene pensado ampliar el personal para agilizar los procesos.

Cuando un cliente está interesado en hacer un pedido, puede ir presencialmente al local de la empresa y ser atendido por uno de los vendedores o puede comunicarse por chat con la empresa, caso en donde es atendido directamente por el administrador. Posteriormente, el encargado de atender al cliente toma el pedido, hace manualmente la lista de productos pedidos y calcula el precio total que deberá pagar el cliente. Luego, entrega una factura hecha a mano al cliente donde le indica la cantidad de cada producto con su valor y el valor total del pedido, además conserva una copia de esa factura para luego hacer la contabilidad.

Los productos que se venden están divididos y contenidos en distintas categorías y tienen planeado aumentar la cantidad de categorías en un futuro. Dentro de cada una de las categorías hay distintos productos identificados con un código único. Cada categoría tiene para cada uno de sus productos un valor unitario el cual varía dependiendo de la cantidad de productos de esa

categoría que el cliente lleve: entre más productos lleve el cliente de una misma categoría en la misma operación, el valor unitario será menor.

El sistema de precios por unidades compradas funciona así:

- De 1 a 5 unidades, se cobra el precio base de los productos.
- De 6 a 11 unidades se descuenta un 10% del valor unitario.
- De 12 a 119 unidades se descuenta un 15% del valor unitario.
- De 120 unidades en adelante se descuenta un 20% del valor unitario.

Para dar solución al problema, se hace necesario implementar un software que permita a los usuarios:

- Iniciar sesión para iniciar el sistema.
- Registrar nuevos usuarios con nombre, número de identificación, contraseña y cargo (opcional).
- Consultar la cantidad disponible en inventario de cualquier producto.
- Generar facturas que contengan: nombre de vendedor, nombre del cliente, lista de productos y cantidades y el precio total del pedido (tomando en cuenta porcentajes de descuento de cantidad).
- Consultar lista de facturas generadas.
- Calcular el precio de un pedido.
- Crear nuevas categorías de productos con sus respectivos nombres y precios base.
- Ingresar nuevos productos a las categorías, cada uno con su respectivo código identificativo.
- Modificar valores de categorías (precio) o de productos (cantidad).

### Requisitos funcionales

<b>Nombre</b>	<b>R1 - Registrar usuario</b>
<b>Resumen</b>	El sistema debe permitir registrar a los usuarios
<b>Entradas</b>	1. Número de identificación del usuario 2. Nombre del usuario 3. Contraseña del usuario 4. Cargo del usuario
<b>Resultado</b>	1. El sistema recibe los datos de registro 2. El sistema muestra un mensaje de que el usuario ha sido registrado correctamente. 3. El sistema guarda el usuario registrado en el archivo. 4.El sistema muestra el menú inicial de registro/inicio de sesión

<b>Pasos</b>	<b>Método</b>	<b>Responsable</b>
Registrar usuario y guardarlo	registrar_usuario()	Registro

<b>Nombre</b>	<b>R2 – Iniciar sesión usuario</b>
<b>Resumen</b>	El sistema debe permitir a los usuarios iniciar sesion para iniciar el programa
<b>Entradas</b>	1. Número de identificación de usuario 2. Contraseña de usuario
<b>Resultado</b>	1. El sistema compara los datos ingresados de inicio de sesión con los datos de registro guardados en el archivo 1.1. Si los datos ingresados coinciden con algún registro 1.1.1 El sistema deja ingresar al usuario 1.2. Si los datos ingresados no coinciden con algún registro 1.2.1. El sistema solicita ingresar los datos de nuevo al usuario

<b>Pasos</b>	<b>Método</b>	<b>Responsable</b>
comparar datos de inicio de sesion con datos de registro	iniciar_sesion()	UsuarioInicioSesion

<b>Nombre</b>	<b>R3 - Crear categoría</b>
Resumen	El sistema debe permitir a los usuarios ingresar nuevas categorías de productos
Entradas	- Nombre de la categoría - Precio general de los productos de la categoría
Resultado	1. El sistema añade la categoría a la lista de categorías 1.1.El sistema muestra un mensaje de que la categoría fue creada 2. El sistema guarda la categoría en el archivo. 2. El sistema muestra el menú de usuario

<b>Pasos</b>	<b>Método</b>	<b>Responsable</b>
Añadir categoría y guardarla	crear_categoria(nombre_categoria, precio)	Inventario

<b>Nombre</b>	<b>R4 - Ingresar productos</b>
Resumen	El sistema debe permitir a los usuarios ingresar nuevos productos con sus categorías correspondientes y códigos numéricos únicos
Entradas	1. Categoría del producto 2. Código numérico 3. Cantidad de producto
Resultado	1. El sistema Agrega el producto a la lista de la categoría correspondiente 2. El sistema guarda el producto dentro del archivo. 3. El sistema muestra el menú de usuario

<b>Pasos</b>	<b>Método</b>	<b>Responsable</b>
Ingresar producto y guardarlo	crear_categoria(nombre, precio)	Inventario

<b>Nombre</b>	<b>R5 - Modificar valores de categoría o producto</b>
<b>Resumen</b>	El sistema debe permitir a los usuarios modificar los valores de los productos o de las categorías
<b>Entradas</b>	- Tipo de modificación (categoría o producto) - Nuevo valor
<b>Resultado</b>	1. El sistema solicita al usuario seleccionar el tipo de modificación 2. Si el usuario elige modificar categoría 2.1. le permite al usuario seleccionar la categoría deseada y cambiar el precio 3 Si el usuario elige modificar producto 3.1 le permite al usuario modificar la cantidad del producto. 4. El sistema guarda los cambios en el archivo. 5. Muestra el menú de usuario

<b>Pasos</b>	<b>Método</b>	<b>Responsable</b>
Modificar cantidad de categoría y guardarla	modificar_categoria(categoria , nuevo_precio)	Inventario
Modificar precio producto y guardarlo	modificar_producto(categoria , id_producto, nueva_cantidad)	Inventario

<b>Nombre</b>	<b>R6 - Mostrar inventario</b>
<b>Resumen</b>	El sistema debe mostrar el inventario (categorías y productos)
<b>Entradas</b>	Ninguna
<b>Resultado</b>	1. El sistema accede al archivo de inventario. 2. El sistema muestra los datos guardados en inventario. 3. El sistema muestra el menú de usuario

<b>Pasos</b>	<b>Método</b>	<b>Responsable</b>
Obtener los datos del inventario guardados en el archivo y mostrarlos	mostrar_inventario()	Inventario

<b>Nombre</b>	<b>R7 - Calcular precio de pedido</b>
Resumen	El sistema debe permitir a los usuarios calcular el precio subtotal de un pedido
Entradas	- Referencias de productos - Cantidad de productos
Resultado	<ol style="list-style-type: none"> <li>1. Calcula el precio total teniendo en cuenta los productos ingresados, las cantidades y los descuentos de cantidad. <ol style="list-style-type: none"> <li>1.1. De 1 a 5 unidades de la misma categoría se toma el precio base de los productos.</li> <li>1.2. De 6 a 11 unidades de la misma categoría se descuenta un 10% del valor unitario.</li> <li>1.3. De 12 a 119 unidades de la misma categoría se descuenta un 15% del valor unitario.</li> <li>1.4. De 120 unidades en adelante se descuenta un 20% del valor unitario</li> </ol> </li> <li>2. El sistema muestra el precio total del pedido.</li> </ol>

<b>Pasos</b>	<b>Método</b>	<b>Responsable</b>
Obtener del archivo el precio de los productos ingresados y calcular el precio total con descuento	calcular_precio()	Factura

<b>Nombre</b>	<b>R8 - Generar factura</b>
Resumen	El sistema debe permitir a los usuarios generar una factura que contenga el nombre del encargado de la venta, el nombre del cliente, la cantidad de productos y el valor unitario y total
Entradas	- Nombre del vendedor - Nombre de cliente - Referencia de producto - cantidad de producto - precio total
Resultado	<ol style="list-style-type: none"> <li>1. El sistema recibe nombre del vendedor, nombre del cliente, referencia(s) de producto(s) y cantidad de productos.</li> <li>2. El sistema ejecuta R7 para calcular el precio total del pedido.</li> </ol>

	3. El sistema muestra la factura con los datos 4. El sistema guarda la factura en el archivo 5. El sistema muestra el menú de usuario
--	---

Pasos	Método	Responsable
Calcular el precio del pedido	calcular_precio()	Factura
Generar factura con los datos requeridos	generar_factura(nombre_vendedor)	Factura

Nombre	R9 – Consultar historial de facturas
Resumen	El sistema debe permitir a los usuarios consultar todas las facturas generadas
Entradas	Ninguna
Resultado	1. El sistema carga del archivo todas las facturas generadas 2. El sistema muestra todas las facturas generadas 3. El sistema muestra el menú de usuario

Pasos	Método	Responsable
Consultar en el archivo todas las facturas generadas y mostrarlas	consultar_facturas()	Factura



