



UNIVERSIDAD ABIERTA Y A DISTANCIA DE MÉXICO

División de Ciencias Exactas, Ingeniería y Tecnología

Desarrollo de software

Semestre: Primer semestre.

Unidad didáctica: 01 – Fundamentos de programación.

Unidad de aprendizaje: **Unidad 2.** Introducción al lenguaje C.

Actividad: Beneficios de los algoritmos secuenciales.

Nombre del estudiante: Orlando Antonio Maturano Pizaña

Matrícula: ES251107915

Grupo: DS-DFPR-2501-B1-013

Figura académica: JOSÉ MANUEL NORIEGA BARRERA

Fecha de entrega: 18 de febrero de 2025

Ciudad de México, a 12 de febrero del 2025.

ÍNDICE

INTRODUCCIÓN	1
DESARROLLO DE LA ACTIVIDAD	1
ANÁLISIS DEL PROBLEMA	1
PSEUDOCÓDIGO	3
DESARROLLO DEL CÓDIGO EN C	10
CONCLUSIÓN	14
Citas de autor	14
REFERENCIAS BIBLIOGRÁFICAS	15

INTRODUCCIÓN

La programación estructurada en lenguaje C constituye un pilar fundamental en la formación de competencias lógico-matemáticas y la resolución de problemas en el ámbito computacional. Su relevancia radica en la capacidad de diseñar algoritmos secuenciales que automatizan procesos críticos, como la gestión financiera personal mediante pasos claros y ordenados. En este contexto, el caso de estudio planteado ejemplifica cómo la programación trasciende lo teórico para impactar en situaciones cotidianas. Los antecedentes de esta actividad se enmarcan en la necesidad de desarrollar herramientas precisas que optimicen la toma de decisiones económicas, evitando errores humanos en cálculos manuales. El objetivo central es «aplicar estructuras algorítmicas secuenciales para resolver problemas prácticos, integrando entrada/salida de datos y operaciones aritméticas, bajo los principios de claridad y eficiencia» (UnADM, s.f. Pág 4) Esta tarea se delimita en tres ejes: 1) Diseño de interfaces sencillas, 2) implementación de cálculos con precisión numérica, 3) presentación estructurada de resultados. Al abordar estos aspectos, se busca no sólo dominar la sintaxis del lenguaje C sino que también fomentar una mentalidad analítica ante desafíos reales.

DESARROLLO DE LA ACTIVIDAD

Análisis del problema:

Un trabajador desea organizar su salario mensual distribuyendo en seis categorías con porcentajes predefinidos: Renta (25%), alimentos (30%), pagos bancarios (7%), educación (20%), otros gastos (10%) y diversión (el porcentaje restante). Para ello se desarrollará un programa en lenguaje C que realice este cálculo de manera automática

El programa debe iniciar mostrando mis datos de estudiante (nombre, matrícula y grupo). Posteriormente, solicitará al usuario del programa su nombre (nombre del trabajador) y su salario mensual. A partir de estos datos el programa calculará el monto correspondiente para cada categoría, aplicando los porcentajes establecidos. Finalmente se imprimirá un desglose detallado en pantalla, incluyendo cada categoría, su porcentaje y el monto en pesos asignado.

Este programa facilita la organización financiera del trabajador, brindándole una visualización clara de cómo se distribuye su salario, lo que puede ayudar en la toma de decisiones económicas.

Variables identificadas

Nombre del estudiante desarrollador.

Matrícula del estudiante.

Grupo del estudiante.

Nombre del trabajador.

Salario mensual ingresado por el usuario.

renta, alimentos, pagos bancarios, educación, otros gastos, diversión (flotantes): Montos asignados a cada categoría.

Constantes

Porcentaje de renta = 0.25 (25%)

Porcentaje de alimentos = 0.30 (30%)

Porcentaje de pagos bancarios = 0.07 (7%)

Porcentaje de educación = 0.20 (20%)

Porcentaje de otros gastos = 0.10 (10%)

Porcentaje de diversión = 1 - (Suma de los otros porcentajes) (Calculado dinámicamente)

Expresiones matemáticas

Cada categoría se calcula multiplicando el salario por su porcentaje correspondiente:

renta = salario * PORCENTAJE_RENTA

alimentos = salario * PORCENTAJE_ALIMENTOS

pagos_bancarios = salario * PORCENTAJE_PAGOS_BANCARIOS

educacion = salario * PORCENTAJE_EDUCACION

otros_gastos = salario * PORCENTAJE_OTROS_GASTOS

diversion = salario * PORCENTAJE_DIVERSION

Librerías necesarias

<stdio.h>: Para funciones de entrada y salida (printf, scanf).

<stdlib.h>: Para funciones adicionales si se requiere (como system("pause") en Windows).

Funciones necesarias

void imprimirDatosEstudiante(); Muestra el nombre, matrícula y grupo del estudiante.

void solicitarDatosTrabajador(char nombre[], float *salario); Solicitud del nombre del trabajador y su salario.

void calcularMontos(float salario, float *renta, float *alimentos, float *pagos, float *educacion, float *otros, float *diversion); Realiza los cálculos de los montos.

void imprimirDesglose(float renta, float alimentos, float pagos, float educacion, float otros, float diversion); Muestra el desglose final en pantalla.

Pseudocódigo

Se utilizaron las siguientes letras para las variables del pseudo código:

N_E (Nombre del estudiante).
M (Matrícula).
G (Grupo).
N_T (Nombre del trabajador).
S (Salario mensual).
R (Renta).
A (Alimentos).
P (Pagos bancarios)
E (Educación)
O (Otros gastos)
D (División)

Librerías utilizadas:

Uso de entrada/salida: L_E/S (Equivalente a <stdio.h>)
Funciones del sistema: L_S (Equivalente a <stdlib.h>)

Orlando Antonio Maturana Pizanía
ES 25 11079 15

Inicio

1. Mostrar datos del estudiante

Usar L-E/S para pedir N-E

Usar L-E/S para pedir M

Usar L-E/S para pedir G

Usar L-E/S para imprimir N-E, M, G.

2. Solicitar datos del trabajador.

Usar L-E/S para pedir N-T

Usar L-E/S para pedir S

3. Calcular montos de cada categoría

$$R = S \times 0.25$$

$$A = S \times 0.30$$

$$P = S \times 0.07$$

$$E = S \times 0.20$$

$$O = S \times 0.10$$

$$D = S - (R + A + P + E + O)$$

4. Mostrar desglose

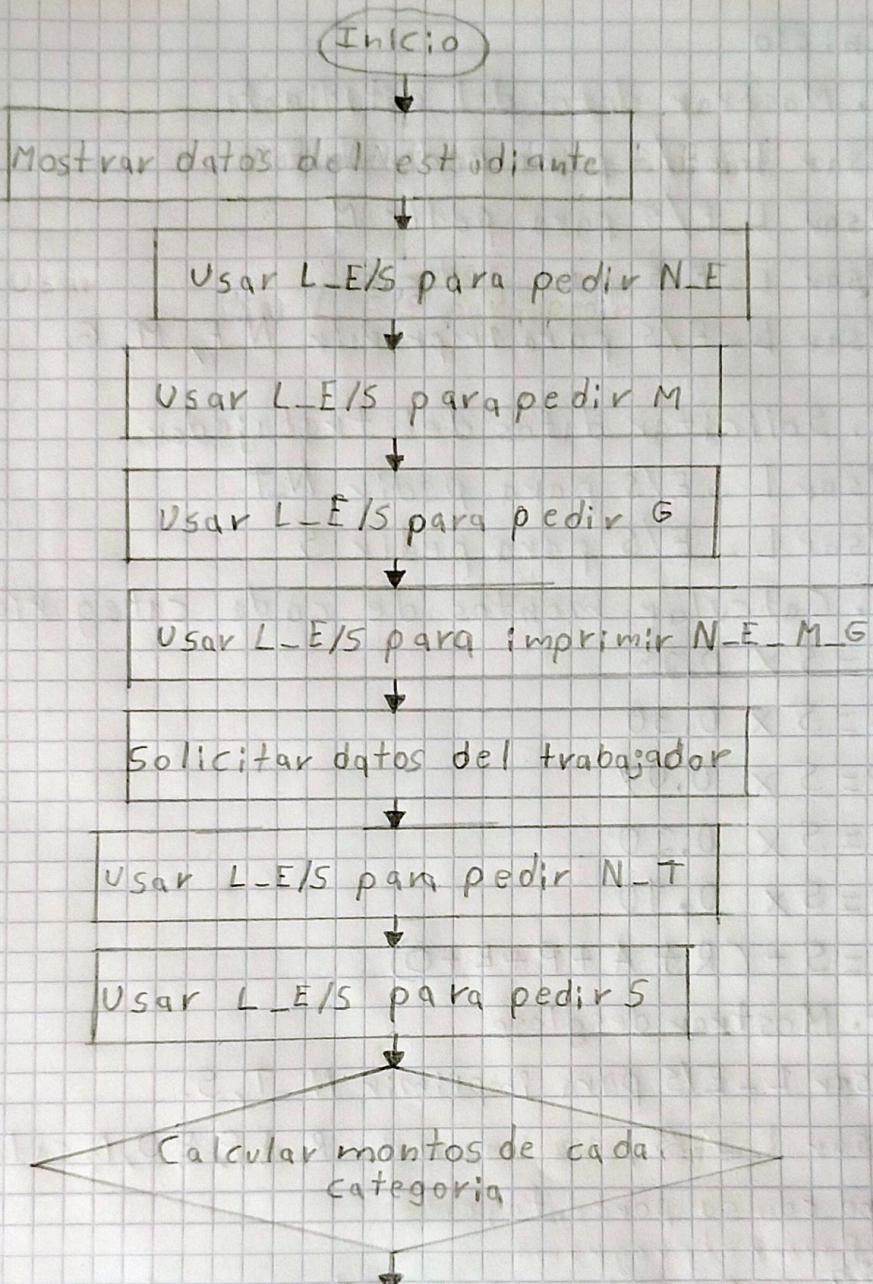
Usar L-E/S para imprimir N-T, S.

Usar L-E/S para imprimir R, A, P, E, O, D, cada

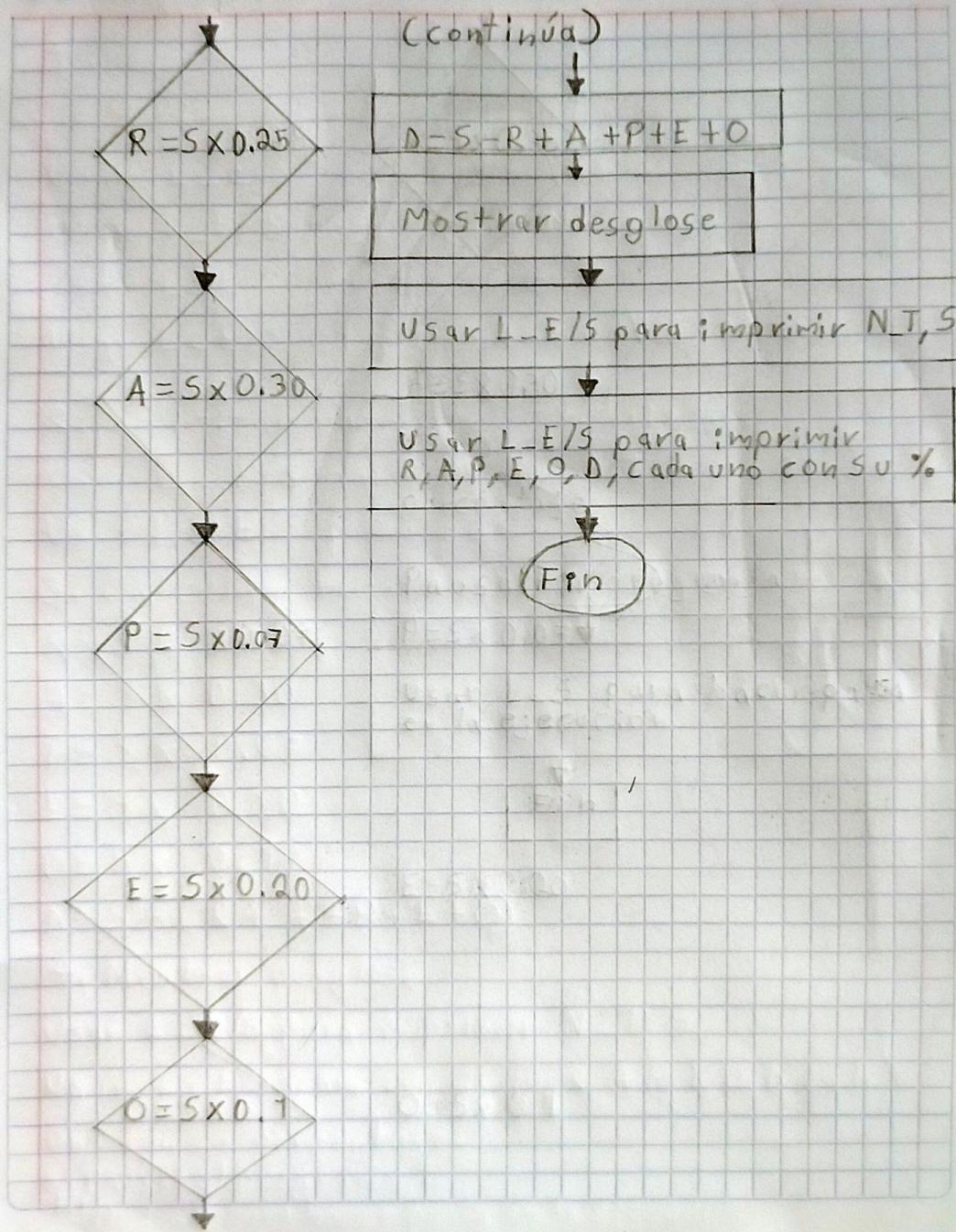
uno con su porcentaje.

Fin

Orlando Antonio Maturano Pizaña
ES257107915



Orlando Antonio Maturano Pizana
ES 251107975



Prueba de escritorio

Orlando Antonio Maturano Pizaña
ES 25 11 07975

7. Datos del estudiante:

Salida:

Nombre del estudiante: Orlando Antonio
Maturano Pizaña

Matrícula: ES 25 11 07975

Grupo: DS-DFPR-2507-B1-013

Acción: El programa muestra los datos.

2-Datos del trabajador.

Se solicita el nombre de usuario o N-T y
el salario mensual.

Ingresé su nombre: Aníra López Valencia

Ingresé su salario mensual: 500

3-Proceso:

Se calcula el desglose de gastos segun
el salario de gastos

Categoría	Formula	Calculo
R	5×0.25	500×0.25
A	5×0.30	500×0.30
P	5×0.07	500×0.07
E	5×0.20	500×0.20
O	5×0.70	500×0.70
D	$5 \times (R+A+P+E+O)$	$500 - (125 + 150$ $+ 35 + 100 + 50)$

Orlando Antonio Maturano Pizana
ES257107975

Resultados

R	725
A	750
P	35
E	100
O	50
D	40

Salida esperada:

Datos del estudiante: N, M, G

Datos del trabajador: N y S.

Desglose

Renta (25%): 725

Alimentos (30%): 750

Pagos bancarios (7%): 35

Educación (20%): 100

Otros gastos (10%): 50

División (Restante): 40

Estos datos muestran resultados coherentes.

Un ejemplo de salida erronea sería:

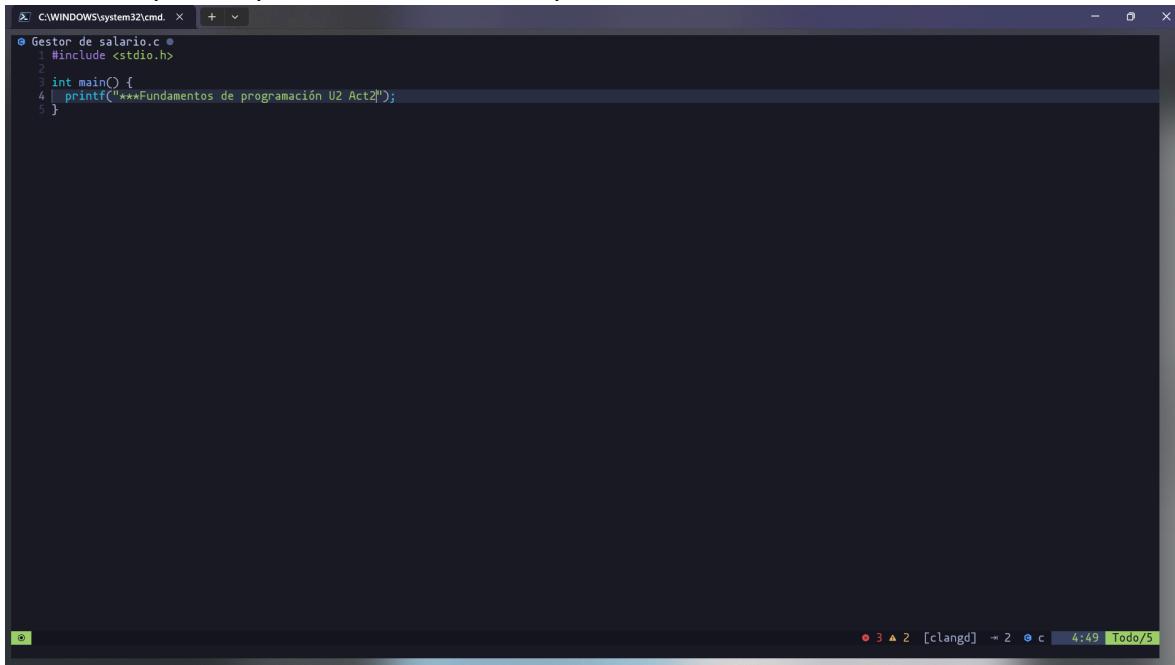
R: - 725

A: - 750

P: - 35

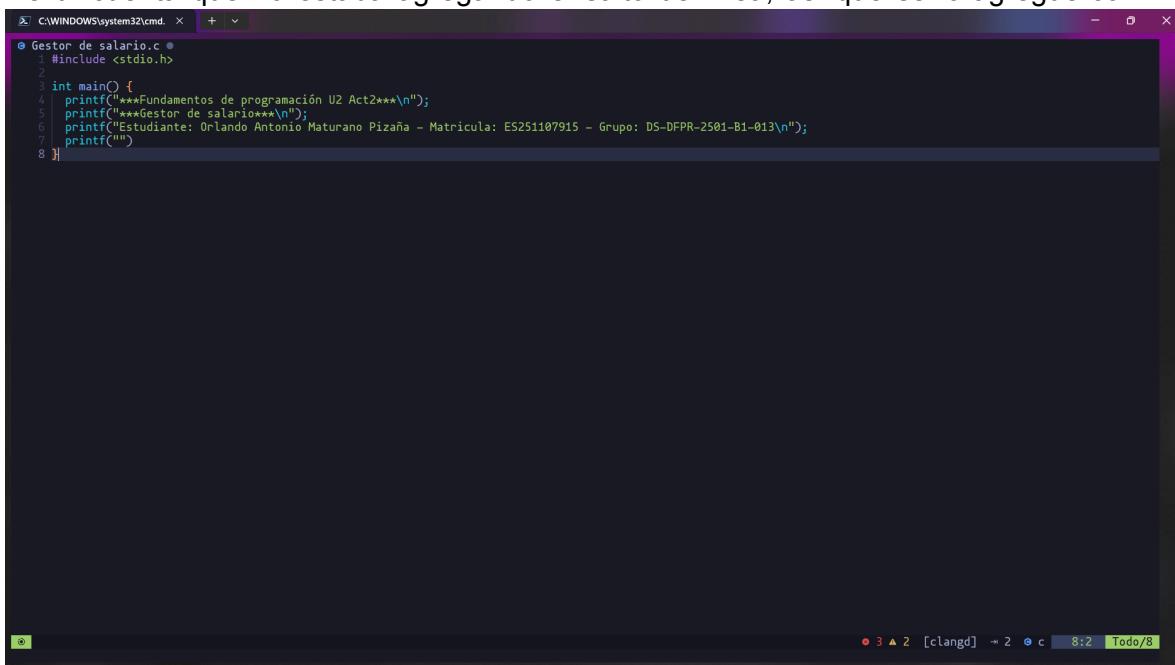
Desarrollo del código en lenguaje C:

Primero empecé imprimiendo los datos requeridos:



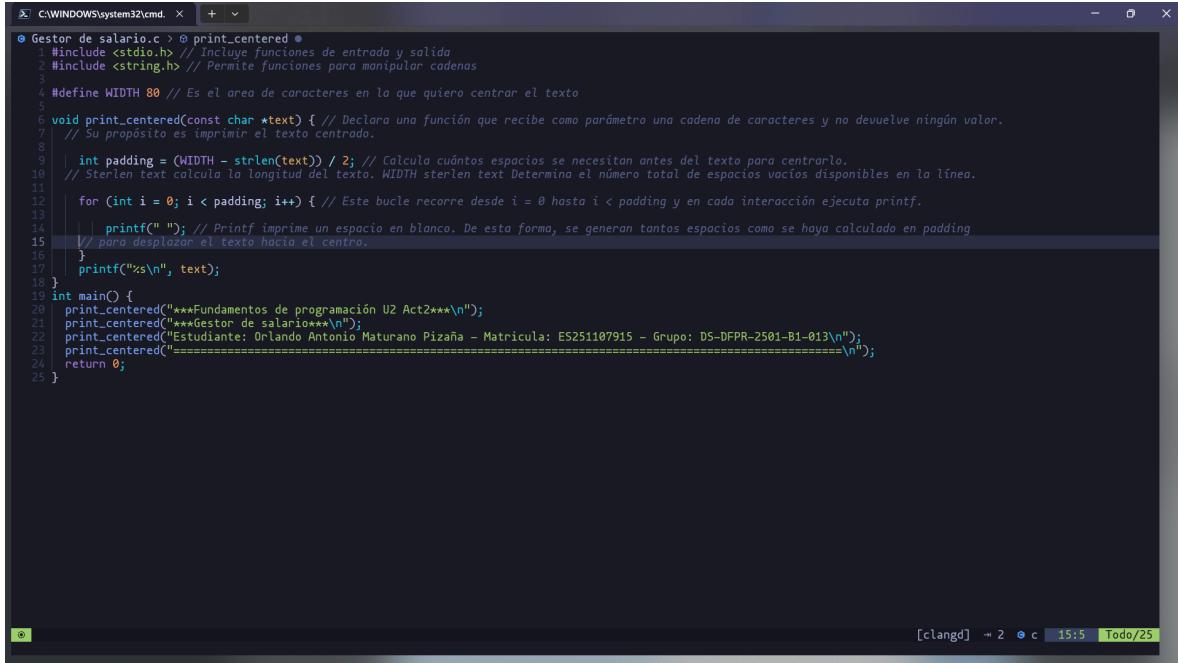
```
C:\WINDOWS\system32\cmd. + 
• Gestor de salario.c •
1 #include <stdio.h>
2
3 int main() {
4     printf("****Fundamentos de programación U2 Act2****");
5 }
```

Me di cuenta que no estaba agregando el salto de linea, asi que se lo agregué con \n.



```
C:\WINDOWS\system32\cmd. + 
• Gestor de salario.c •
1 #include <stdio.h>
2
3 int main() {
4     printf("****Fundamentos de programación U2 Act2****\n");
5     printf("****Gestor de salario****\n");
6     printf("Estudiante: Orlando Antonio Maturano Pizaña - Matricula: ES251107915 - Grupo: DS-DFPR-2501-B1-013\n");
7     printf("");
8 }
```

Quise centrar el texto que se imprimirá e hice uso de #include <string.h>



```

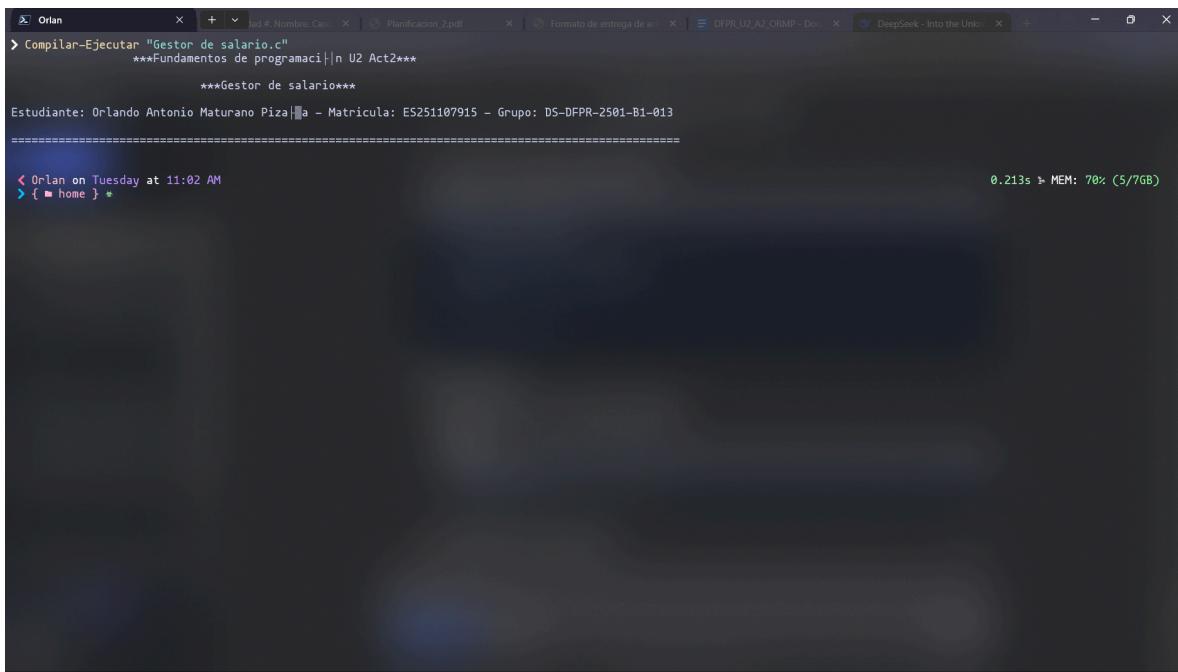
C:\WINDOWS\system32\cmd. x + ^

Gestor de salario.c ② print_centered •
1 #include <stdio.h> // Incluye funciones de entrada y salida
2 #include <string.h> // Permite funciones para manipular cadenas
3
4 #define WIDTH 80 // Es el área de caracteres en la que quiero centrar el texto
5
6 void print_centered(const char *text) { // Declara una función que recibe como parámetro una cadena de caracteres y no devuelve ningún valor.
7 // Su propósito es imprimir el texto centrado.
8
9     int padding = (WIDTH - strlen(text)) / 2; // Calcula cuántos espacios se necesitan antes del texto para centrarlo.
10    // strlen(text) calcula la longitud del texto. WIDTH determina el número total de espacios vacíos disponibles en la línea.
11
12    for (int i = 0; i < padding; i++) { // Este bucle recorre desde i = 0 hasta i < padding y en cada interacción ejecuta printf.
13        printf(" "); // Printf imprime un espacio en blanco. De esta forma, se generan tantos espacios como se haya calculado en padding
14        // para desplazar el texto hacia el centro.
15    }
16    printf("%s\n", text);
17 }
18
19 int main() {
20     print_centered("***Fundamentos de programación U2 Act2***\n");
21     print_centered("***Gestor de salario***\n");
22     print_centered("Estudiante: Orlando Antonio Maturano Pizana - Matrícula: ES251107915 - Grupo: DS-DFPR-2501-B1-013\n");
23     print_centered("=====*\n");
24     return 0;
25 }

```

[clangd] → 2 ② c 15:5 Todo/25

Al compilar y ejecutar el código algunos caracteres no se imprimieron bien:



```

Orlan x + ^ Jef #, Nombre, Cognom x Planificación_2.pdf x Formato de entrega de a... x DFPR_U2_A2_ORMP - Doc... x DeepSeek - Into the Unkn...
Compilar-Ejecutar "Gestor de salario.c"
***Fundamentos de programaci|n U2 Act2***
***Gestor de salario***
Estudiante: Orlando Antonio Maturano Piza|a - Matrícula: ES251107915 - Grupo: DS-DFPR-2501-B1-013
=====
< Orlan on Tuesday at 11:02 AM 0.213s ▶ MEM: 70% (5/7GB)
> { [ home ] *

```

Para que los caracteres se imprimen bien se debe guardar el archivo en UTF-8 y codificar UTF-8 por defecto.

```

C:\WINDOWS\system32\cmd. x + v
@ Gestor de salario.c
 4 #include <locale.h> // Para configuración regional (setlocale)
 5 #ifndef _WIN32 // Si el sistema es Windows...
 6 #include <windows.h> // Incluir funciones de la API de Windows (SetConsoleOutputCP)
 7 #endiff
 8
 9 #define WIDTH 80 // Ancho de la consola para centrado (estándar: 80 caracteres)
10
11 // --- Configuración de codificación ---
12 void configure_encoding() {
13     #ifdef _WIN32
14         // Windows: Configura la consola para usar UTF-8 (evita caracteres corruptos)
15         SetConsoleOutputCP(CP_UTF8);
16     #endiff
17     // Configuración regional para todos los sistemas (habilita caracteres especiales)
18     setlocale(LC_ALL, "en_US.UTF-8"); // "en_US.UTF-8" es compatible con la mayoría de sistemas
19 }
20
21 // --- Función para imprimir texto centrado ---
22 void print_centered(const char *text) {
23     int len = strlen(text); // Longitud original del texto
24
25     // Ajuste de longitud si el texto termina en salto de línea:
26     if (len > 0 && text[len - 1] == '\n') len--;
27     // Ignora '\n' para cálculos de centrado
28
29     // Cálculo de espacios a la izquierda:
30     int padding = (WIDTH - len) / 2; // Espacios = (AnchoTotal - LongitudTexto) / 2
31     if (padding < 0) padding = 0; // Evita valores negativos (textos más largos que WIDTH)
32
33     // Impresión eficiente:
34     // '%*s': Crea una cadena de 'padding' espacios en blanco
35     // '%s': Imprime el texto original (incluyendo su '\n' si lo tiene)
36     printf("%*s%s", padding, "", text);
37 }
38 // --- Función principal ---
39 int main() {
40     configure_encoding(); // Primero: Asegurar que la consola maneje UTF-8
41     // Impresión de líneas centradas:
42     print_centered("****Fundamentos de programación U2 Act2***\n");
43     print_centered("****Gestor de salario***\n");
44     print_centered("Estudiante: Orlando Antonio Maturano Pizana - Matrícula: ES251107915 - Grupo: DS-DFPR-2501-B1-013\n");
45     print_centered("=====*\n");
46 }

```

[clangd] + 2 o c | 45:29 Final/46

Ahora si se imprime como debe ser:

```

Orlan x + v
Compilar-Ejecutar "Gestor de salario.c"
    ***Fundamentos de programación U2 Act2***
    ***Gestor de salario***
Estudiante: Orlando Antonio Maturano Pizana - Matrícula: ES251107915 - Grupo: DS-DFPR-2501-B1-013
=====
Orlan on Tuesday at 11:47 AM
> { home } *
0.457s ▶ MEM: 71% (5/7GB)

```

Ahora procederemos con la solicitud de entrada de datos:

```

C:\WINDOWS\system32\cmd. x + 
@ Gestor de salario.c > @ main
15 |     SetConsoleOutputCP(CP_UTF8);
16 | #endif
17 // Configuración regional para todos los sistemas (habilita caracteres especiales)
18 | setlocale(LC_ALL, "en_US.UTF-8"); // "en_US.UTF-8" es compatible con la mayoría de sistemas
19 }
20
21 // --- Función para imprimir texto centrado ---
22 void print_centered(const char *text) {
23     int len = strlen(text); // Longitud original del texto
24
25     // Ajuste de longitud si el texto termina en salto de línea:
26     if (len > 0 && text[len - 1] == '\n') len--;
27
28     // Cálculo de espacios a la izquierda:
29     int padding = (WIDTH - len) / 2; // Espacios = (AnchoTotal - LongitudTexto) / 2
30     if (padding < 0) padding = 0; // Evita valores negativos (textos más largos que WIDTH)
31
32     // Impresión eficiente:
33     // - "%*s": Crea una cadena de 'padding' espacios en blanco
34     // - "%s": Imprime el texto original (incluyendo su '\n' si lo tiene)
35     printf("%*s%s", padding, "", text);
36 }
37 // --- Función principal ---
38 int main() {
39     configure_encoding(); // Primero: Asegurar que la consola maneje UTF-8
40     // Impresión de líneas centradas:
41     print_centered("****Fundamentos de programación U2 Act2***\n");
42     print_centered("****Gestor de salario****\n");
43     print_centered("Estudiante: Orlando Antonio Maturano Pizaña - Matrícula: ES251107915 - Grupo: DS-DFPR-2501-B1-013\n");
44     print_centered("=====*\n");
45
46     // --- Entrada de datos ---
47     char nombre[100];
48     float salario;
49
50     printf("----Ingrese su nombre: ");
51     fgets(nombre, sizeof(nombre), stdin); // Lee nombre con espacios
52     nombre[strcspn(nombre, "\n")] = '\0'; // Elimina salto de línea
53
54     printf("Ingrese su salario mensual: ");
55     scanf("%f", &salario); // Lee salario
56     return 0; // Fin exitoso
57 }

```

"Gestor de salario.c" 46L, 2198B escritos [clangd] ↗ 2 ⚡ c 55:66 Final/57

Después con los cálculos:

```

C:\WINDOWS\system32\cmd. x + 
@ Gestor de salario.c > @ main
24
25     // Ajuste de longitud si el texto termina en salto de línea:
26     if (len > 0 && text[len - 1] == '\n') len--; // Ignora '\n' para cálculos de centrado
27
28     // Cálculo de espacios a la izquierda:
29     int padding = (WIDTH - len) / 2; // Espacios = (AnchoTotal - LongitudTexto) / 2
30     if (padding < 0) padding = 0; // Evita valores negativos (textos más largos que WIDTH)
31
32     // Impresión eficiente:
33     // - "%*s": Crea una cadena de 'padding' espacios en blanco
34     // - "%s": Imprime el texto original (incluyendo su '\n' si lo tiene)
35     printf("%*s%s", padding, "", text);
36 }
37 // --- Función principal ---
38 int main() {
39     configure_encoding(); // Primero: Asegurar que la consola maneje UTF-8
40     // Impresión de líneas centradas:
41     print_centered("****Fundamentos de programación U2 Act2***\n");
42     print_centered("****Gestor de salario****\n");
43     print_centered("Estudiante: Orlando Antonio Maturano Pizaña - Matrícula: ES251107915 - Grupo: DS-DFPR-2501-B1-013\n");
44     print_centered("=====*\n");
45
46     // --- Entrada de datos ---
47     char nombre[100];
48     float salario;
49
50     printf("----Ingrese su nombre: ");
51     fgets(nombre, sizeof(nombre), stdin); // Lee nombre con espacios
52     nombre[strcspn(nombre, "\n")] = '\0'; // Elimina salto de línea
53
54     printf("Ingrese su salario mensual: ");
55     scanf("%f", &salario); // Lee salario
56
57     // --- Cálculos:
58     const float RENTA = salario * 0.25f;
59     const float ALIMENTOS = salario * 0.30f;
60     const float BANCO = salario * 0.07f;
61     const float EDUCACION = salario * 0.20f;
62     const float OTROS = salario * 0.10f;
63     const float DIVERSION = salario - (RENTA + ALIMENTOS + BANCO + EDUCACION + OTROS);
64
65     return 0; // Fin exitoso
66 }

```

"Gestor de salario.c" 46L, 2198B escritos [clangd] ↗ 2 ⚡ c 64:1 Final/66

Y por último con la salida formateada:

```

C:\WINDOWS\system32\cmd. x + v
Gestor de salario.c @ main
34 | // - '%s': Imprime el texto original (incluyendo su '\n' si lo tiene)
35 |     printf("%s%s", padding, "", text);
36 }
37 // --- Función principal ---
38 int mainQ {
39     config_encoding(); // Primero: Asegurar que la consola maneje UTF-8
40     // Impresión de líneas centradas:
41     print_centered("***Fundamentos de programación U2 Act2***\n");
42     print_centered("Gestor de salario\n");
43     print_centered("Estudiante: Orlando Antonio Maturano Pizaña - Matrícula: ES251107915 - Grupo: DS-DFPR-2501-B1-013\n");
44     print_centered("=====\\n");
45
46 // --- Entrada de datos ---
47 char nombre[100];
48 float salario;
49
50 printf("----Ingrese su nombre: ");
51 fgets(nombre, sizeof(nombre), stdin); // Lee nombre con espacios
52 nombre[strcspn(nombre, "\n")] = '\0'; // Elimina salto de línea
53
54 printf("Ingrese su salario mensual: ");
55 scanf("%f", &salario); // Lee salario █ 'scanf' is deprecated: This function or variable may be unsafe. Consider using scanf_s instead.
56
57 // --- Cálculos ---
58 const float RENTA = salario * 0.25f;
59 const float ALIMENTOS = salario * 0.30f;
60 const float BANCO = salario * 0.07f;
61 const float EDUCACION = salario * 0.20f;
62 const float OTROS = salario * 0.10f;
63 const float DIVERSION = salario - (RENTA + ALIMENTOS + BANCO + EDUCACION + OTROS);
64
65 // --- Salida formateada ---
66 printf("\n--Nombre del trabajador: %s\n", nombre);
67 printf("Distribución del salario:\n");
68 printf("Renta: $%.2f (25%%)\n", RENTA);
69 printf("Comida: $%.2f (30%%)\n", ALIMENTOS);
70 printf("Pagos del banco: $%.2f (7%%)\n", BANCO);
71 printf("Educación: $%.2f (20%%)\n", EDUCACION);
72 printf("Otros gastos: $%.2f (10%%)\n", OTROS);
73 printf("División: $%.2f (0.8%%)\n", DIVERSION, (DIVERSION/salario)*100);
74
75 return 0; // Fin exitoso
76 }

"Gestor de salario.c" 76L, 3401B escritos

```

Al ejecutar y compilar el programa, e ingresar los datos de usuario el programa funciona correctamente, en esta ocasión colocaré la cifra 15759.33 para comprobar que los cálculos funcionen correctamente:

```

Orlan x + v
Compilar-Ejecutar "Gestor de salario.c"
Gestor de salario.c:55:5: warning: 'scanf' is deprecated: This function or variable may be unsafe. Consider using scanf_s instead. To disable deprecation, use
      _CRT_SECURE_NO_WARNINGS. See online help for details. [-Wdeprecated-declarations]
  55 |     scanf("%f", &salario); // Lee salario
C:\Program Files (x86)\Windows Kits\10\Include\10.0.22621.0\ucrt\stdio.h:1275:20: note: 'scanf' has been explicitly marked deprecated here
1275 |     _Check_return_ __CRT_INSECURE_DEPRECATED(scanf_s
C:\Program Files (x86)\Microsoft Visual Studio\2022\BuildTools\VC\Tools\MSVC\14.43.34808\include\vcruntime.h:368:55: note: expanded from macro '__CRT_INSECURE_DEPRECATED'
  368 | #define __CRT_INSECURE_DEPRECATED(_Replacement) __CRT_DEPRECATE_TEXT(_
C:\Program Files (x86)\Microsoft Visual Studio\2022\BuildTools\VC\Tools\MSVC\14.43.34808\include\vcruntime.h:358:47: note: expanded from macro '__CRT_DEPRECATE_TEXT'
  358 | #define __CRT_DEPRECATE_TEXT(_Text) __declspec(deprecated(_Text))
  1 warning generated.
      ***Fundamentos de programación U2 Act2***
      ***Gestor de salario***
Estudiante: Orlando Antonio Maturano Pizaña - Matrícula: ES251107915 - Grupo: DS-DFPR-2501-B1-013
=====
---Ingrese su nombre: Orlando
Ingrese su salario mensual: 15759.33
---Nombre del trabajador: Orlando
Distribución del salario:
Renta: $3939.83 (25%)
Comida: $4727.80 (30%)
Pagos del banco: $103.45 (7%)
Educación: $3151.87 (20%)
Otros gastos: $1575.93 (10%)
División: $1260.75 (8%)

```

Al final corregí el cierre automático del programa con la librería `#include <stdlib.h>` y coloqué la función de cerrar con enter para su uso en Mac OS y distribuciones de Linux.

```

C:\WINDOWS\system32\cmd. x | + | Planificación x | Formato de x | DeepSeek- x | Documento x | ANTLOGIA x | algoritmos x | Algoritmos x |
@ Gestor de salario.c @ main
42 print_centered("****Fundamentos de programación U2 Act2***\n");
43 print_centered("****Gestor de salario**\n");
44 print_centered("Estudiante: Orlando Antonio Maturano Pizaña - Matrícula: ES251107915 - Grupo: DS-DFPR-2501-B1-013\n");
45 print_centered("=====\n");
46
47 // --- Entrada de datos ---
48 char nombre[100];
49 float salario;
50
51 printf("---Ingrese su nombre: ");
52 fgets(nombre, sizeof(nombre), stdin);           // Lee nombre con espacios
53 nombre[strcspn(nombre, "\n")] = '\0';           // Elimina salto de línea
54
55 printf("Ingrese su salario mensual: ");
56 scanf("%f", &salario);                         // Lee salario ■ 'scanf' is deprecated: This function or variable may be unsafe. Consider using scanf_s instead
57
58 // --- Cálculos ---
59 const float RENTA = salario * 0.25f;
60 const float ALIMENTOS = salario * 0.30f;
61 const float BANCO = salario * 0.07f;
62 const float EDUCACION = salario * 0.20f;
63 const float OTROS = salario * 0.10f;
64 const float DIVERSION = salario - (RENTA + ALIMENTOS + BANCO + EDUCACION + OTROS);
65
66 // --- Salida formateada ---
67 printf("\n---Nombre del trabajador: %s\n", nombre);
68 printf("Distribución del salario:\n");
69 printf("Renta: %.2f (30%) \n", RENTA);
70 printf("Comida: %.2f (30%) \n", ALIMENTOS);
71 printf("Gastos de banco: %.2f (7%) \n", BANCO);
72 printf("Educación: %.2f (20%) \n", EDUCACION);
73 printf("Otros gastos: %.2f (10%) \n", OTROS);
74 printf("Diversion: %.2f (0.8%) \n", DIVERSION, (DIVERSION/salario)*100);
75
76 // --- Pausa para evitar que la consola se cierre ---
77 #ifndef _WIN32
78 ||| system("pause");    // Solo en Windows
79 #else
80 ||| system("read -p 'Presiona Enter para continuar...'");
81 #endif
82
83 ||| return 0; // Fin exitoso
84

```

■ 1 [clangd] ■ 2 0 c ■ 84:1 ■ Final/84

CONCLUSIÓN

El desarrollo del programa nombrado gestor de salario permitió consolidar los conceptos clave de la programación secuencial en lenguaje C, demostrando cómo un algoritmo bien estructurado transforma entradas simples en información valiosa para la gestión financiera. Los puntos críticos abordados incluyen el manejo de codificación UTF-8 para garantizar caracteres especiales, el uso de fgets para capturar nombres completos y la precisión en cálculos flotantes mediante float, asegurando resultados confiables. Un desafío relevante fue ajustar el centro dinámico de texto de la consola, resuelto con el formato %*s en printf, lo que evidenció la importancia de dominar funciones de entrada/salida. Además, la reflexión sobre el porcentaje residual para la sección diversión, destacó la necesidad de validar lógica empresarial en cada solución técnica. Estos aprendizajes refuerzan la relevancia de C como herramienta para modelar problemas reales con rigor metodológico. La actividad no solo cumplió con el objetivo de ingresar algoritmos en contextos prácticos, sino que también subrayó el rol de programación en la optimización de contextos cotidianos. En un mundo en donde la automatización es clave, dominar estos fundamentos es una competencia indispensable para cualquier profesional técnico.

CITAS DE AUTOR

Los algoritmos secuenciales son esenciales para resolver problemas mediante pasos ordenados, como señala el Instituto Politécnico Nacional (IPN, 2020) al destacar su aplicación en la enseñanza de programación básica.

La implementación de algoritmos secuenciales mejora la comprensión lógica en estudiantes de ingeniería, ya que estructuran soluciones sin bifurcaciones complejas (Universidad Nacional Autónoma de México, UNAM, et al., 2018)

«La desventaja de este tipos de algoritmos es su inflexibilidad, no permite un retorno a una secuencia de repetición, que en caso de necesitar se deben escribir las acciones tantas veces como sea necesario.» (García, 2012)

«Desarrollar la inteligencia lógico-matemática en realidad no es una tarea difícil, es una tarea que requiere de retar al cerebro a pensar, de encontrar diversos caminos para resolver un problema y principalmente de elegir el indicado» (Gómez, s.f.)

REFERENCIAS BIBLIOGRÁFICAS

Universidad Autónoma del estado de Hidalgo (s.f.). ALGORITMOS.

De: <https://www.uaeh.edu.mx/scige/boletin/prepa4/n10/e1.html>

Universidad Abierta y a Distancia de México (s.f.). Introducción al lenguaje C. De:

https://dmd.unadmxico.mx/contenidos/DCEIT/Compartidas/FPR/U2/descargables/FPR_U2_Contenido.pdf

UNIVERSIDAD VERACRUZANA (2012). Algoritmos Secuenciales.

De: https://www.uv.mx/personal/clgarcia/files/2012/10/ANTOLOGIA_pag_19_33.pdf