

## **Proyecto Programado #1**

### ***Sistema de Punto de Venta***

### **Introducción**

Un sistema punto de venta es una versión computarizada de una caja registradora tradicional o de un cuaderno de llevar las cuentas, que gracias a la tecnología actual, añade muchas funcionalidades más que antes no existían o dependían de sistemas muy complejos, grandes y caros.

El software de punto de venta mejora el control administrativo mediante un seguimiento preciso de todas las transacciones de un negocio en tiempo real.

### **Proyecto a desarrollar**

Se deberá crear un sistema de punto de venta en C que permita gestionar las principales funciones de control de inventario, ventas y facturación de productos. El programa se utilizará a través de la consola, no se implementará interfaz gráfica. La información se almacenará y consultará desde el motor de Bases de Datos, se sugiere PostgreSQL o MySQL, aplicando normalización.

El programa deberá desplegar un menú al usuario con dos submenús o grupo de funciones (el menú debe mostrarse recurrentemente después de cada opción realizada -a excepción del salir- y tener la opción de subir al menú principal desde los menús secundarios):

#### **Opciones Administrativas**

La solución debe disponer de una sección de opciones administrativas en donde se accederá a todas estas funcionalidades indicando un usuario y una contraseña (almacenadas en la base de datos). Al indicar el usuario y la contraseña se habilitará un menú donde se habilitarán las siguientes funcionalidades:

- Registro de familias de productos
- Registro de productos

- Cargar inventario
- Consulta de facturas
- Estadísticas

### **Valores iniciales**

Al iniciar la ejecución, la aplicación debe tener disponible la siguiente información: nombre del local comercial, teléfono, cédula jurídica, horario de atención, número secuencial de la siguiente factura, usuario y contraseña de administrador.

### **Registro de familias de productos**

El sistema debe manejar información de familias de productos que vende el local comercial, se debe permitir incluir en lote (incluye con información correcta y únicamente nuevos). La información que se debe almacenar por familias de producto será: Identificador y Descripción Familia.

No pueden existir dos familias con la misma identificación. Reporte de no procesados.

Fam1,Enlatados

Fam2,Granos básicos

### **Registro de productos**

El sistema debe permitir dar mantenimiento de productos que vende el local comercial, se debe permitir incluir o eliminar productos, para incluir se debe hacer en lote (incluye con información correcta y únicamente nuevos), para eliminar manual por medio de código. La información que se debe almacenar por producto serán: identificador, nombre, familia, costo, precio y cantidad en stock.

No pueden existir dos productos con el mismo identificador y no se pueden eliminar productos que se hayan cotizado o facturado. La familia debe existir.

Prod1,Atún,Enlatados,500.3,655.4,45

Prod2,Arroz,Granos básicos,500.3,655.4,45

### **Carga de inventario (actualizar cantidades)**

El sistema debe permitir dar mantenimiento del stock de productos por medio de una carga en lote. Esta información debe cargarse por medio de un archivo que incluirá el id del producto y la cantidad, de cada uno, esta cantidad se sumará o restará al stock existente. Validar datos del archivo (el stock no puede quedar en negativo). Reportar líneas no procesadas.

Prod1,23

Prod2,5

### **Consulta de facturas**

Muestra una lista de todas las facturas generadas por cotizaciones. Por cada factura debe mostrar lo siguiente: identificador, fecha y hora, sub total, total. Al seleccionar una factura se debe permitir mostrar el detalle de la factura, que incluye toda la información que se generó al facturar (encabezado y líneas).

## **Estadísticas**

Se debe mostrar la siguiente información, relacionada a las cotizaciones y facturas históricas del sistema:

- Cantidad de cotizaciones pendientes
- Cantidad de cotizaciones facturadas
- Promedio de compra (promedio de total)
- Top 5 de productos más vendidos
- Producto más vendido por familia
- Monto vendido por familia

## **Opciones Generales**

Para acceder a estas funcionalidades no se requiere contraseña, serán las siguientes funciones:

- Consulta de catálogo
- Cotizar
- Modificar cotización
- Facturar
- Salir

## **Consulta de catálogo**

Muestra una lista de todos los productos que vende el local comercial. Por cada uno debe mostrar lo siguiente: identificador, nombre, descripción, precio sin impuestos y cantidad en stock. Se debe poder filtrar la información por familia de producto opcionalmente.

## **Cotizar**

Se debe permitir la selección de los elementos del catálogo y colocarlos en una lista visible al cliente (nombre, descripción y precio), se debe poder filtrar por familia. Se debe permitir elegir varios productos e indicar la cantidad (detalle). La cotización debe guardarse, para su posterior facturación. Y se guarda en estado pendiente de pago, se genera un número de cotización. Además, se puede seleccionar algún producto agregado a la cotización y removerlo de la misma. El cliente puede elegir la cantidad de productos que desee. Si al agregar un nuevo producto a la cotización éste ya existe, lo que se hace es que se aumenta la cantidad. Se debe validar que exista la cantidad de productos en stock, si no, no se agrega la línea y se informa al usuario.

La información debe incluir la lista de productos, con las cantidades y precios negociados. Después de agregar cada línea se debe preguntar si desea agregar línea, eliminar línea (elimina por número de línea), guardar o salir sin guardar. Se deben mostrar las líneas de la cotización con su número.

### **Modificar cotización**

Se ingresa indicando el número de cotización, permite adicionar o eliminar productos a la cotización creada anteriormente (elimina por número de línea), se muestra la información similar al punto anterior y la cotización se debe guardar en estado pendiente de pago.

### **Facturar**

Se debe indicar un número de cotización y el nombre del cliente, y se entregará una factura (se mostrará en pantalla) al cliente con la siguiente información: identificador, nombre del local comercial, cédula jurídica y teléfono; fecha, cliente, detalle de productos (con precio), subtotal, impuesto de ventas (13% del subtotal) y total (subtotal más impuestos).

Se debe validar que la cotización exista y que esté en estado pendiente de pago. Una vez facturada la cotización se debe cambiar a estado facturado.

Cuando se genera la facturación se deben descontar de stock la cantidad facturada de cada producto.

### **Salir**

Finaliza el programa. Al seleccionar esta opción, se debe asegurar que toda la información del programa se almacena de forma persistente, es decir, que al volver a correr el programa, todos los datos se encuentran disponibles.

### **Otros aspectos**

- a. Cada grupo deberá definir la base de datos a utilizar.
- b. El programa deberá hacer uso de struct.

### **Puntos Extra**

Se darán 10 puntos extra si a la hora de facturar, genera una funcionalidad de verificar el stock a la hora de facturar y generar una opción que se llame Ajustar Cantidades, para verificar que todas las líneas se puedan facturar, en caso que la cantidad sobre pase el stock se modifica la cantidad del producto a facturar (si no hay nada disponible elimina la línea), informar el producto, la cantidad anterior y nueva cantidad. Debe mostrar las líneas ajustadas.

### **Aspectos técnicos**

El proyecto deberá estar escrito en el lenguaje de programación C (no C++), y deberá de funcionar en el sistema operativo elegido por el estudiante, se recomienda Linux. En caso de requerir librerías adicionales para compilar y ejecutar el programa, deberán especificarlo en la documentación, ya que de lo contrario se descontarán puntos en la evaluación.

Deberán utilizar el sistema de control de versiones GitHub, el repositorio deberá ser público o incluir al profesor en el control de acceso de este. Se deben visualizar avances periódicos en el proyecto por parte de los commits.

Debe prevalecer un uso correcto de la memoria por medio de punteros y funciones propias de `stdlib.h` y `string.h`, no se permite el uso de arreglos de caracteres con dimensiones estáticas. No seguir estas disposiciones implicaría penalizaciones en las funcionalidades o consideradas como no desarrolladas.

## Documentación

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de estos.

Para la documentación interna, deberán incluir comentarios descriptivos para cada función, con sus entradas, salidas, restricciones y **objetivo**.

La documentación externa deberá incluir:

1. Portada.
2. Manual de usuario: **instrucciones de compilación, ejecución y uso**.
3. Pruebas de funcionalidad: incluir *screenshots*.
4. Descripción del problema.
5. Diseño del programa: decisiones de diseño, algoritmos usados, diagrama de archivos.
6. Librerías usadas: manejo de archivos, análisis, etc.
7. Análisis de resultados (lista de cotejo): objetivos alcanzados, objetivos no alcanzados, y razones por las cuales no se alcanzaron los objetivos (en caso de haberlos).
8. Bitácora (autogenerada en git, commit por usuario incluyendo comentario).

## Forma de trabajo

El trabajo se debe realizar en grupos de dos.

## Evaluación

La evaluación se va a centrar en dos elementos: programación y documentación.

El proyecto programado tiene un valor de **10%** de la nota final, en el rubro de Proyectos.

Desglose de la evaluación del proyecto programado:

1. Documentación interna 2 ptos.
2. Documentación externa 8 ptos.
3. Funcionalidad 80 ptos (ver detalle en Proyecto a Desarrollar)
4. Revisión del proyecto (según completitud del proyecto y gestión del tiempo) 5 ptos.
5. Hora de Entrega 5 ptos.

## Aspectos administrativos

Debe crear un archivo **.zip** ("PP1\_Est.zip") que contenga únicamente un archivo **info.txt** y 2 carpetas llamadas **documentacion** y **programa**, en la primera deberá incluir el documento de *word* o pdf solicitado y en la segunda los archivos y carpetas necesarias para la implementación de este proyecto programado, y/o link en git del repositorio. El archivo **info.txt** debe contener la siguiente información (cualidades):

- a. Nombre del curso
- b. Número de semestre y año lectivo
- c. Nombre de los Estudiante
- d. Número de carnet de los estudiantes
- e. Número de proyecto programado
- f. Fecha de entrega
- g. Estatus de la entrega (debe ser **CONGRUENTE** con la solución entregada):  
[Deplorable | Regular | Buena | MuyBuena | Excelente | Superior]

## Entrega

Deberá subir el archivo antes mencionado al TEC Digital en el curso de LENGUAJES DE PROGRAMACIÓN GR 60, en la asignación llamada "P1" debajo del rubro de "Proyectos". En la evaluación del Proyecto el rubro de "Hora de Entrega" valdrá por 5 puntos de la nota total del proyecto, según la siguiente escala:

- a. Si se entrega antes de las 11:55:55 **PM** del lunes 31 de marzo de 2025, 5 puntos.
- b. Si se entrega antes de las 11:55:55 **AM** del martes 1 de abril de 2025, 2.5 puntos.
- c. Si se entrega antes de las 11:55:55 **PM** del martes 1 de abril de 2025, 0 puntos.

**NO SE ACEPTARÁN** trabajos que contengan "commits" posterior a esta fecha.

Cada estudiante deberá participar de la revisión donde se demuestre la funcionalidad y la autoría del proyecto.