|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor(a):* | Ariel Adara Mercado Martínez |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 2 |
| *No. de práctica(s):* | 7 |
| *Integrante(s):* | 1 |
| *No. de lista o brigada:* | 29 |
| *Semestre:* | Segundo |
| *Fecha de entrega:* | 12 de marzo de 2025 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Indice**

Introducción ….. 3

Desarrollo….. 3

Conclusión ….. 10

Introducción

En este conjunto de ejercicios se busco desarrollar y aplicar diferentes algoritmos utilizando el lenguaje de programación C,enfocados en la solución de problemas prácticos mediante el uso de estructuras de control con condicionales (if, else),bucles y otras funcionalidades como el manejo de cadenas de texto. Cada uno de los ejercicios propone un desafío distinto desde la implementación de un cálculo de impuestos progresivos hasta la optimización de rutas en un sistema de transporte,aumentando gradualmente la dificultad

Desarrollo

Ejercicio 1: Cálculo de Impuestos sobre el Salario

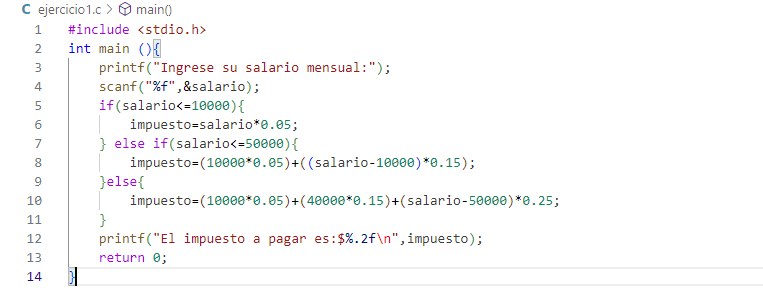
Este programa calcula el impuesto sobre el salario mensual de un trabajador, aplicando diferentes tasas a partir del nivel de ingresos.El impuesto es progresivo: el primer tramo de hasta 10000 unidades de salario paga un 5%,el siguiente tramo hasta 50000 paga un 15% y todo lo que supere los 50000 paga un 25%.

Ilustración 1Ejercicio 1

Reto:

Uno de los desafíos más importantes fue asegurarse de que las variables salario e impuesto estuvieran correctamente declaradas antes de su uso.Además tuve que manejar correctamente la entrada de datos y realizar los cálculos correctamente.

Ejercicio 2: Identificación del Tipo de Triángulo

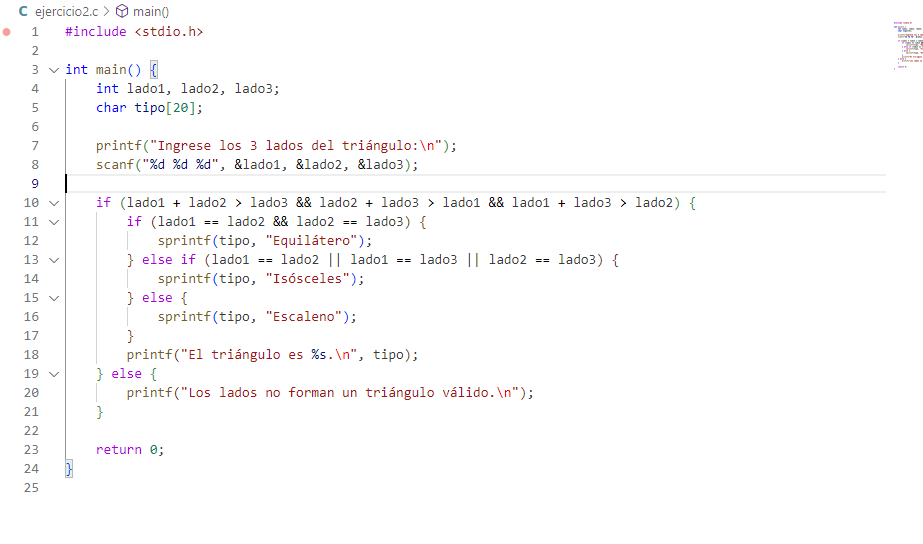
Este programa permite identificar el tipo de triángulo (equilátero,isósceles o escaleno) a partir de sus tres lados,se valida que los lados ingresados efectivamente formen un triángulo comprobando la desigualdad triangular.

Ilustración Ejercicio 2

Reto:

El reto fue asegurar que la comparación de los lados fuera correcta y que las condiciones para identificar los tipos de triángulo se implementaran correctamente a su vez fue necesario manejar correctamente las cadenas de texto con la función sprintf.

Ejercicio 3: Validación de Contraseña

Este programa valida una contraseña mediante dos comprobaciones: que la contraseña confirmada coincida con la ingresada y que la longitud sea al menos de 8 caracteres.Se utilizo la función strcmp para comparar las contraseñas y strlen para verificar la longitud mínima.Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Ilustración EJercicio 3

Reto:

El reto aquí fue manejar correctamente los saltos de línea al leer la entrada con fgets y asegurarse de que las contraseñas no incluyeran un salto de línea al final además fue importante que el programa verificara tanto la coincidencia como la longitud de la contraseña de manera eficiente.

Ejercicio 4: Cálculo del Nivel de Experiencia

Este programa evalúa el nivel de experiencia de un usuario basado en sus puntos de experiencia,los niveles son "Novato","Intermedio","Avanzado" y "Experto", los cuales son definidos por rangos de puntos.

Ilustración Ejercicio 4

Reto:

El reto aquí fue usar la función strcpy para asignar el nivel,así como organizar las condiciones de manera clara para que cubrieran todos los casos posibles.

Ejercicio 5: Determinación de Días en un Mes

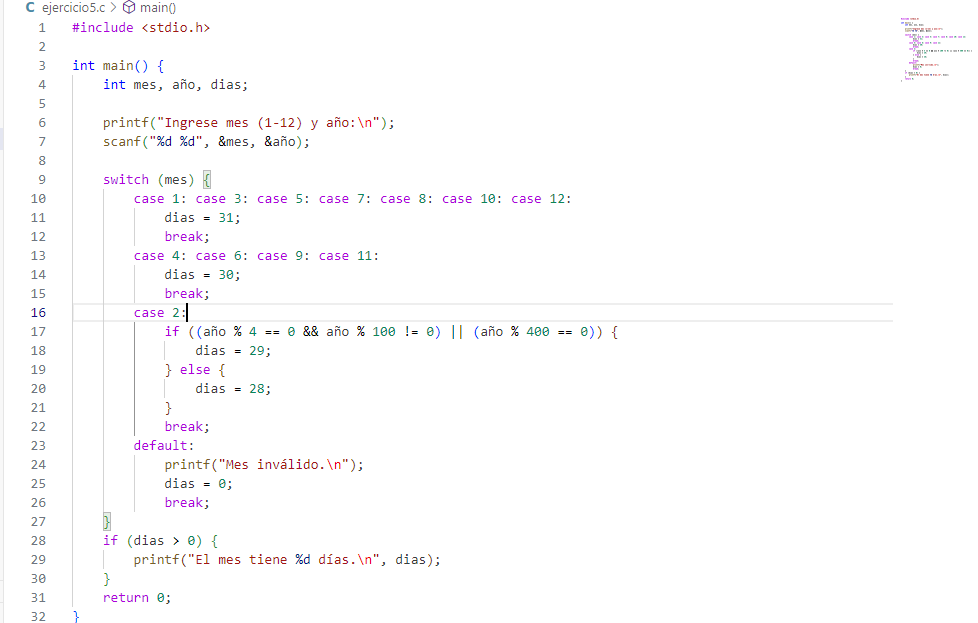
El programa determina la cantidad de días que tiene un mes dado considerando si el año es bisiesto en el caso del mes de febrero,se implementa usando una estructura de switch que maneja los casos de meses con 31, 30 y 28/29 días.

Ilustración Ejercicio 5

Reto:

Un reto en este ejercicio fue implementar correctamente la condición del año bisiesto ya que requiere verificar que el año sea divisible por 4 y que no sea divisible por 100,excepto si también es divisible por 400.

Ejercicio 6: Cálculo de Descuento según Monto de Compra

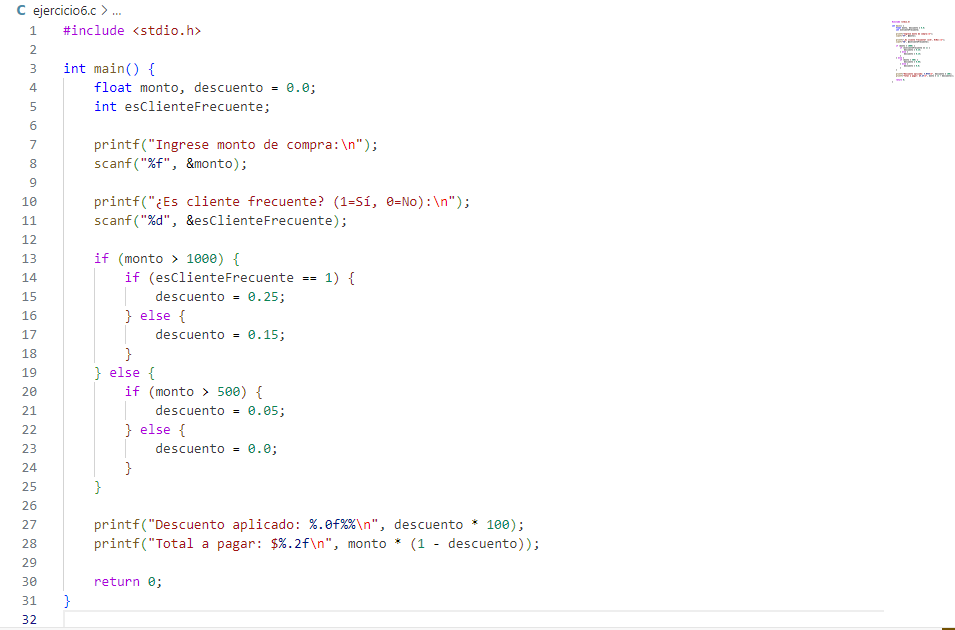
Este programa calcula el descuento que se debe aplicar en una compra dependiendo del monto y si el cliente es frecuente o no.Los descuentos se aplican en diferentes porcentajes dependiendo de la cantidad de dinero gastado.

Ilustración Ejercicio 6

Reto:

El reto en este caso fue manejar correctamente las condiciones para la aplicación de los descuentos y asegurarse de que el cálculo del total a pagar se hiciera correctamente considerando el descuento.

Ejercicio 7: Determinación de Ruta Óptima de Transporte

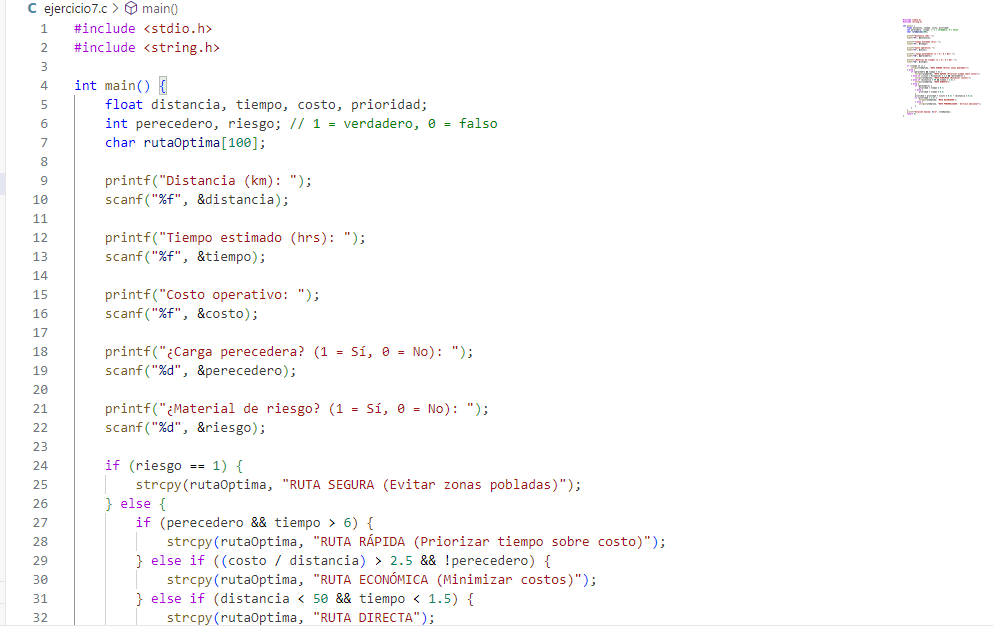
Este programa evalúa las mejores rutas para transportar mercancías,teniendo en cuenta factores como la distancia,el tiempo,el costo operativo y las características de la carga,según estos parámetros el programa determina una ruta óptima.

Ilustración Ejercicio 7.1Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Ilustración Ejercicio7.2

Reto:

El reto principal aquí fue la complejidad de las condiciones anidadas, que requerían una correcta priorización de los diferentes factores (como la urgencia del tiempo y los costos). Además, el cálculo de "prioridad" para determinar la ruta balanceada fue una parte crítica.

Conclusión

El desarrollo de estos siete ejercicios permitió aplicar de manera práctica los fundamentos esenciales del lenguaje C,referente a estructuras condicionales,manejo de variables,entrada/salida de datos y manipulación de cadenas

También la relevancia del manejo adecuado de tipos de datos y funciones auxiliares, como strcmp,strlen,strcpy, y el uso de fgets en lugar de scanf para leer cadenas de texto de forma segura.Estas herramientas son esenciales cuando se trabaja con datos ingresados por el usuario ya que permiten prevenir errores comunes como desbordamientos de búfer o lecturas incorrectas.En el ejercicio de la contraseña,por ejemplo,fue crucial eliminar los saltos de línea para que la comparación de cadenas funcionara correctamente.

Además, varios ejercicios presentaron retos de diseño lógico,donde no bastaba con traducir las condiciones del problema a código sino que también fue necesario reflexionar sobre posibles casos especiales o excepciones.Por ejemplo en el ejercicio de los días del mes,se tuvo que considerar la validez del mes ingresado y aplicar correctamente las reglas de los años bisiestos,lo que implicó el uso de operadores lógicos compuestos y estructuras condicionales múltiples.

En resumen estos ejercicios no solo reforzaron habilidades técnicas específicas del lenguaje C,sino que también fomentaron un pensamiento crítico y estructurado para la resolución de problemas.La práctica constante,el análisis de errores y la mejora progresiva del código fueron pilares fundamentales en el aprendizaje obtenido durante esta experiencia.