



UNIVERSIDAD ESTATAL A DISTANCIA
VICERRECTORÍA ACADÉMICA
ESCUELA DE CIENCIAS EXACTAS Y NATURALES



Cátedra Tecnología de Sistemas

Proyecto# 1. Valor 1%

Temas de Estudio

Tema 1: Fundamentos de C++

Subtemas

1. Introducción a las computadoras y a C++

- Historia y evolución de las computadoras.
- Conceptos básicos de hardware y software.
- Introducción al lenguaje de programación C++.

2. Introducción a la programación en C++, entradas/salidas y operadores

- Estructura básica de un programa en C++.
- Uso de cin y cout para entradas y salidas.
- Operadores aritméticos, relacionales y lógicos.

3. Introducción funciones miembro y cadenas

- Definición y uso de funciones.
- Manejo de cadenas de caracteres (string).

Tema 2: Algoritmos e instrucciones de control

Subtemas

1. Desarrollo de algoritmos e instrucciones de control: Parte 1

- Concepto de algoritmo.
- Operadores matemáticos y relacionales.
- Estructuras de control: if, else, switch.

2. Estructuras de control; Parte 2: operadores lógicos

- Bucles: for(), while(), do-while().
- Operadores lógicos y su uso en condiciones.

Objetivo

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

Software de Desarrollo

Las instrucciones para la instalación de CodeBlocks, están disponibles en el campus virtual AprendeU.

Desarrollo

Desarrollar un programa en C++ que implemente un sistema de gestión de tareas. El sistema debe incluir un menú con las siguientes opciones: agregar, eliminar, mostrar y buscar tareas. Además, debe validar las entradas del usuario y utilizar estructuras de repetición, selección y condicionales.

Funcionalidades. El programa contará con un menú con las siguientes opciones:

<p style="text-align: center;">Menú Principal</p> <ol style="list-style-type: none">1. Agregar Tareas2. Buscar Tarea3. Mostrar todas las tareas4. Eliminar Tarea5. Salir del Programa

1. Agregar Tareas

- El usuario puede agregar entre 1 y 5 tareas como máximo.
- Se debe solicitar la siguiente información para cada tarea:
 1. Código de la tarea (valor numérico de 4 dígitos)
 2. Nombre de la tarea (texto descriptivo)
 3. Fecha de vencimiento en formato dd/mm/aaaa
 4. Hora de inicio en formato hh:mm
 5. Hora de finalización en formato hh:mm
- Es obligatorio que todos los campos estén completos; no se aceptarán valores vacíos.
- Se deben validar los datos de fecha y hora, en el formato que se describe, y mostrar un mensaje de error en caso de no cumplir con dicho formato.

- Una vez registrada la tarea correctamente, se debe mostrar el mensaje: “Tarea agregada exitosamente”.
- Mostrar un mensaje para regresar al menú principal, en caso de ser Si (S), vuelve al menú: en caso de ser No (N), puede iniciar nuevamente el proceso para agregar tareas.
- La única forma de salir del programa es a través de la opción “Salir del Programa”

```

-----
Ingrese la cantidad de tareas que desea agregar:
-----
Número de código tarea 1 (4 dígitos):_____
Nombre de la tarea : _____
Fecha de vencimiento (dd/mm/aaaa):_____
Hora de inicio (hh:mm)      : _____
Hora de finalización (hh:mm) : _____
    
```

Desea regresar al menú principal (S/N):_____

2. Buscar Tarea:

- Solicitar al usuario el código de la tarea que desea buscar.
- Verificar si la tarea existe en el registro.
- Si la tarea es encontrada, mostrar la siguiente información:
 1. Nombre de la tarea,
 2. Fecha de vencimiento (dd/mm/aaaa)
 3. Duración de la tarea en minutos. (Debe realizar el cálculo de forma automática, considerando la hora inicio y la hora fin)
- Si la tarea no se encuentra, mostrar el mensaje: “Tarea no encontrada”

Ejemplo de entrada y salida:

Ingrese el código de la tarea a buscar: 1234

+-----+-----+-----+-----+			
Código	Nombre	Fecha Vencimiento	Minutos invertidos
+-----+-----+-----+-----+			
1234	Clasificar material	20/02/2025	75
+-----+-----+-----+-----+			

Opción 3: Mostrar Todas las Tareas

- **Regla:** Se deben listar todas las tareas almacenadas en el sistema.
- **Validación:** Antes de mostrar las tareas, verificar si hay registros disponibles.

- Si no hay tareas registradas, mostrar el mensaje “No hay tareas registradas” y vuelve al menú principal.
- Si existen tareas almacenadas, mostrar la información en el siguiente formato:

Lista de todas las tareas:

+-----+-----+-----+-----+			
Código	Nombre	Fecha Vencimiento	Minutos invertidos
+-----+-----+-----+-----+			
1234	Clasificar material	20/02/2025	35
6543	Elaborar ensayo	21/02/2025	148
5555	Entregar ensayo	22/02/2025	10
+-----+-----+-----+-----+			

4. Eliminar Tarea

- Solicitar al usuario el código de la tarea que desea eliminar.
- Verificar si la tarea existe en el sistema antes de eliminarla.
- Si la tarea existe, solicitar confirmación con el siguiente mensaje: “¿Está seguro de que desea eliminar la tarea [Nombre de la tarea]?,(S/N).”
 - Si el usuario ingresa “S”, eliminar la tarea y mostrar el mensaje: “Tarea eliminada exitosamente”
 - Si el usuario ingresa “N”, cancelar la acción y regresar al menú principal.

5. Salir del Programa

- Esta opción finaliza la ejecución del sistema.

El sistema debe utilizar ciclos (estructuras de control o bucles) y variables (enteras) para el manejo de los datos, adicionalmente, puede utilizar constantes para los porcentajes y rangos, así como estructuras de datos.

Codificación:

- El menú debe incluir únicamente las opciones indicadas **(1-2-3-4-5)**, sin **omitir ni agregar opciones adicionales. Además, debe validar que solo permita ingresar valores numéricos dentro de este rango, evitando el uso de** letras u otros caracteres especiales.
- El programa debe ser compatible con caracteres especiales, permitiendo mostrar las letras con tildes y los caracteres como la “ñ”.
- Si el usuario digite un valor fuera del rango del menú, se debe mostrar el mensaje de: **“Opción inválida, vuelva a intentarlo”** y solicitar nuevamente una opción válida.
- En la opción “Agregar Tareas”, todos los campos son obligatorios. En caso de que no ingrese un valor, debe de programar lo necesario para que el campo sea requerido, **todos los campos son requeridos**.
- Si no ha ingresado datos previamente, cuando ingrese a las opciones 2-3-4, debe desplegar un mensaje que indique **“Debe ingresar primero la tarea”** y devolverlo al menú principal.
- Deben existir validaciones con los datos que se ingresan:
 - a. El menú solo debe permitir opciones numéricas (1 a 5). Si se ingresa un valor inválido, debe mostrar **“Opción inválida, vuelva a intentarlo”**.
 - b. Al registrar tareas, si un campo está vacío, debe solicitarse nuevamente.
 - c. La cantidad de tareas debe estar en el rango de 1 a 5. Si el valor está fuera de este rango, mostrar **“Cantidad de tareas fuera de rango”**.
 - d. El número de código debe tener exactamente 4 dígitos y ser numérico.
 - e. El nombre de la tarea debe permitir varias palabras **“cadena de caracteres”**.
 - f. Las opciones que despliegue aceptar "S" o "N" (convertir minúsculas a mayúsculas automáticamente).
- El menú debe mostrarse de forma continua hasta que el usuario ingrese el valor de salida #5. Es decir, la única forma de finalizar el programa es regresando al menú principal y eligiendo la opción #5 para Salir.

Honestidad Académica



[https://audiovisuales.uned.a
c.cr/play/player/23048](https://audiovisuales.uned.a.c.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 **< Proyecto 1>**.
- El **< Proyecto 1 >** debe estar desarrollado en **[CodeBlocks]** que es la herramienta oficial del curso.
- El programa debe ser modular, utilizando de la mejor manera las 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; **incluir en el encabezado el nombre del autor, fecha y una descripción breve del programa.**
- Si utiliza código de algún ejemplo del libro, o de otra fuente que no sea de su autoría, debe de indicarlo en las referencias que debe anotar en la documentación.
- Comprima todos los archivos en un solo archivo .zip o .rar. Asegúrese que el archivo no éste dañado, de lo contrario no se calificará el proyecto y pierde los puntos y el derecho a la defensa.
- **Nombre del archivo que envía:** debe ser nombre y primer apellido del estudiante, y nombre del proyecto. **Ejemplo: JuanRojas-Proyecto1.**
- La entrega de la **<Proyecto 1>** debe realizarla 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	5	3	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. Se evalúa correcta indentación del código, correcto uso de nombres significativos para las variables y estructura general de la lógica presentada.	10	5	3	0.1
Estructuras de control – Secuenciales. Utiliza <i>if</i> , <i>if/else</i> y <i>switch</i> en la solución de forma adecuada. Se evalúa que su uso en la lógica sea correcto, en los siguientes puntos del enunciado: <ul style="list-style-type: none"> Opciones de menú evaluadas correctamente (uso de <i>switch</i>). Cantidad de tareas a agregar 	25	15	10	0.1
Estructuras de control - iterativas. Utiliza <i>while</i> , <ul style="list-style-type: none"> <i>do/while</i> y <i>for</i> en la 	25	15	10	0.1

<p>solución de forma adecuada. En este punto se evalúa: Si alguna estructura iterativa se encicla por la lógica errónea utilizada.</p> <ul style="list-style-type: none"> • Uso correcto de estructura iterativa para manejo del menú. • Uso correcto de estructura iterativa para preguntas S/N e ingreso de datos. 				
<p>Validaciones. Utiliza las necesarias y suficientes adicionales a la o las solicitadas en el enunciado. En este punto se evalúa el completo desarrollo y funcionalidad de las validaciones solicitadas en el enunciado:</p> <ul style="list-style-type: none"> • código • nombre de tarea • Formato de fecha • Respuestas a preguntas de Si o No (S/N). • Opciones de menú diferente a 1-5 • Cualquier otr a requerida. 	20	10	5	0.1
<p>Impresión de información en pantalla (Calidad-validez datos/presentación tabulada). Presentación adecuada de entrada y salida de datos por pantalla. Solicitud de información, validaciones, presentación de lo mínimo solicitado. En específico en este punto también se evalúa toda la información que se indica en el enunciado, completa y correcta.</p>	10	5	2	0.1
<p>Interfaz de usuario en general. (NO GUI) - aplicación fácil usar e intuitiva (<i>mensajes</i>). Se refiere a</p>	5	3	1	0.1

evaluar la distribución general 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.				
TOTAL	100			