



Cátedra Tecnología de Sistemas

Introducción a la Programación

Código: **00831**

Proyecto 2. Valor [2] %

Temas de Estudio

1. Tema 3
 - a. Funciones y una introducción a la recursividad
 - b. Plantillas de clase array y vector, captura de excepciones.
2. Tema 4
 - a. Apuntadores
 - b. Entrada/Salida de flujos un análisis
 - c. Procesamiento de archivos

Objetivo

Se debe crear una interfase para, para ingresar recetas en una farmacia. Esta farmacia tiene un catálogo de productos y un catálogo de asociaciones de productos con claves de indicación.

Software de Desarrollo

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

Desarrollo

Se debe de desarrollar un programa que ingrese una receta y la imprima en pantalla. Debe contar con un catálogo de 10 productos y un archivo de asociaciones de palabras claves relacionadas con el producto. Y al final se imprime una receta en pantalla.

El programa debe contar con un menú con las siguientes opciones:

1. Ingresar productos.
2. Ingresar asociaciones con productos.
3. Registrar una receta.
4. Reporte de catálogo de productos
5. Reporte impresión de la receta
6. Salir.

Ingresar productos.

Se debe ingresar un catálogo de medicamentos de farmacia, catálogo debe ser de la siguiente manera.

- Código (puede ser un número nada más, pero debe ser numérico, ejemplo 1,2,3...).
- Nombre (un nombre único, por ejemplo, acetaminofén, aspirina) Cantidad.
- Esta información se debe registrar en un archivo llamado MEDICAMENTOS.TXT.
- Para cada registro se debe preguntar al final "INGRESAR EL MEDICAMENTO (S/N)"
 - En caso de que la respuesta sea S, entonces se guarda la información.
 - En caso de que sea N se devuelve al menú.

Ingresar asociaciones con productos.

El catálogo de asociaciones debe ser de la siguiente manera.

- Se debe ingresar el código del producto y luego desplegar el producto.
- Se debe ingresar la palabra asociativa única.
- Se debe guardar en el archivo de asociaciones llamado CLAVES.TXT.
- Para cada registro se debe preguntar al final "INGRESAR LA ASOCIACION (S/N)"
 - En caso de que la respuesta sea S, entonces se guarda la información.
 - En caso de que sea N se devuelve al menú.

Ejemplo de asociación.

Código, asociación. 1, TABLETAS. Donde 1 es el código del producto y la palabra asociativa es TABLETAS.

Es importante que se demuestre que se está registrando la información. Debería al menos registrar 5 asociaciones.

Registrar una receta.

Se debe registrar una receta por producto, al menos deben existir tres recetas para desplegarlas en el reporte.

La receta debe tener la siguiente estructura.

- Número (debe ser un número a partir de 1000, y para cada receta se debe validar que no se repita)
- Paciente, se debe ingresar el nombre en un campo de texto.
- Código del producto, se ingresa el código y se despliega el nombre del producto.
- Indicación se debe ingresar en un campo de texto.
- Cantidad debe ser un valor con dos decimales.

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

El proceso es el siguiente:

- Se debe validar que la receta que se va a ingresar no exista, en caso de que suceda debe desplegar el mensaje “RECETA YA EXISTE, SALIR DE REGISTRO DE RECETA (S/N)”.
 - Si presiona S vuelve al menú.
 - Si presiona N limpia el campo y vuelve a pedir el número de receta.
- Se ingresa el nombre.
- Se ingresa el código y se debe desplegar el nombre del medicamento.
- Se ingresa la indicación.
- Se ingresa la cantidad. (se debe contemplar, el rebajo de la cantidad de producto de la existencia de cada medicamento, así como no permitir registrar la receta si la cantidad es mayor a la existencia actual).
- Se despliega un mensaje “REGISTRAR RECETA (S/N).”
 - Si presiona S, se debe invocar a un procedimiento que valide la asociación del producto con la indicación. Esta validación se va a explicar en el siguiente punto.
 - Si presiona N, se devuelve al menú.

Validación de Asociación.

La validación consiste en localizar el código del producto en el archivo CLAVES.TXT y extraer la palabra asociada. Esta palabra se debe buscar en la indicación, si existe en la indicación entonces está correcto, si no existe se debe desplegar un mensaje indicando “LA INDICACION NO ES CORRECTA” y no se permite guardar, se debe quedar en la pantalla para modificar la indicación.

Ejemplo.

Si la palabra asociada para ACETAMINOFEN es TOMAR, y se digita una receta por ACETAMINOFEN con una indicación que dice “APLICAR UN SUPOSITORIO CADA 3 HORAS “, entonces debes buscar esta palabra (TOMAR) dentro de la indicación y si no la encuentra es porque la indicación no es correcta. Se va a proporcionar una lista con productos y palabras asociadas como referencia. Se debe demostrar que cada producto se guarda en el archivo desde el programa.

Producto	Asociación
ACETAMINOFEN	TABLETA
PAYDOS	APLICAR
PENICILINA	APLICAR
DICLOFENAC	INYECTAR
TERRAMICINA	APLICAR
AMOXICILINA	CAPSULA
DEXAMENTASONA	INYECTAR

Reporte de catálogo de productos.

Lista de productos		
Código	Descripción	Cantidad
1	Clinoril	10

Reporte impresión de la receta.

Receta No.	1000	
Paciente		
Rita Obando Mora		
Medicamento		
1	ACETAMINOFEN	
TOMAR UNA DOS VECES AL DIA		
Cantidad	10	

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. Se debe validar que los códigos de producto y número de receta sean estrictamente numéricos.
3. Se sugiere que toda la información de la indicación y de las asociaciones, así como la descripción del medicamento se desplieguen y conviertan a mayúscula.
4. Se debe guardar toda la información según sea el archivo de texto indicado.
5. Se debe incluir funciones y métodos en la programación.
6. Se debe incluir todas las validaciones requeridas:
 - a. Números sin letras.
 - b. Campos vacíos.
 - c. Opciones S y N validando mayúscula y minúscula.
 - d. Entrada de menú opciones controladas.
7. Tomar en consideración presentación en pantalla:
 - a. Menú posicionado en el mismo lugar.
 - b. No despliegue de caracteres especiales en la interfase de pantalla.
 - c. Correcta visualización y posicionamiento de los campos.
8. Documentación básica del proyecto:
 - a. El inicio del programa indicar: Fecha, autor del programa (nombre del estudiante)
 - b. Comentar variables, procedimientos y funciones.
 - c. Agregar cualquier otro comentario pertinente.

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
Formato: Nitidez y presentación del código, incluyendo Redacción - Ortografía // Documentación interna dentro del código	5	3	2	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. Logra completar la funcionalidad solicitada en el enunciado.	5	3	2	0.1
Estructuras de control – Secuenciales. 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, identifica correctamente vocales, etc. .	5	3	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. Se evalúa si alguna estructura se encicla por la lógica errónea utilizada. Ejemplo: recorrido de números, validaciones, matrices, etc.	10	6	3	0.1

Funciones General. Utiliza las necesarias y suficientes adicionales a la o las solicitudes en el enunciado. Con un correcto nombre, parámetro (si aplica) y llamado de las mismas.	10	6	3	0.1
Archivos. Utiliza manejo de archivos en su solución. Para registro de medicamentos, claves.	20	15	5	0.1
Archivos. Utiliza manejo de archivos en su solución. Para registro recetas.	20	15	5	0.1
Manejo de Excepciones. Utiliza las necesarias y suficientes adicionales a la o las solicitudes en el enunciado. Con un correcto nombre, parámetro (si aplica) y llamado de las mismas	10	6	3	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 lo mínimo solicitado	5	3	2	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.	10	6	3	0.1
TOTAL	100			