



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



Cátedra Tecnología de Sistemas

Proyecto # 2. Valor 2%

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

El propósito de este proyecto es que el estudiante desarrolle un programa en C++, poniendo en práctica los conceptos aprendidos en los temas de estudio. A través de esta experiencia, fortalecerá su comprensión y habilidades en el uso de estructuras de control, funciones, arreglos, manipulación de archivos y manejo de excepciones aplicándolos en un contexto práctico y funcional.

Software de Desarrollo

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

Desarrollo

Descripción del Problema

Se requiere desarrollar un sistema en C++ que permita gestionar las calificaciones de los estudiantes en diferentes materias. Este sistema deberá contar con un menú de opciones, donde el usuario pueda realizar las siguientes operaciones:

1. Registrar Estudiante (almacenando datos como nombre, ID, edad, y otros datos relevantes.).
2. Ingresar Calificaciones en distintas materias (mínimo 3 materias).
3. Modificar Datos Estudiante. Solo se permite la modificación de Edad y lugar de residencia.
4. Modificar Registro de Notas por Estudiante. El programa debe permitir la modificación únicamente de las notas registradas, no se debe modificar el nombre del estudiante, ni tampoco el ID y Edad.
5. Eliminar Registro de Estudiante. El programa debe solicitar el número de identificación, si lo encuentra, debe preguntar si esta seguro que desea eliminar el registro (S/N). Si es Si (S), se debe eliminar todo el registro tanto del archivo estudiantes.txt, así como del archivo notas.txt. En caso de ser N, debe preguntar si desea eliminar otro estudiante y vuelve el ciclo.
6. Reporte de Estudiantes, promedios y estado.
7. Salir del programa.

Registrar Estudiante

Para registrar un estudiante en el sistema, se deben solicitar y validar los siguientes datos:

- a. Identificación del Estudiante, tamaño de 10 caracteres (0101110111), no debe permitir menos o más de esa cantidad de caracteres y asegurarse que no se repita y que no permita letras.
- b. Nombre completo del Estudiante (nombre y dos apellidos)
- c. Lugar de Residencia: Provincia, cantón y Distrito
- d. Edad. Definir el rango entre 18 y 100.
- e. Genero (debe seleccionar entre Femenino y Masculino) y cualquier otro que considere necesario, máximo 3 géneros.
- f. Al finalizar debe mostrar un mensaje que indique que el registro se guardó satisfactoriamente.

Ejemplo:

```
-----
|          REGISTRO DE NUEVO ESTUDIANTE          |
|-----|
Ingrese la identificación del estudiante (10 dígitos): 0101110111
☒ Identificación válida.

Ingrese el nombre completo (Nombre y dos apellidos): Juan Pérez Rodríguez

Ingrese el lugar de residencia:
- Provincia: San José
- Cantón: Central
- Distrito: Catedral

Ingrese la edad: 20

Seleccione el género:
[1] Masculino
[2] Femenino
[3] Otro
Opción: 1
☒ Género seleccionado: Masculino

☒ Estudiante registrado con éxito en "ESTUDIANTES.txt".
```

Ingresar Calificaciones: Para registrar las calificaciones de un estudiante, el sistema deber seguir lo siguientes pasos y validaciones:

- Se debe solicitar la Identificación del estudiante y buscarla en el archivo ESTUDIANTE.txt.
- Si la identificación no existe, debe mostrar un mensaje que indique ***“Estudiante no registrado”*** y preguntar si desea ingresar otra identificación, en caso de ser Si ***“S”***, vuelve a solicitar la identificación, en caso de ser No ***“N”***, vuelve al menú principal. Debe convertir minúsculas a mayúsculas.
- Materia que cursa, se debe registrar la materia en la que el estudiante este inscrito. No se permite dejar este campo vacío.
- Debe registrar las notas en base 10 para cada uno de los siguientes ITEMS:

Proyecto 1, Proyecto 2, Ensayo, Foro, Defensa.

- Obtener el promedio y agregar un campo que indique si aprobó la materia bajo la siguiente escala: de 70 a 100 ***“Aprobó”***, de 50 a 69 ***“Reposición”***, menos de 50 ***“Reprobó”***.

- f. La información de las notas debe estar contenidas en otro archivo denominado NOTAS.txt, cuyo campo de relación con el archivo ESTUDIANTES.txt, debe ser la cédula.

Ejemplo:

```

-----
|          REGISTRO DE NOTAS POR MATERIA          |
-----
Estudiante: Juan Pérez Rodríguez

Ingrese la cantidad de materias a registrar (Máximo 3): 2

-----
|          MATERIA 1          |
-----
Ingrese el nombre de la materia: Matemáticas

Ingrese las notas en base 10:
- Proyecto 1: 8.5
- Proyecto 2: 9.0
- Ensayo: 7.8
- Foro: 9.5
- Defensa: 8.0

✅ Promedio calculado: 8.56
✅ Estado: Aprobó

```

Reglas:

- i. No se pueden dejar campos vacíos.
- ii. La identificación debe ser un valor numérico y no se puede repetir
- iii. Se almacena en ESTUDIANTES.TXT. los datos del estudiante.
- iv. Las materias y notas se almacenan en NOTAS.txt, con el campo identificación que lo relaciona con ESTUDIANTES.txt.
- v. Las notas deben permitir el registro de dos decimales
- vi. No pueden existir valores nulos en los campos de notas.
- vii. El promedio debe considerar la siguiente escala de evaluación:

Proyecto 1 = 1%

Proyecto 2 = 2%

Ensayo = 3%

Foro = 1%

Defensa = 3%

- viii. El cálculo del promedio debe ser automático y guardarse en un campo ***promedio (incluye decimales)***. ***Este promedio debe quedar registrado en el archivo notas.txt***

- ix. El estado debe responder al promedio obtenido, según la escala dada en el inciso d, punto 2
- x. El promedio y el estado deben registrarse en el archivo notas.txt

Modificar notas de estudiante

- a. Se debe solicitar la Identificación del estudiante y buscarla en el archivo ESTUDIANTE.txt.
- b. Si la identificación no existe, debe mostrar un mensaje que indique “*Estudiante no registrado*” y preguntar si desea buscar otra identificación, en caso de ser Si “S”, vuelve a solicitar la identificación, en caso de ser No “N”, vuelve al menú principal. Debe convertir minúsculas a mayúsculas.
- c. Al encontrar la identificación, debe mostrar las materias registradas y seleccionar la que desea modificar (Recuerde que solo modificará las notas)

```
Ingrese la identificación del estudiante: 0101110111  
  
Materias registradas:  
1. Matematica  
2. Historia  
Seleccione la materia a modificar (1-2): |
```

- d. Debe mostrar las notas originales para la materia seleccionada e ingresar las nuevas notas en el orden establecido (mismas reglas para validar los campos de notas).

```
Seleccione la materia a modificar (1-2): 1  
  
Notas actuales para Matematica:  
Nota 1: 7.5  
Nota 2: 6.5  
Nota 3: 9.5  
Nota 4: 10  
Nota 5: 10  
Ingrese nueva nota 1 (0 a 100): |
```

- f. Al finalizar de actualizar las notas, debe indicar con un mensaje que la actualización fue exitosa y regresar al menu, o bien, puede agregar una opción para solicitar una nueva modificación (opcional)

Reporte de Estudiantes, promedios y estado.

Debe crear un reporte con las siguientes características:

- ✓ Encabezado claro con título "REPORTE DE ESTUDIANTES - NOTAS FINALES".
- ✓ Formato de tabla con columnas alineadas:
 - ID del estudiante (10 caracteres).
 - Nombre completo.
 - Materia cursada.
 - Promedio obtenido.
 - Estado final (Aprobó, Reposición, Reprobó).
- ✓ Diseño compacto y fácil de leer.

Ejemplo

```
=====
|   REPORTE DE ESTUDIANTES - NOTAS FINALES   |
=====
-----
| ID          | Nombre                | Materia    | Promedio | Estado  |
-----
| 0101110111 | Juan Pérez Rodríguez | Matemáticas | 8.56    | Aprobó  |
| 0101110111 | Juan Pérez Rodríguez | Física     | 6.56    | Reposición|
| 0234567890 | María López Fernández | Química    | 4.60    | Reprobó  |
| 0234567890 | María López Fernández | Biología   | 7.90    | Aprobó  |
-----
```

Los colores son ilustrativos.

Requerimientos Técnicos

- Uso de Variables: Numéricas, de cadena de caracteres y booleanas.
- Estructuras de Control: Condicionales (if, switch) y repetitivas (for(), while(), do-while()).
- Funciones:
 - Con y sin parámetros.
 - Manejo de excepciones “*try – catch - throw*” (al menos una función con captura de errores).
- Uso de Arreglos y Matrices:
 - Vector de estructuras para almacenar estudiantes.
 - Matriz para representar calificaciones de estudiantes.
- Almacenamiento de Información:
 - Uso de funciones para escribir y leer datos de un archivo de texto.

Validaciones y Reglas

1. Validación de Entrada:
 - Verificar que el ID de estudiante no se repita.
 - Limitar las calificaciones entre 0 y 100 (***debe permitir uso de dos decimales***)
 - Evitar el ingreso de datos vacíos o incorrectos.
2. Manejo de Excepciones:
 - Capturar errores en la entrada de datos numéricos inválidos.
 - Manejo de archivos con control de apertura y escritura.

Salir

- a. Preguntar antes de salir si desea salir Si “S”, finaliza, si es No “N”, se mantiene en el programa.

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 deben utilizar funciones y métodos en la programación para mejorar la organización y reutilización del código.
4. Se debe incluir todas las validaciones requeridas:
 - a. La cédula debe contener exactamente 10 dígitos numéricos, sin letras ni caracteres especiales.
 - b. Las notas y promedio deben permitir dos decimales
 - c. Campos vacíos. En las opciones de solicitud de información no se puede ingresar valores en blanco.
 - d. Opciones S y N validando mayúscula y minúscula. Respuestas a preguntas de Si o No (S/N). Si ingresa algo diferente no debe accionar.
 - e. Entrada de menú opciones controladas.
 - f. Validaciones
 - Números sin letras
 - Cédula debe ser de 10 dígitos.
 - Opciones "S/N" deben ser validadas.
 - Si se ingresa una opción inválida en el menú, el sistema debe mostrar un mensaje de error.
 - g. Manejo de Datos en Memoria
 - Implementación de Matrices para almacenar y manipular las notas de los estudiantes.
 - Carga automática de datos desde archivos al iniciar.
 - Guardado automático de datos en archivos al salir o bien cuando registra datos.
 - h. Funciones Principales mínimas
 - Registro de datos del estudiante.
 - Ingresar notas.
 - Modificar notas.
 - Eliminar registro.
 - Generar reportes
 - Carga y almacenamiento de datos en archivos.
 - i. Los colores y líneas mostradas en las pantallas de ejemplo son opcionales.
5. Tomar en consideración presentación en pantalla:
 - a. El menú debe mantenerse en la misma posición en todas las pantallas para garantizar una navegación uniforme.
 - b. No se deben desplegar caracteres especiales en la interfase de pantalla.
 - c. Correcta visualización y posicionamiento de los campos.
6. Documentación básica del proyecto:
 - a. El inicio del programa indicar: Fecha, autor del programa (nombre del estudiante), una breve descripción y referencias.
 - b. Comentar variables, procedimientos y funciones.
 - c. Agregar cualquier otro comentario pertinente.
7. 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 usuario vuelva al menú desde donde este y presiona la opción #5 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 < Proyecto #2>.
- La < Proyecto #2 > debe estar desarrollado en [CodeBlocks] 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-proyecto2.**
- La entrega de la <Proyecto #2> 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 estudiantes	15	8	3	0.1
Archivos. Utiliza manejo de archivos en su solución, en particular para llevar control de <i>notas</i>	15	8	3	0.1
Arreglos. Correcta declaración, recorrido y uso en general de los	15	8	3	0.1

arreglos solicitados o propios, en la solución planteada. Tomando en consideración cualquier solicitud específica en el enunciado. Incluye manejo de vector y matriz solicitadas para <i>estudiantes y notas</i> , según se ajuste al programa/código				
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 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			