

VMS store

Anyela Valentina Sánchez Valencia

Mario Guillermo Salcedo Muñoz

Sara Sofía Camacho Sierra

Facultad FIMEB, Universidad Antonio Nariño

Lógica Computacional




VMS store

Mario Guillermo Salcedo Muñoz

Anyela Valentina Sánchez Valencia

Sara Sofía Camacho Sierra

Este sistema de gestión de licencias es un programa diseñado para administrar compras, ventas y promociones de licencias de videojuegos. Este sistema permite a los usuarios consultar información detallada de juegos, realizar compras y ventas de licencias, aplicar descuentos automáticos y generar reportes de ventas.

 DIRECCIÓN NACIONAL DE DERECHO DE AUTOR Unidad Administrativa Especial Ministerio del Interior			
OFICINA DE REGISTRO			
SOLICITUD INSC RIPCIÓN SOPORTE LÓGICO (SOFTWARE)			
1. DATOS DEL AUTOR O AUTORES			
Nombre:	Mario Guillermo Salcedo Muñoz	Doc Ident:	T.I 1120818341 De: Acacias—Meta
Nacionalidad:	Colombiana	Dirección completa y ciudad:	Calle 22 Sur #12G-26, Bogotá
Seudónimo (Allegar escritura: Ver instrucción 1.2):	No Aplica	Fecha de defunción:	No Aplica
Nombre:	Sara Sofia Camacho Sierra	Doc Ident:	T.I 1018430500 De: Bogotá—Cundi
Nacionalidad:	Colombiana	Dirección completa y ciudad:	Transversal 63 #68F-27 Sur, Bogotá
Seudónimo (Allegar escritura: Ver instrucción 1.2):	No Aplica	Fecha de defunción:	No Aplica
Nombre:	Anyela Valentina Sánchez Valencia	Doc Ident:	T.I 1059241639 De: Bogotá—Cundi
Nacionalidad:	Colombiana	Dirección completa y ciudad:	Carrera 6A #22A-30 Sur, Bogotá
Seudónimo (Allegar escritura: Ver instrucción 1.2):	No Aplica	Fecha de defunción:	No Aplica
2. DATOS DEL PRODUCTOR			
Nombre :	Mario Guillermo Salcedo Muñoz		
Correo electrónico:	msalcedo341@uan.edu.co	Sitio web:	Nit:
Dirección:	Calle 22 Sur #12G-26, Bogotá	Ciudad:	Acacias Meta País: Colombia
3. DATOS DE LA OBRA			
Título:	VMS Store		
Año de Creación:	2025	<input type="checkbox"/> Inédito	
Pais de Origen:	Colombia	<input checked="" type="checkbox"/> Publicado	Año de Publicación: 2025
Carácter de la obra (Ver instrucción 3.1):			
<input type="checkbox"/> Obra individual <input checked="" type="checkbox"/> Obra colectiva <input type="checkbox"/> Obra originaria <input type="checkbox"/> Obra anónima <input type="checkbox"/> Obra seudónima <input type="checkbox"/> Obra en colaboración <input type="checkbox"/> Obra por encargo <input type="checkbox"/> Obra derivada <input type="checkbox"/> Obra póstuma <input type="checkbox"/> Otra (Especificar):			
4. ELEMENTOS DEL SOPORTE LÓGICO (SOFTWARE) APORTADOS			
A. Programa de Computador <input checked="" type="checkbox"/> B. Descripción de Programa <input checked="" type="checkbox"/> C. Material Auxiliar <input checked="" type="checkbox"/>			
Breve Descripción de Funciones: Sistema en línea que simula una aplicación donde se pueden realizar varias funciones como vender y comprar licencias o visualizar la información detallada de un juego en específico.			
5. TRANSFERENCIAS 			
6. OBSERVACIONES GENERALES			
7. DATOS DEL SOLICITANTE			
Nombre:	Sara Sofia Camacho Sierra	Doc. Ident:	T.I 1018430500 De: Bogotá—Cundi
Nacionalidad:	Colombiana	Dirección:	Transversal 63 #68F-27 Sur, Bogotá Teléfono: 3053494155
Correo electrónico:	scamacho500@uan.edu.co	Sitio web:	Nit:
Ciudad:	Bogotá	País:	Colombia En representación de:
			
Sara Sofia Camacho Sierra Firma del Solicitante			

Tu Puerta de Entrada al Mundo de los Videojuegos

En la era digital actual, los videojuegos se han convertido en una parte importante de nuestro entretenimiento. Desde emocionantes historias y aventuras, hasta desafiantes rompecabezas, existe un sin fin de universos de experiencias. Con este sistema nos propusimos crear una herramienta intuitiva y funcional que simule de manera adecuada la experiencia de una tienda de aplicaciones de videojuegos.

Este programa, desarrollado en c++, implementa un sistema de gestión de ventas de licencias digitales para una tienda virtual llamada VMS Store. El sistema permite:

- Visualizar información detallada sobre cada juego.
- Comprar y vender licencias digitales de manera interactiva.
- Identificar los títulos más vendidos.
- Consultar compatibilidad de los juegos con distintas plataformas.
- Aprovechar descuentos por volumen de compra según la categoría del juego.

Propósito del documento

El propósito de esta guía es suministrar toda la información necesaria para que el usuario pueda interactuar adecuadamente con la aplicación. Aquí encontrarás desde una descripción general del sistema y sus funcionalidades hasta los requisitos técnicos para su ejecución, pasos de instalación, preguntas frecuentes y documentación técnica asociada.

Estructura del documento

Este manual está organizado en secciones claras para facilitar la navegación:

- **Introducción:** Presenta una descripción general del sistema, su finalidad y los cambios incorporados en la versión actual (bitácora).
- **Requisitos previos:** Define los conocimientos y recursos necesarios para ejecutar correctamente la aplicación.
- **Instalación y configuración:** Instrucciones para compilar y ejecutar el programa en distintos entornos.
- **Guía de uso:** Explica paso a paso cómo utilizar las funciones principales del sistema, como consultar información de juegos, comprar o vender licencias, y aplicar descuentos.

- **Preguntas frecuentes (FAQ):** Contiene respuestas a dudas comunes sobre el uso del programa.
- **Documentación técnica:** Incluye diagramas del diseño del software, como el **diagrama entidad-relación, casos de uso y clases.**
- **Datos de contacto:** Información para contactar a los desarrolladores/autores en caso de dudas, comentarios o contribuciones.

Cómo usar esta guía

Para navegar este documento de manera efectiva:

1. Si es tu primera vez con el sistema, comienza por la **introducción** y los **requisitos previos.**
2. Si ya tienes conocimientos básicos, puedes ir directamente a la sección de **instalación** y luego a la **guía de uso.**
3. Para entender el diseño del sistema o realizar mejoras, consulta la **documentación técnica.**
4. Si surgen dudas, la sección de **preguntas frecuentes** puede ayudarte rápidamente.
5. Para sugerencias o soporte, puedes consultar los **datos de contacto de los autores.**

Este manual ha sido preparado para ofrecer una experiencia completa y estructurada, tanto a usuarios nuevos como a programadores interesados en el funcionamiento interno del sistema.

v.

Introducción

a. Este programa simula una tienda virtual de videojuegos llamada VMS Store **desarrollada en lenguaje C++.** Su principal objetivo es **ofrecer una experiencia interactiva que permita al usuario gestionar licencias digitales de videojuegos, simulando funcionalidades similares a las de plataformas reales como Steam, PlayStation Store o App Store.**

A través de un menú interactivo basado en estructuras de control y datos, este proyecto ofrece una oportunidad educativa para comprender cómo se diseñan y ejecutan operaciones típicas de un sistema de ventas digital.

Esperamos que, al interactuar con este código, no solo encuentres la información que necesitas, sino que también comprendas mejor los principios fundamentales de la **programación orientada a objetos y la manipulación de datos en C++.**

b. Bitácora de actualizaciones

- 1) Marzo 22. Primera versión del código, solo se cuenta con un menú que muestra los requerimientos funcionales que cumple el código, tales como visualizar la información detallada de un juego o vender licencias.
- 2) Marzo 30. Se le ha agregado al código los atributos con los que contarán los distintos juegos que estarán disponibles en la aplicación.
- 3) Abril 13. Se agregan 4 juegos a la aplicación, cada uno con sus respectivos atributos.
- 4) Abril 26. Se cuenta con un caso para cada requerimiento con el que cumple el sistema (Detallar la información de cada juego, comprar o vender licencias, etc)
- 5) Mayo 11. Todos los casos están completos, los 5 requerimientos funcionales están funcionando de la manera correcta.
- 6) Mayo 18. Se agrega una matriz al código que relaciona a cada juego con sus respectivos dispositivos en los cuales es compatible.
- 7) Mayo 30, última actualización. Se agrega un carrito de compra, que permite agregar más de una licencia por transacción. Código completado. (Al final el código permite con solo ingresar una ID de juego conocer su información detallada, comprar o vender licencias de dicho juego, además de poder acceder a descuentos por volumen de compra, por último, saber la compatibilidad de dispositivos con determinado juego.)

vi.

Sección de requisitos previos necesarios para usar el sistema

a. Conocimientos Mínimos del Usuario

- **Conocimiento Básico.** El usuario debe saber el cómo interactuar con una computadora con una consola o terminal, incluyendo la entrada de datos e interpretación en la salida de los mismo.
- **Familiaridad con Menús:** El usuario debe estar familiarizado para navegar por un sistema basado en un menú, manejándolo únicamente por opciones numéricas.
- **Conceptos Básicos de Compras y Ventas:** El usuario debe manejar los conceptos más elementos del programa dado que estos son los aspectos principales dentro de los requerimientos funcionales, los descuentos, precios y compatibilidad engloban la AppStore.
- **Lenguaje:** El usuario debe manejar la comprensión del idioma español que es en el que fue escrito el código.

b. Requisitos Técnicos Previos (Capacidades técnicas mínimas del equipo, Software asociado necesario)

- **Procesador:** Un procesador básico (ej. Intel Core i3 o equivalente) capaz de ejecutar aplicaciones de consola sin retrasos.

- **Memoria RAM:** Mínimo 2 GB de RAM para garantizar un rendimiento fluido del programa.
- **Almacenamiento:** Espacio libre de al menos de 100 MB en el disco duro para instalar el compilador y almacenar el programa ejecutable.
- **Sistema Operativo:** Compatible con sistemas operativos comunes como: Windows (7, 10, 11), Linux (distribuciones como Ubuntu, Fedora, etc.) , macOS
- **Entrada/Salida:** Teclado para ingresar datos y un monitor o pantalla para visualizar la salida del programa.
- **Resolución de Pantalla:** Mínima de 800x600 píxeles para una visualización perfecta de la interfaz de la consola.
- **Compilador de C++:** Un compilador compatible con el estándar de C++.
- **Terminal /Consola:** Software para interactuar con el programa como:

Windows: CMD o PowerShell

Linux/macOS: Terminal Nativa

Dependencias no se requieren, el programa únicamente usa la biblioteca estándar de C++ (iostream) y el espacio de nombres std.

c. Mecanismo para acceder al sistema

Obtención del Código: El usuario debe tener acceso al código o descargarlo para hacer uso del mismo.

Se ha de usar el código en un compilador que acepte C++ para el correcto uso del mismo.

vii.

Una sección de instalación y configuración (No aplica)

viii.

Guía de Uso de la App Store

Esta guía te ayudará a interactuar con la aplicación de App Store y a utilizar cada una de sus funcionalidades.

1. Visualizar Información Detallada de un Juego

Para ver los detalles de un juego:

- Selecciona la opción 1 en el menú principal.
- La aplicación te pedirá que ingreses el ID del juego que deseas consultar.
- Después de ingresar el ID, la aplicación mostrará el nombre, ID, categoría, tamaño, precio, licencias disponibles y licencias vendidas de ese juego.
- Si el ID no corresponde a ningún juego, recibirás un mensaje indicando que no se encontró el juego.

2. Comprar Licencias del Juego

Para adquirir licencias de un juego:

- Selecciona la opción 2 en el menú principal.
- La aplicación te pedirá que ingreses el ID del juego del que deseas comprar licencias.
- Luego, te solicitará la cantidad de licencias que deseas adquirir.
- La aplicación calculará el total a pagar, aplicando automáticamente los descuentos si cumples con las políticas (por ejemplo, si compras 25 o más licencias de juegos de rompecabezas, o una combinación específica de juegos de deportes y acción).
- Se te mostrará el total a pagar. Si no hay suficientes licencias disponibles, recibirás un mensaje de error.

3. Vender Licencias del Juego

Para vender licencias de un juego:

- Selecciona la opción 3 en el menú principal.
- La aplicación te pedirá que ingreses el ID del juego del que deseas vender licencias.
- Luego, te solicitará la cantidad de licencias que deseas vender.
- La aplicación calculará el total de la venta y te lo mostrará.
- Si el ID del juego no se encuentra, recibirás un mensaje de error.

4. Consultar el Juego Más Vendido

Para saber cuál es el juego más vendido:

- Selecciona la opción 4 en el menú principal.
- La aplicación procesará la información y te mostrará directamente el nombre, ID y la cantidad de licencias vendidas del juego que ha tenido más éxito.

5. Consultar los Descuentos Aplicados por Volumen de Compra

Para conocer las políticas de descuento:

- Selecciona la opción 5 en el menú principal.
- La aplicación te presentará un mensaje detallando las condiciones para obtener descuentos, como el porcentaje de descuento y las categorías/cantidades de licencias requeridas para aplicarlos.

6. Consultar Compatibilidad de Juegos con Plataformas

Para verificar la compatibilidad de los juegos:

- Selecciona la opción 6 en el menú principal.
- La aplicación mostrará una lista de cada juego y las plataformas específicas (PC, Teléfono, Tablet, Consola) con las que es compatible.

7. Salir de la Aplicación

Para salir de la App Store:

- Selecciona la opción 7 en el menú principal.
- La aplicación te mostrará un mensaje de despedida y finalizará.

Nota: En cualquier momento, si ingresas una opción no válida en el menú principal, la aplicación te pedirá que escojas una opción válida nuevamente.

Ix

Sección de preguntas frecuentes. (FAQ)

1. ¿Qué hace este programa?

El programa simula una tienda de aplicaciones (Appstore) donde puedes visualizar información de juegos, comprar o vender licencias, consultar el juego más vendido, revisar descuentos por volumen y verificar la compatibilidad de los juegos con diferentes plataformas.

2. ¿Qué opciones están disponibles en el menú?

El menú incluye:

- 1: Visualizar información detallada de un juego por ID.
- 2: Comprar licencias de un juego, con posibles descuentos.
- 3: Vender licencias de un juego.

- 4: Consultar el juego más vendido.
- 5: Ver descuentos aplicados por volumen de compra.
- 6: Consultar compatibilidad de juegos con plataformas.
- 7: Salir del programa.

3. ¿Qué sucede si ingreso un ID de juego inválido?

Si el ID no corresponde a ningún juego (1 al 6), el programa mostrará el mensaje "No se encontró un juego con ese ID" y permitirá continuar.

4. ¿Cómo funcionan los descuentos al comprar licencias?

- **Rompecabezas:** Si compras 25 o más licencias de un juego como "Jigsawscapes" (ID 2), se aplica un 20% de descuento al total.
- **Deportes y Acción:** Si compras 20 o más licencias de "Haikyuu" (ID 1) y aceptas comprar 15 o más de "Bullet Echo" (ID 3), se aplica un 15% de descuento al total combinado.
- **Acción y Deportes:** Si compras 15 o más licencias de "Bullet Echo" (ID 3) y aceptas comprar 20 o más de "Haikyuu" (ID 1), se aplica un 15% de descuento al total combinado.
- Si se cumplen varias condiciones, se prioriza el descuento del 20%.

5. ¿Qué pasa si no hay suficientes licencias disponibles para comprar?

Si intentas comprar más licencias de las disponibles, el programa muestra "No hay suficientes licencias" y no procesa la compra.

6. ¿El programa actualiza las licencias disponibles o vendidas después de una transacción?

No, el código actual no actualiza los valores de `cantidadlicenciasdisponibles` ni `cantidadlicenciasvendidas` tras comprar o vender. Las transacciones solo calculan y muestran el total a pagar o recibir.

7. ¿Qué plataformas son compatibles con los juegos?

El programa incluye una matriz de compatibilidad para 4 plataformas (PC, Teléfono, Tablet, Consola). Al seleccionar la opción 6, se muestra qué plataformas soporta cada juego (1 indica compatible, 0 no).

8. ¿Qué pasa si ingreso una opción inválida en el menú?

Si seleccionas un número fuera de las opciones 1 a 7, el programa muestra "Escoge nuevamente una opción válida" y te permite intentarlo de nuevo. Si ya estás muy avanzado en una compra e ingresas mal ciertos datos, el sistema dará error y saldrá de su ejecución.

9. ¿Necesito conexión a internet para usar el programa?

No, es una aplicación de consola que se ejecuta localmente en tu equipo. No requiere acceso a internet.

10. ¿Cómo salgo del programa?

Selecciona la opción 7 en el menú. El programa mostrará "Has salido de la aplicación" y "Gracias por usar la app" antes de cerrarse.

11. ¿Qué unidades se usan para el tamaño y el precio de los juegos?

- **Tamaño**: Se mide en megabytes (MB).
- **Precio**: Se expresa en dólares (\$).

12. ¿Puedo modificar la lista de juegos?

Sí, pero requiere cambiar el código. La lista de juegos está definida en un arreglo de estructuras `juego listaJuego[6]` en la función `main`. Puedes editar los valores o agregar más entradas, ajustando el tamaño del arreglo y el bucle correspondiente.

13. ¿Es seguro ingresar cualquier cantidad de licencias?

El programa verifica si hay suficientes licencias disponibles al comprar, pero no valida si los valores ingresados son negativos o no numéricos. Ingresar datos inválidos (como letras) puede causar errores o comportamiento inesperado.

14. ¿Cómo se determina el juego más vendido?

El programa compara el campo `cantidadlicenciasvendidas` de cada juego y muestra el que tiene el valor más alto, incluyendo su nombre, ID y número de licencias vendidas.

15. ¿Qué hago si el programa no se ejecuta?

- Asegúrate de tener un compilador de C++ instalado
- Verifica que el archivo esté guardado como `.cpp`.
- Revisa si hay errores de sintaxis o problemas con el entorno.

X

Sección de documentación técnica con el diseño de software

a. Diagrama Entidad-Relación (ER)

Descripción

El diagrama ER modela las entidades principales del sistema y sus relaciones, enfocándose en los datos de los juegos y las transacciones.

Diagrama ER (En texto)

- **Entidad: Juego**
 - o **Atributos:**
 - ♣ idJuego (entero, clave primaria): Identificador único del juego.
 - ♣ nombre (cadena): Nombre del juego.
 - ♣ categoria (cadena): Género del juego (e.g., Deportes, Rompecabezas).
 - ♣ tamaño (flotante): Tamaño en MB.
 - ♣ precio (flotante): Costo por licencia en dólares.
 - ♣ cantidadlicenciasdisponibles (entero): Licencias disponibles para compra.
 - ♣ cantidadlicenciasvendidas (entero): Licencias ya vendidas.
- **Entidad: Plataforma**
 - o **Atributos:**

- ♣ nombrePlataforma (cadena, clave primaria): Nombre de la plataforma (e.g., PC, Teléfono).

- **Relación: Compatibilidad**

- o Representa si un juego es compatible con una plataforma.

- o **Atributos:**

- ♣ compatible (entero): 1 si es compatible, 0 si no.

- o **Relación:** Juego (1) --- Compatibilidad --- (N) Plataforma

- **Notas:**

- o Los datos de juegos están predefinidos en un arreglo.
- o La compatibilidad se almacena en una matriz estática de 6 juegos x 4 plataformas.

[Juego] --(idJuego, nombre, categoria, tamaño, precio, cantidadlicenciasdisponibles, cantidadlicenciasvendidas)

|

| Compatibilidad (compatible: 1 o 0)

|

[Plataforma] --(nombrePlataforma)

Diagrama de Casos de Uso

Descripción

El diagrama de casos de uso describe las interacciones entre el usuario y el sistema, representando las funcionalidades disponibles en el menú.

Actores

- **Usuario:** Persona que interactúa con la Appstore a través de la consola.

Casos de Uso

1. Visualizar Información de un Juego

- o **Actor:** Usuario
- o **Descripción:** Ingresa el ID del juego para ver detalles como nombre, categoría, tamaño, precio y licencias.
- o **Precondición:** ID válido (1-6).
- o **Flujo:** Selecciona opción 1, ingresa ID, sistema muestra datos o mensaje de error.

2. Comprar Licencias

- o **Actor:** Usuario

- o **Descripción:** Compra licencias de un juego, con descuentos según categoría y cantidad.
 - o **Precondición:** Licencias disponibles suficientes.
 - o **Flujo:** Selecciona opción 2, ingresa ID y cantidad, sistema calcula total, aplica descuentos si aplica.
3. **Vender Licencias**
- o **Actor:** Usuario
 - o **Descripción:** Vende licencias de un juego y calcula el monto a recibir.
 - o **Flujo:** Selecciona opción 3, ingresa ID y cantidad, sistema muestra total de venta.
4. **Consultar Juego Más Vendido**
- o **Actor:** Usuario
 - o **Descripción:** Muestra el juego con más licencias vendidas.
 - o **Flujo:** Selecciona opción 4, sistema compara y muestra nombre, ID y licencias vendidas.
5. **Consultar Descuentos**
- o **Actor:** Usuario
 - o **Descripción:** Muestra reglas de descuentos por volumen.
 - o **Flujo:** Selecciona opción 5, sistema lista descuentos (e.g., 20% para rompecabezas, 15% para deportes+acción).
6. **Consultar Compatibilidad**
- o **Actor:** Usuario
 - o **Descripción:** Muestra plataformas compatibles por juego.
 - o **Flujo:** Selecciona opción 6, sistema muestra lista de juegos y plataformas compatibles.
7. **Salir**
- o **Actor:** Usuario
 - o **Descripción:** Finaliza la ejecución del programa.
 - o **Flujo:** Selecciona opción 7, sistema muestra mensaje de salida.

Diagrama de Casos de Uso (En texto)

[Usuario]

|

|---(1) Visualizar Información de un Juego

|---(2) Comprar Licencias

|---(3) Vender Licencias

|---(4) Consultar Juego Más Vendido

- |---(5) Consultar Descuentos
- |---(6) Consultar Compatibilidad
- |---(7) Salir
- |

[Sistema Appstore]

c. Diagrama de Clases

Descripción

El diagrama de clases representa la estructura estática del programa, mostrando la clase principal y sus métodos.

Clases

- **Clase: juego**
 - o **Atributos:**
 - ♣ idJuego: int (Identificador único)
 - ♣ nombre: string (Nombre del juego)
 - ♣ categoria: string (Género del juego)
 - ♣ tamaño: float (Tamaño en MB)
 - ♣ precio: float (Costo por licencia)
 - ♣ cantidadlicenciasdisponibles: int (Licencias disponibles)
 - ♣ cantidadlicenciasvendidas: int (Licencias vendidas)
 - o **Métodos:**
 - ♣ mostrarnombre(): void (Muestra el nombre)
 - ♣ mostraridJuego(): void (Muestra el ID)
 - ♣ mostrarcategoria(): void (Muestra la categoría)
 - ♣ mostrartamano(): void (Muestra el tamaño)
 - ♣ mostrarprecio(): void (Muestra el precio)
 - ♣ mostrarcantidadlicenciasdisponibles(): void (Muestra licencias disponibles)
 - ♣ mostrarcantidadlicenciasvendidas(): void (Muestra licencias vendidas)

Diagrama de Clases (en notación textual)

Clase: juego

+-----+

| Atributos: |

```

| - idJuego: int          |
| - nombre: string        |
| - categoria: string     |
| - tamaño: float         |
| - precio: float         |
| - cantidadlicenciasdisponibles: int |
| - cantidadlicenciasvendidas: int   |
+-----+
| Métodos:                |
| + mostrarnombre(): void   |
| + mostraridJuego(): void  |
| + mostrarcategoria(): void |
| + mostrartamano(): void   |
| + mostrarprecio(): void   |
| + mostrarcantidadlicenciasdisponibles(): void |
| + mostrarcantidadlicenciasvendidas(): void   |
+-----+

```

Diagrama de Clases (En texto)

Clase: juego

```

+-----+
| Atributos:                |
| - idJuego: int            |
| - nombre: string          |

```



```

| - categoria: string      |
| - tamaño: float         |
| - precio: float          |
| - cantidadlicenciasdisponibles: int |
| - cantidadlicenciasvendidas: int   |
+-----+
| Métodos:                |
| + mostrarnombre(): void   |
| + mostraridJuego(): void  |
| + mostrarcategoria(): void |
| + mostrartamano(): void   |
| + mostrarprecio(): void   |
| + mostrarcantidadlicenciasdisponibles(): void |
| + mostrarcantidadlicenciasvendidas(): void   |
+-----+

```

xi.

Datos de contacto de los autores

Mario Guillermo Salcedo Muñoz / 3125650928 / msalcedo341@uan.edu.co

Sara Sofia Camacho Sierra / 3054394155 / scamacho500@uan.edu.co

Anyela Valentina Sánchez Valencia / 3052384200 / asanchez639@uan.edu.co

b. Entrega el programa con la última versión generada.

```
#include <iostream>
```

```
using namespace std;
```

```
struct juego
```

```
{
```

```
    int idJuego;
```

```
    string nombre;
```

```
    string categoria;
```

```
    float tamaño;
```

```
    float precio;
```

```
    int cantidadlicenciasdisponibles;
```

```
int cantidadlicenciasvendidas;

void mostrarnombre() {
    cout << "Nombre del juego: " << nombre << endl;
}

void mostraridJuego() {
    cout << "ID del juego: " << idJuego << endl;
}

void mostrarcategoria() {
    cout << "Categoria: " << categoria << endl;
}

void mostrartamano() {
    cout << "Tamaño: " << tamaño << " MB" << endl;
}

void mostrarprecio() {
    cout << "Precio: $" << precio << endl;
}

void mostrarcantidadlicenciasdisponibles() {
    cout << "Licencias disponibles: " << cantidadlicenciasdisponibles << endl;
}

void mostrarcantidadlicenciasvendidas() {
    cout << "Licencias vendidas: " << cantidadlicenciasvendidas << endl;
}

};

int main()
{
```

```

int z, licenciasdeseadas, idJuego, i, confirmacion, continuar;

float totalpagar = 0, montoPagado, cambio;

int carrito[6] = {0, 0, 0, 0, 0, 0}; // Arreglo para almacenar licencias por juego en el
carrito

int deportesLicencias = 0, accionLicencias = 0, rompecabezasLicencias = 0;

cout << "Bienvenido a la Appstore" << endl;

cout << "¿Qué desea realizar en la Appstore? Digite un número de caso" << endl;


// Lista de juegos
juego listaJuego[6] = {
    {1, "Haikyuu", "Deportes", 500, 9.99, 100, 50},
    {2, "Jigsawscapes", "Rompecabezas", 200, 4.99, 150, 80},
    {3, "Bullet Echo", "Acción", 600, 7.99, 200, 120},
    {4, "WSC", "Aventura", 300, 5.99, 250, 150},
    {5, "FIFA 17", "Deportes", 700, 12.49, 180, 90},
    {6, "Candy Crush", "Rompecabezas", 150, 3.99, 300, 200}
};

do
{
    cout << "1 para visualizar información de cada juego" << endl;
    cout << "2 para comprar licencias de un juego" << endl;
    cout << "3 para vender las licencias de un juego" << endl;
    cout << "4 para consultar el juego más vendido" << endl;
    cout << "5 para consultar los descuentos aplicados por volumen de compra" <<
endl;

```

```
    cout << "6 para consultar la compatibilidad de los juegos en distintas
plataformas" << endl;

    cout << "7 para salir" << endl;

    cin >> z;

    switch (z)
    {
    case 1:
        cout << "Visualizar la información detallada de cada juego" << endl;
        cout << "Ingresa ID del juego: ";
        cin >> idJuego;

        for (i = 0; i < 6; i++)
        {
            if (listaJuego[i].idJuego == idJuego)
            {
                listaJuego[i].mostrarnombre();
                listaJuego[i].mostraridJuego();
                listaJuego[i].mostrarcategoria();
                listaJuego[i].mostrartamano();
                listaJuego[i].mostrarprecio();
                listaJuego[i].mostrarcantidadlicenciasdisponibles();
                listaJuego[i].mostrarcantidadlicenciasvendidas();
                break;
            }
        }
        if (i == 6)
```

```
{  
    cout << "No se encontró un juego con ese ID." << endl;  
}  
break;
```

case 2:

```
do  
{  
    cout << "Comprar licencias de un juego" << endl;  
    cout << "Ingrese ID del juego del que desea comprar licencias" << endl;  
    cin >> idJuego;  
    for (i = 0; i < 6; i++)  
    {  
        if (listaJuego[i].idJuego == idJuego)  
        {  
            listaJuego[i].mostrarnombre();  
            listaJuego[i].mostrarcategoria();  
            listaJuego[i].mostrarcantidadlicenciasdisponibles();  
            listaJuego[i].mostrarprecio();  
            cout << "Ingrese la cantidad de licencias que desea comprar: ";  
            cin >> licenciasdeseadas;  
            if (licenciasdeseadas > listaJuego[i].cantidadlicenciasdisponibles)  
            {  
                cout << "No hay suficientes licencias" << endl;  
                return 0;  
            }  
            carrito[i] += licenciasdeseadas; // Agregar al carrito
```

```
        cout << licenciasdeseadas << " licencias de " << listaJuego[i].nombre <<
" agregadas al carrito." << endl;
```

```
        break;
```

```
    }
```

```
}
```

```
if (i == 6)
```

```
{
```

```
    cout << "No se encontró juego para esa ID" << endl;
```

```
    return 0;
```

```
}
```

```
cout << "Desea agregar otro juego al carrito? (1 para sí, 0 para no): ";
```

```
cin >> continuar;
```

```
if (continuar != 0 && continuar != 1)
```

```
{
```

```
    cout << "Has ingresado una opción no válida" << endl;
```

```
    return 0; // Forzar salida del bucle si la entrada es inválida
```

```
}
```

```
} while (continuar == 1);
```

```
// Calcular total y aplicar descuentos
```

```
totalpagar = 0;
```

```
deportesLicencias = 0;
```

```
accionLicencias = 0;
```

```
rompecabezasLicencias = 0;
```

```
cout << "\n--- Contenido del Carrito ---" << endl;
```

```
for (i = 0; i < 6; i++)
```

```

{
    if (carrito[i] > 0)
    {
        cout << "Juego: " << listaJuego[i].nombre << ", Licencias: " << carrito[i]
        << ", Subtotal: $" << carrito[i] * listaJuego[i].precio << endl;

        totalpagar += carrito[i] * listaJuego[i].precio;

        if (listaJuego[i].categoria == "Deportes")

            deportesLicencias += carrito[i];

        else if (listaJuego[i].categoria == "Acción")

            accionLicencias += carrito[i];

        else if (listaJuego[i].categoria == "Rompecabezas")

            rompecabezasLicencias += carrito[i];

    }
}

cout << "Total sin descuento: $" << totalpagar << endl;

// Verificar políticas de descuento

if (rompecabezasLicencias >= 25)
{
    cout << "Por haber comprado al menos 25 licencias de juegos en la sección
    de Rompecabezas, recibirás un descuento del 20%" << endl;

    totalpagar = 0.8 * totalpagar;
}

else if (deportesLicencias >= 20 && accionLicencias >= 15)
{
    cout << "Por haber comprado al menos 20 licencias de juegos en la sección
    de Deportes y 15 de Acción, recibirás un descuento del 15%" << endl;

    totalpagar = 0.85 * totalpagar;
}

```



```

    }

    cout << "Total a pagar con descuentos: $" << totalpagar << endl;

    // Proceso de pago

    cout << "Ingrese la cantidad de dinero con la que pagará: $";

    cin >> montoPagado;

    if (montoPagado < totalpagar)
    {
        cout << "El monto ingresado no es suficiente para cubrir el total a pagar."
<< endl;
    }
    else
    {
        cambio = montoPagado - totalpagar;

        cout << "Pago exitoso! Su cambio es: $" << cambio << endl;

        // Actualizar licencias disponibles y vendidas

        for (i = 0; i < 6; i++)
        {
            if (carrito[i] > 0)
            {
                listaJuego[i].cantidadlicenciasdisponibles -= carrito[i];

                listaJuego[i].cantidadlicenciasvendidas += carrito[i];

                carrito[i] = 0; // Limpiar carrito
            }
        }
    }

    break;

```

case 3:

```
cout << "Vender licencias de un juego" << endl;

cout << "Ingrese ID del juego del que desea vender licencias" << endl;

cin >> idJuego;

for (i = 0; i < 6; i++)
{
    if (listaJuego[i].idJuego == idJuego)
    {
        listaJuego[i].mostrarnombre();

        listaJuego[i].mostrarcantidadlicenciasvendidas();

        cout << "Ingrese la cantidad de licencias que desea vender: ";

        cin >> licenciasdeseadas;

        totalpagar = licenciasdeseadas * listaJuego[i].precio;

        cout << "Su total de venta es $" << totalpagar << endl;

        break;
    }
}

if (i == 6)
{
    cout << "No se encontró juego para esa ID" << endl;
}

break;
```

case 4:

```
{

    cout << "Consultar el juego más vendido" << endl;
```

```

juego* masVendido = &listaJuego[0];

for (i = 1; i < 6; i++) {

    if (listaJuego[i].cantidadlicenciasvendidas > masVendido-
>cantidadlicenciasvendidas) {

        masVendido = &listaJuego[i];

    }

}

cout << "\n--- JUEGO MAS VENDIDO ---" << endl;

masVendido->mostrarnombre();

masVendido->mostraridJuego();

cout << "Licencias vendidas: " << masVendido->cantidadlicenciasvendidas <<
endl;

cout << "-----" << endl;

break;

}

```

case 5:

```

cout << "Consultar los descuentos aplicados por volumen de compra" << endl;

cout << "Si solicitas al menos 25 licencias de juegos de rompecabezas, se dará
un 20% de descuento sobre el total del pedido." << endl;

cout << "Si solicitas al menos 20 licencias de juegos de deportes y 15 de acción,
se hará un 15% de descuento." << endl;

cout << "Si se cumplen ambas condiciones, se aplicará solo el descuento del
20%." << endl;

break;

```

case 6:

```

{

    // Matriz de compatibilidad para 6 juegos y 4 plataformas

```

```

// Columnas: 0 = PC, 1 = Teléfono, 2 = Tablet, 3 = Consola

int compatibilidad[6][4] = {
    {1, 0, 0, 1}, // Haikyuu
    {1, 1, 1, 0}, // Jigsawsapes
    {1, 1, 1, 1}, // Bullet Echo
    {1, 0, 0, 1}, // WSC
    {1, 0, 1, 1}, // FIFA 17
    {0, 1, 1, 1} // Candy Crush
};

string plataformas[4] = {"PC", "Teléfono", "Tablet", "Consola"};

cout << "\n--- Compatibilidad de Juegos con Plataformas ---" << endl;
for (int i = 0; i < 6; i++) {
    cout << "Juego: " << listaJuego[i].nombre << " => Compatible con: ";
    for (int j = 0; j < 4; j++) {
        if (compatibilidad[i][j] == 1) {
            cout << plataformas[j] << " ";
        }
    }
    cout << endl;
}
break;
}

case 7:
    cout << "Has salido de la aplicación" << endl;

```

```
break;
```

```
default:
```

```
cout << "Has ingresado una opción no válida" << endl;
```

```
// Regresar al menú inicial
```

```
continue;
```

```
}
```

```
} while (z != 7);
```

```
cout << "Gracias por usar la app" << endl;
```

```
return 0;
```

```
}
```