



Cátedra Tecnología de Sistemas

[Introducción a la programación]

Código: [00831]

Proyecto. Valor 3%

Temas de Estudio

Tema 1 y Tema 2, vistos anteriormente

Tema 3

Subtemas

1. Funciones y una introducción a la recursividad
2. Plantillas de clase array y vector; captura de excepciones

Tema 4

Subtemas

1. Apuntadores
2. Entrada/salida de flujos: un análisis detallado
3. Procesamiento de archivos

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 debe de desarrollar un programa para las Asociaciones Administradoras de los Sistemas de Acueductos y Alcantarillados (ASADAS), que maneje **ingreso de las piezas de repuesto en inventario y pedidos de piezas para arreglos por parte del personal**. El programa debe contar con un menú con las siguientes opciones:

1. Inventario
 - a. Ingresar pieza al inventario.
 - b. Consultar pieza del inventario
 - c. Modificar pieza del inventario.
 - d. Eliminar pieza del inventario

2. Pedido
 - a. Ingreso de pedido
 - b. Consulta de Pedido
 - c. Cancelar Pedido
 - d. Registrar entrega de pedido
3. Reporte de Inventario
4. Reporte de Pedidos
6. Salir.

Inventario

Se debe ingresar registro de las piezas con la siguiente información:

1. Código de pieza, cuyo formato sería P y un consecutivo de no más de 8 dígitos, por ejemplo, P00000890
2. Nombre de la pieza.
3. Costo de la pieza por unidad
4. Cantidad de piezas
5. Nombre de la Empresa que provee las piezas.
6. Factura de la empresa que provee las piezas
7. Número de cédula de quien ingresa las piezas al inventario.

La información se debe registrar en un archivo llamado INVENTARIO.TXT.

Consulta por número de pedido y presenta todos los datos

Modificar primero consulta por número de pedido y solamente permite cambiar: Costo y Cantidad.

Eliminar primero consulta por número de pedido, y luego permite eliminar del inventario. Debe presentar la información y preguntar si está seguro de eliminar esa pieza.

Pedido

Se debe ingresar registro de las piezas con la siguiente información:

1. Código de pedido, cuyo formato sería PE y un consecutivo de no más de 8 dígitos, por ejemplo, PE00000890
2. Código de pieza
3. Cantidad de piezas a solicitar
4. Número de identificación del empleado
5. Fecha solicitud, en formato DD/MM/YYYY
6. Estado del pedido, Cancelado, Entregado o En proceso.

La información se debe registrar en un archivo llamado PEDIDOS.TXT.

Ingreso de pedido, el estado debe grabarse como "En proceso"

Consulta primero por número de pedido y luego presenta todos los datos

Cancelar primero consulta por número de pedido y luego solamente permite cambiar: Estado del pedido a Cancelado.

Registrar primero consulta por número de pedido y luego actualiza entrega de pedido: Estado del pedido a Entregado.

Reporte de Inventario

Se debe generar un reporte con el inventario con la siguiente información:

Reporte de Inventario						
Código Pieza	Nombre Pieza	Costo	Cantidad	Nombre de la Empresa	Factura	# Cédula quien ingresa pieza
....
¿Desea volver al menú principal? S/N: _						

Reporte de Pedidos

Se debe generar un reporte de pedidos, con la siguiente información:

Reporte de Pedidos					
Código Pedido	Código Pieza	Cantidad	# Cédula quien solicita pieza(s)	Fecha Solicitud	Estado
....	
¿Desea volver al menú principal? S/N: _					

Requerimientos técnicos para el programador:

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. La información debe almacenarse en un medio permanente para que si apaga el equipo pueda levantar el aplicativo luego y consultar la información almacenada nuevamente. Por lo tanto, se debe guardar toda la información según sea el archivo de texto indicado.
3. Se debe incluir funciones y métodos en la programación.
4. Se debe incluir todas las validaciones requeridas:
 - a. Números sin letras. Cédula de 9 dígitos, Cantidad y Costo
 - b. Campos vacíos. En las opciones de solicitud de información no se puede ingresar valores en blanco.
 - c. Opciones S y N validando mayúscula y minúscula. Respuestas a preguntas de Si o No (S/N). Si ingresa algo diferente no debe accionar.
 - d. Entrada de menú opciones controladas.
 - e. Valores preexistentes y registrados de Clientes y Doctores previo a generar la cita
 - f. Código de pieza, cuyo formato sería P y un consecutivo de no más de 8 dígitos, por ejemplo, P00000890, si el valor no está correcto debe desplegar un mensaje que indique

- “Valor de código incorrecto” . Existe la opción de generarlo automáticamente y no pedirlo al usuario.
- g. Con respecto al Código de pedido, cuyo formato sería PE y un consecutivo de no más de 8 dígitos, por ejemplo, PE00000890, si el valor no está correcto debe desplegar un mensaje que indique “Valor del código incorrecto”. Existe la opción de generarlo automáticamente y no pedirlo al usuario.
5. Tomar en consideración presentación en pantalla:
- Menú posicionado en el mismo lugar.
 - No despliegue de caracteres especiales en la interfase de pantalla.
 - Correcta visualización y posicionamiento de los campos.
6. Documentación básica del proyecto:
- El inicio del programa indicar: Fecha, autor del programa (nombre del estudiante)
 - Comentar variables, procedimientos y funciones.
 - Agregar cualquier otro comentario pertinente.
7. Debe crear dos matrices con la información de Inventario y de Pedidos que leerá de los archivos indicados, cuando la aplicación sea ejecutada inicialmente. Y deberá actualizar los archivos TXT con la información de las matrices, al indicar la **opción #6 del menú** para Salir. Esto le permite manipular la información en las diferentes opciones durante la ejecución del programa. NOTA: si considera que requiere otra estructura matriz o vector para manejar la información la puede crear. Lo indicado es lo mínimo que se pide para cumplir.
8. Debe crear mínimo funciones para la menos las siguientes funcionalidades: **Puede crear mas según su necesidad
- Ingresar pieza al inventario.
 - Consultar pieza del inventario
 - Modificar pieza del inventario.
 - Eliminar pieza del inventario
 - Ingreso de pedido
 - Consulta de Pedido
 - Cancelar Pedido
 - Registrar entrega de pedido
 - Reporte de Inventario
 - Reporte de Pedidos
 - Carga de información de los Archivos a Matriz
 - Descarga de información de Matriz a Archivos
9. El menú debe repetirse **siempre** hasta que se ingrese el valor de salida #6. Es decir, la **única forma de que el programa termine** debe ser que el usuario vuelve al menú desde donde este y presiona la **opción #6 del menú** para Salir. En cada pantalla donde ingrese del menú deberá tener al final la pregunta: *¿Desea volver al menú principal? S/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.

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
Formato: Nitidez y presentación del código, incluyendo Redacción-Ortografía // Documentación interna dentro del código	3	2	1	0.1
Orden y claridad en el planteamiento (lógica). Cómo ordena las ideas para lograr la mejor solución, aplicando correctamente los conocimientos y herramientas vistos hasta el momento en el curso.	3	2	1	0.1
Estructuras de control – Secuenciales. Utiliza <i>if</i> , <i>if/else</i> y <i>switch</i> en la solución de forma adecuada. Ejemplo: Menús, Preguntas, <u>validaciones de datos</u> , evaluación de ganador, etc.	10	5	2	0.1
Estructuras de control - iterativas. Utiliza <i>while</i> , <i>do/while</i> y <i>for</i> en la solución de forma adecuada. Ejemplo: <u>recorrido de arreglos y/o datos de archivos</u> , validaciones, repetición de menú, etc.	10	5	2	0.1
Funciones General. Utiliza las necesarias y suficientes adicionales a la o las solicitadas en el enunciado. Con un correcto nombre, parámetro (si aplica) y llamado de las mismas.	20	15	5	0.1
Archivos. Utiliza manejo de archivos en su solución, en particular para llevar control del <i>Inventario</i>	15	8	3	0.1
Archivos. Utiliza manejo de archivos en su solución, en particular para llevar control de los <i>Pedidos</i>	15	8	3	0.1

Arreglos. Correcta declaración, recorrido y uso en general de los arreglos solicitados o propios, en la solución planteada. Tomando en consideración cualquier solicitud específica en el enunciado. Incluye manejo de matrices solicitadas para <i>Pedidos e Inventario</i> .	15	8	3	0.1
Manejo de Excepciones. Utiliza las necesarias y suficientes adicionales a la o las solicitadas en el enunciado. Con un correcto nombre, parámetro (si aplica) y llamado de las mismas. Estructuras Try, Catch y/o Throw	3	2	1	0.1
Impresión de información en pantalla (Calidad-validez datos/presentación tabulada). Uso correcto de entrada y salida de datos por pantalla. Solicitud de información, validaciones, presentación de reportes solicitados.	3	2	1	0.1
Interfaz de usuario en general. (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.	3	2	1	0.1
TOTAL	100			