



Programación II

Partes escritas 1 y 3

BRANDON MUÑOZ GÓMEZ

PROFESOR ALEXANDER BENJAMIN CURLING

Universidad Hispanoamericana

Ingeniería informática

2025

Parte 1

1. Entorno elegido Cafetería universitaria “Café Tibas”.

2. Problema a resolver: Cada día la cafetería vende decenas de productos (cafés, sándwiches, postres, etc.). El administrador debe saber rápidamente:

- Cuántas unidades de cada artículo se han vendido.
- La recaudación total y por categoría.
- El stock actual para decidir cuándo reabastecer. Hoy todo se lleva en papel, se cometen errores y el cierre de caja tarda mucho. Se necesita un pequeño sistema de consola que permita registrar ventas, consultar resúmenes y actualizar existencias sin depender de bases de datos ni clases avanzadas (aún no vistas en el curso).

3. Organización de la información Combinación de vectores y una matriz. • Un vector de strings almacena los nombres de los productos.

- Un vector de doubles almacena el precio de venta correspondiente.
- Una matriz $\text{int} [N][2]$ guarda, para cada producto, [0] stock actual y [1] unidades vendidas. Justificación:

– Los vectores paralelos (nombre, precio) son suficientes para datos estáticos que rara vez cambian.

– La matriz permite dos datos “numéricos” por producto sin crear estructuras anidadas. Con índices coherentes (fila = producto, columna 0 = stock, columna 1 = ventas) todo se mantiene compacto y accesible con operaciones aritméticas simples.

– Esta combinación aprovecha al máximo los temas del curso: recorrido de arreglos, paso de parámetros, funciones para buscar, insertar y actualizar.

4. Significado de cada estructura

- $\text{nombres}[i] \rightarrow$ nombre del producto i .
- $\text{precios}[i] \rightarrow$ precio unitario del producto i .
- $\text{datos}[i][0] \rightarrow$ cantidad en inventario del producto i .
- $\text{datos}[i][1] \rightarrow$ acumulado de unidades vendidas del producto i durante el día.

Así, con un solo índice i puedo relacionar toda la información de un artículo y con el segundo índice (0 o 1) distingo stock vs. ventas sin sobre complicar la lógica en resumen: El sistema en consola para Café Tibas usará dos vectores paralelos (nombres y precios) más una matriz de control (stock y ventas).

Parte 3

Justificación Técnica

1. ¿Por qué elegiste vectores, matrices o ambos?

Elegí usar una combinación de **vectores y una matriz** porque me permitió representar todos los datos necesarios del sistema de forma sencilla y eficiente, sin necesidad de estructuras complejas que todavía no hemos visto en el curso. Usé:

- Un vector `nombres[]` para los nombres de productos.
- Un vector `precios[]` para los precios de venta.
- Una matriz `datos[i, 0]` y `datos[i, 1]` para el stock y la cantidad vendida respectivamente.

Esta combinación me permitió mantener todos los datos relacionados a un producto usando un solo índice. Además, me facilitó aplicar recorridos, validaciones, búsquedas y cálculos sin complicar la lógica. Sentí que es la estructura más clara

¿Qué fue lo más difícil del desarrollo y cómo lo resolviste?

Lo más complicado fue manejar **las validaciones de entrada** para que el usuario no ingresara datos fuera de rango o incorrectos (por ejemplo, letras en lugar de números o cantidades mayores al stock disponible). Para resolver eso, creé dos funciones personalizadas: `LeerEntero()` y `LeerDouble()`, que se encargan de validar y forzar que el usuario repita la entrada hasta que sea válida. Esto me ayudó a mantener el programa estable y evitar errores que lo detuvieran.

También me costó un poco organizar el menú de forma que fuera claro y que cada opción funcionara bien por separado. Usar funciones específicas para registrar ventas, agregar productos y mostrar reportes me ayudó a mantener todo más ordenado.

Si tuvieras que extender el sistema, ¿qué le agregarías y cómo?

Si tuviera que mejorar el sistema, me gustaría agregar:

- Guardado y carga de datos desde archivos, para no tener que ingresar los productos cada vez que se abre el programa.
- Categorías por producto (como bebidas, comidas, postres), agregando un vector más con la categoría y mostrando reportes por tipo.
- Un historial de ventas por día, usando otro arreglo que registre cuántas unidades se vendieron por jornada, así se podrían ver estadísticas semanales.
- También me gustaría agregar búsqueda por nombre para que el usuario pueda escribir parte del nombre del producto en lugar de usar solo números.

¿Qué habilidades y competencias pusiste en práctica durante este examen?

Durante el desarrollo puse en práctica varias cosas que hemos visto en el curso, como:

Lógica de programación: condiciones, ciclos, vectores, arreglos, acumuladores y menú interactivo.

Manejo de arreglos unidimensionales y bidimensionales, especialmente para almacenar y acceder a datos relacionados.

Validación de entradas del usuario, para evitar errores en tiempo de ejecución.

Organización modular: usé funciones para cada parte del programa (ventas, reportes, reabastecimiento), lo que me ayudó a que el código fuera más fácil de entender y mantener.

También reforcé el pensamiento lógico

Resumen del sistema "Café Tibas"

Este programa es un sistema de ventas en consola diseñado para una cafetería universitaria. Permite llevar el control de productos, ventas y stock utilizando **vectores y una matriz**.

Estructuras utilizadas

nombres[]: almacena el nombre de cada producto.

precios[]: guarda el precio de cada producto.

datos[,]: matriz con 2 columnas: [0] para el **stock** disponible y [1] para las **unidades vendidas**.

nProductos: lleva la cuenta de cuántos productos han sido registrados.

Funciones de apoyo

ListarProductos(): muestra todos los productos con su índice, nombre, stock y precio.

LeerEntero() / LeerDouble(): validan y capturan entradas numéricas del usuario dentro de un rango permitido.

CargarProductosDemo(): carga 5 productos de ejemplo para facilitar pruebas.