



## Cátedra Tecnología de Sistemas

[Introducción a la Programación]

Código: [00831]

**Proyecto. Valor 3%**

### Temas de Estudio

1. Tema 1 y Tema 2 vistos anteriormente
2. Tema 4 Entrada y salida de datos e información

Subtemas

- a. Capítulo VI - Entrada /Salida de flujo en C++
- b. Capítulo VII - Procesamiento de Archivos
- c. Capítulo VIII - Manejo de excepciones

### Objetivo

Resolver un problema, con un programa en el lenguaje de C++ aplicando lo aprendido en los temas de estudio

### Software de Desarrollo

CodeBlocks, en la plataforma MOODLE está disponible las instrucciones para su instalación

### Desarrollo

Se requiere un desarrollo para manejar inventario de algunos productos de una pasamanería. Debe llevar el inventario y pedidos de los productos del inventario.

La aplicación inicialmente contará con un menú de la siguiente forma

#### Software para Pasamanería

1. Inventario productos
2. Generar pedido

3. Enviar Pedido
4. Reporte Inventario
5. Reporte Pedidos Pendientes
6. Salir del Programa

Importante: El menú debe repetirse siempre hasta que se ingrese el valor de salida #5

### **Inventario productos**

Debe tener la opción de ingresar producto, consultar producto por código o bien eliminar un producto por código. No se pueden repetir códigos de producto. En particular, antes de eliminar debe confirmar si está seguro de eliminar el producto.

#### **Inventario Productos**

1. Ingresar producto
2. Consultar producto (Por código)
3. Eliminar producto
4. Volver al menú principal

Cada producto cuenta con: código de cuatro dígitos, nombre y cantidad disponible. En la opción de consultar solamente se puede presentar, si existe el código, el nombre y la cantidad de producto disponible.

### **Generar pedido**

En esta opción debe pedir, el código de pedido o bien generarlo. También pedirá un nombre para el pedido. Posteriormente, solicita el código de los productos a agregar y la cantidad de cada producto. Esto se puede repetir hasta que el usuario no requiera agregar productos al pedido. Debe presentar en pantalla los productos que va agregando al pedido. Debe validar que no exceda la cantidad de producto disponible en el inventario y adicionalmente, una vez se confirme el pedido debe actualizar el inventario, con la cantidad disponible luego del pedido.

### **Enviar pedido**

En esta opción debe pedir el código de un pedido, verificar si tiene el indicador de enviado en N, en ese caso consulta si desea enviar el pedido. En caso de respuesta afirmativa actualiza todos los registros correspondientes a ese pedido en el campo de Enviado a S.

### **Reporte Inventario**

En esta opción debe presentar el inventario que existe:

Código producto	Nombre	Cantidad Disponible
0101	Hilo Poliester Amarillo	40
0102	Hilo Poliester Azul	40

### Reporte Pedidos

En esta opción debe presentar el reporte de pedidos disponibles pendientes de envío:

Código pedido	Nombre Pedido	Código Producto	Cantidad	Enviado (S/N)
0001	Pedido Doña María	0101	1	N
0001	Pedido Doña María	0101	1	N
0002	Pedido Doña Ana	0101	1	N

### Codificación:

1. En el menú deben aparecer solo las opciones indicadas, no debe omitir ninguna ni agregar opciones. En caso de que el usuario digite un valor no existente debe mostrar el mensaje de : **“Opción inválida, vuelva a intentarlo”**
2. Debe generar archivos de texto para almacenar la información. Productos.txt y Pedidos.txt.
3. El archivo de producto contiene: Código de producto (generado o bien ingresado) debe ser único, Nombre del producto y cantidad disponible.
4. El archivo de pedidos contiene: Código de pedido (generado o bien ingresado) debe ser único, Nombre del pedido, código de producto y cantidad solicitada (debe verificar si hay inventario antes) e indicador si el pedido se envió o no.
5. Se debe controlar la lectura de los archivos en caso de no encontrarlo o no poder leer la información debe presentar un mensaje. Utilice excepción para esta validación (TRY-CATCH-THROW).
6. Se debe controlar que tanto el código de producto, como el código de pedido no se repita, en caso de estar existente debe pedir uno nuevo. Utilice excepción para esta validación (TRY-CATCH-THROW).
7. Se debe desarrollar funciones para cada opción sobre los productos: Ingresar, Consultar y Eliminar.
8. Se debe desarrollar funciones para manejo de pedido: Generar pedido y Enviar pedido.
9. Se debe desarrollar una función para cada opción de reporte. Reporte de inventario y Reporte de pedidos pendientes.
10. El menú debe repetirse siempre hasta que se ingrese el valor de salida #5. Es decir, la única forma de que el programa termine debe ser que el jugador vuelve al menú desde donde este y presiona la opción #6 para Salir.

11. Debe controlar el ingreso de valores sea correcto para productos e inventario. Y presentar los mensajes correspondientes en caso de que se ingresen valores o respuestas erróneas.
12. Para las opciones de consulta donde solicita código, en caso de no encontrarlo, debe mostrar el mensaje que lo indique; y dar la opción de seguir consultando, mediante el mensaje: Desea continuar o no (S/N). En caso de no continuar, volverá al menú principal. Debe validar que solo permita el uso de mayúsculas para 'S' o 'N'

## Honestidad Académica



<https://audiovisuales.uned.ac.cr/play/player/23048>

Nota Importante

Cada estudiante es responsable del contenido que entrega, si no es el archivo correcto, no podrá entregarlo posterior a la fecha establecida.

Si el contenido del archivo coincide con algún otro estudiante, o se comprueba que no es de su autoría, se aplicaría lo indicado en la plataforma en el documento [Lineamientos ante casos de plagio](#)

### Indicaciones Importantes

- Es obligatorio que incluya todo el directorio donde se encuentra < nombre del instrumento>.
- La < nombre del instrumento > debe estar desarrollado en [IDE de desarrollo] que es la herramienta oficial del curso.
- El programa debe ser modular, utilizando de la mejor manera funciones definidas por usted.
- Los trabajos deben realizarse en forma individual. Dentro del código del programa debe de indicar la documentación que explique cómo fue realizado el programa.
- Si utiliza código de algún ejemplo del libro, o de otra fuente que no sea de su autoría, debe de indicarlo.
- Comprima todos los archivos en un solo archivo .zip o .rar.
- **Nombre del archivo que envía:** debe ser nombre y primer apellido del estudiante, y nombre de la tarea. **Ejemplo: JuanRojas-tarea1.**
- La entrega de la <Nombre del instrumento> en las fechas establecidas en la plataforma de aprendizaje en línea Moodle en el apartado que se indique.



Si no concluyó a tiempo la tarea, debe entregar lo que pudo hacer e incluir una carta explicando las razones por las cuales no finalizó.

### Rúbrica de Evaluación

Criterio	Cumple a satisfacción lo indicado en la evaluación	Cumple medianamente en lo indicado en la evaluación	Cumple en contenido y formato pero los aportes no son significantes	No cumple o no presenta lo solicitado
<b>Formato:</b> Nitidez y presentación del código, incluyendo Redacción - Ortografía // Documentación interna dentro del código	5	3	1	0
<b>Orden y claridad en el planteamiento</b> (lógica). Cómo ordena las ideas para lograr la mejor solución, aplicando correctamente los conocimientos y herramientas vistos hasta el momento en el curso. Logra completar la funcionalidad solicitada en el enunciado.	5	3	1	0
<b>Estructuras de control – Secuenciales.</b> Utiliza <i>if</i> , <i>if/else</i> y <i>switch</i> en la solución de forma adecuada. Es evalúa que su uso en la lógica sea correcto. Ejemplo: Menús, Preguntas, validaciones de datos, evaluación de datos, etc	15	10	5	0.1
<b>Estructuras de control - iterativas.</b> Utiliza <i>while</i> , <i>do/while</i> y <i>for</i> en la solución de forma adecuada. Se evalúa si alguna estructura se encicla por la lógica errónea utilizada. Ejemplo: menú, validaciones, presentación de información, recorrido archivo (lectura y escritura) , etc.	15	10	5	0.1
<b>Manejo de Excepciones.</b> Utiliza las necesarias y suficientes adicionales a la o las	5	3	1	0.1

solicitadas en el enunciado. Con un correcto nombre, parámetro (si aplica) y llamado de las mismas				
<b>Funciones Solicitadas.</b> Creación y llamado correcto de las funciones según se solicita en el enunciado del problema.	20	15	10	0.1
<b>Funciones Adicionales.</b> Creación y llamado de al menos 2 funciones más para estructurar la lógica de la solución.	10	5	3	0.1
<b>Archivos.</b> Utiliza manejo de archivos en su solución, en particular para la captura y control de los productos y pedidos. Correcta creación y actualización.	5	3	1	0
<b>Archivos.</b> Utiliza manejo de archivos para el despliegue de información (búsquedas y reporte general). Correcta lectura y presentación.	5	3	1	0
<b>Impresión de información en pantalla</b> (Calidad-validez datos/presentación tabulada). Uso correcto de entrada y salida de datos por pantalla. Solicitud de información, validaciones, presentación de lo mínimo solicitado	10	5	2	0
<b>Interfaz de usuario en general.</b> (NO GUI) - aplicación fácil usar e intuitiva. Se refiere a evaluar la distribución y uso de la pantalla, menús y dinámica de uso de la solución. Incluyendo lo mínimo según el ejemplo del enunciado o lo adicional que el estudiante entienda necesario.	5	3	1	0
<b>TOTAL</b>	<b>100</b>			