

Actividad | #2 | Diagramas de flujo

Introducción al desarrollo de software

Ingeniería en Desarrollo de Software



TUTOR: Sandra Luz Lara Dévora

ALUMNO: María Fernanda Madrazo Nasuno

FECHA: 4 de Abril del 2025



Índice

<i>Introducción</i>	<i>3</i>
<i>Descripción</i>	<i>4</i>
<i>Justificación</i>	<i>5</i>
<i>Desarrollo (Primos)</i>	<i>6</i>
<i>Desarrollo (Par/Impar)</i>	<i>7</i>
<i>Desarrollo (Al Revés)</i>	<i>8</i>
<i>Conclusión</i>	<i>9</i>
<i>Referencias</i>	<i>10</i>

Introducción

La programación es una herramienta poderosa para resolver problemas cotidianos y educativos. Los tres programas desarrollados en lenguaje C (Primos, Par/Impar y Al Revés) son ejemplos claros de cómo la tecnología puede simplificar conceptos matemáticos. Estos algoritmos no solo automatizan tareas repetitivas, como verificar si un número es primo o clasificar pares e impares, sino que también enseñan a los estudiantes a pensar de manera lógica y estructurada. Al usar operaciones básicas como divisiones y residuos, los programas demuestran que las matemáticas no son abstractas, sino aplicables en entornos digitales. Por ejemplo, un niño que interactúa con la calculadora Al Revés no solo ve un número invertido, sino que comprende cómo cada dígito tiene un valor según su posición. El lenguaje C, aunque sencillo, es ideal para este propósito: su sintaxis clara obliga a los alumnos a descomponer problemas en pasos pequeños, fomentando la atención al detalle. Además, estos programas son accesibles: funcionan en computadoras básicas y no requieren internet, lo que los hace útiles en escuelas con recursos limitados. En resumen, son puentes entre la teoría y la práctica, diseñados para que cualquier persona, sin importar su nivel técnico, pueda explorar el mundo de la programación mientras refuerza conocimientos matemáticos esenciales.

Descripción

Cada programa aborda un desafío matemático específico con una lógica clara. Primos inicia solicitando un número al usuario. Si es menor o igual a 1, se descarta como primo. Para números mayores, un bucle revisa divisores desde 2 hasta la raíz cuadrada del número, optimizando el proceso. Si encuentra un divisor, el programa concluye que no es primo; de lo contrario, lo confirma. Par/Impar utiliza un bucle que se repite 10 veces: en cada iteración, pide un número y aplica el operador módulo (%) para determinar si es par (resto 0) o impar (resto 1). Este enfoque enseña a los estudiantes cómo una operación simple puede resolver problemas recurrentes. Al Revés es más complejo: primero valida que el número tenga 4 dígitos mediante un bucle do-while, asegurando que el usuario ingrese valores correctos. Luego, divide el número en dígitos individuales usando divisiones y residuos (ejemplo: 7631 se descompone en 7, 6, 3 y 1). Finalmente, reconstruye el número invertido multiplicando cada dígito por potencias de 10. Este proceso refuerza el valor posicional de los números y muestra cómo la manipulación matemática puede generar resultados visibles. Cada programa evita funciones avanzadas, priorizando código comprensible para principiantes.

Justificación

Estos programas son relevantes porque integran dos pilares educativos: matemáticas y pensamiento computacional. Al usar lenguaje C, los estudiantes no solo aprenden a codificar, sino que también entienden cómo interactúan la lógica y la aritmética. Por ejemplo, en Primos, el cálculo de la raíz cuadrada para optimizar el bucle enseña eficiencia, un concepto clave en programación. En Par/Impar, el uso del operador módulo (%) demuestra cómo herramientas simples resuelven problemas frecuentes, animando a los alumnos a buscar soluciones creativas. Además, Al Revés va más allá de invertir números: al manipular dígitos con operaciones matemáticas (sin usar cadenas de texto), los estudiantes comprenden que los números son entidades flexibles, no fijas. Estos programas también son inclusivos: al evitar interfaces gráficas complejas, se enfocan en la esencia de los algoritmos, lo que los hace accesibles incluso en comunidades con tecnología limitada. Para los profesores, son recursos versátiles: pueden usarse en ejercicios individuales, competencias en clase o proyectos grupales. En un mundo donde la tecnología avanza rápidamente, enseñar con ejemplos tangibles como estos prepara a los estudiantes para adaptarse a cambios futuros. No se trata solo de escribir código, sino de cultivar habilidades como el razonamiento crítico, la resolución de problemas y la paciencia ante los errores, competencias esenciales en cualquier ámbito profesional.

Desarrollo (Primos)

La primera calculadora deberá de llevar por nombre Primos, y su objetivo será identificar los números primos que se ingresen, por ejemplo, si el usuario ingresa el número 83, deberá imprimir el siguiente mensaje: “El número (número ingresado) si es primo”, en caso de que no sea primo se imprimirá el siguiente mensaje “El número (número ingresado) no es primo”. Básicamente se encargará de identificar si un número es divisible entre 1 y el mismo.

Objetivo: Determinar si un número es primo (solo divisible entre 1 y sí mismo).

Lógica detallada:

- Si el número es 0, 1 o negativo, no es primo.
- Si es 2, es primo (es el único número par primo).
- Para números mayores a 2:

Si se encuentra un divisor, no es primo.

Si no encuentra divisores, es primo

<https://www.programiz.com/online-compiler/3yApbau9VdSB9>

	main.c	Output
	<pre> 1 #include <stdio.h> 2 #include <stdbool.h> 3 4 int main() { 5 char salir = 'N'; 6 int num; 7 8 while (salir != 'S' && salir != 's') { 9 printf("Ingrese un numero: "); 10 scanf("%d", &num); 11 12 bool es_primo = true; 13 14 if (num <= 1) { 15 es_primo = false; 16 } else { 17 for (int i = 2; i <= num / 2; ++i) 18 { 19 if (num % i == 0) { 20 es_primo = false; 21 break; 22 } 23 } 24 25 printf("\n%s primo\n", es_primo ? "Es" 26 : "No es"); 27 while (getchar() != '\n'); 28 29 printf("\nDesea salir (S/N)? "); 30 scanf(" %c", &salir); 31 } 32 33 return 0; 34 } </pre>	<pre> Ingrese un numero: 53 Es primo Desea salir (S/N)? </pre>

Desarrollo (Par/Impar)

La segunda calculadora se llamará Par/Impar, su objetivo es que se ingresen 10 números, ya sean pares o impares, por ejemplo, si se ingresa el número 9, el programa deberá de indicar que es un número impar, pero si se trata del número 2, el programa deberá indicar que se trata de un número par. De 10 números enteros, se debe determinar cuáles son pares y cuáles son impares.

Objetivo: Clasificar 10 números como pares o impares.

Para cada número:

- Divídelo entre 2.
- Si no sobra nada (resto = 0), es par.
- Si sobra 1 (resto = 1), es impar.

<https://www.programiz.com/online-compiler/0EGZFCgW1kxmZ>

main.c	Output
1 #include <stdio.h>	Ingrese un numero: 65
2	65 es impar
3 int main() {	Ingrese un numero: 221
4 int n;	221 es impar
5 int valor = 0;	Ingrese un numero: 23
6 do {	23 es impar
7 printf("Ingrese un numero: ");	Ingrese un numero: 12
8 scanf("%d", &n);	12 es par
9	Ingrese un numero: 464
10 if (n % 2 == 0) {	464 es par
11 printf("%d es par\n", n);	Ingrese un numero: 4523
12 } else {	4523 es impar
13 printf("%d es impar\n", n);	Ingrese un numero: 121
14 }	121 es impar
15 valor = valor + 1;	Ingrese un numero: 454
16 } while (valor < 10);	454 es par
17 printf("***** Termino el ciclo *****\n"	Ingrese un numero: 512
18);	512 es par
19 return 0;	Ingrese un numero: 5121
20 }	5121 es impar
	***** Termino el ciclo *****
	=== Code Execution Successful ===

Desarrollo (Al Revés)

El último programa se llamará Al Revés, su objetivo es que el usuario ingrese un número de 4 dígitos y que sea un número entero, y este programa se encargará de regresar los números al revés o invertidos. Por ejemplo, si se ingresa el número 7631, el programa matemático deberá regresar 1367.

Objetivo: Invertir un número de 4 dígitos.

Separa cada dígito:

- Primer dígito: Miles.
- Segundo dígito: Centenas.
- Tercer dígito: Decenas.
- Cuarto dígito: Unidades.
- Reordénalos al revés: Unidades → Decenas → Centenas → Miles.
- Junta los dígitos invertidos

<https://www.programiz.com/online-compiler/4ABgiKDyjsZYN>

```

main.c
1 #include <stdio.h>
2
3 int main(void) {
4     int n, resto, invertido = 0;
5
6     printf("Ingrese un numero: ");
7     scanf("%d", &n);
8
9     while (n != 0) {
10         resto = n % 10;
11         n = n / 10;
12         invertido = invertido * 10 + resto;
13     }
14
15     printf("El numero invertido es: %d\n",
16           invertido);
17     return 0;
18 }
  
```

Output

```

Ingrese un numero: 5698
El numero invertido es: 8965

=== Code Execution Successful ===
  
```


Conclusión

Los programas Primos, Par/Impar y Al Revés son más que ejercicios técnicos: son herramientas educativas que transforman conceptos abstractos en experiencias interactivas. Al usar lenguaje C, los estudiantes no solo escriben código, sino que descubren cómo las matemáticas se aplican en la programación. Por ejemplo, al invertir un número como 2024 para obtener 4202, entienden que la tecnología no es magia, sino un proceso lógico basado en reglas claras. Estos proyectos también fomentan la autonomía: los alumnos pueden modificarlos para añadir funciones, como contar cuántos primos hay en un rango o invertir números de más dígitos, lo que estimula la creatividad. Además, al enfrentar errores comunes (como ingresar un número inválido en Al Revés), aprenden a depurar y mejorar sus propios códigos, una habilidad vital en cualquier carrera técnica. En contextos con recursos limitados, estos programas demuestran que la innovación educativa no requiere equipos costosos, sino ideas bien estructuradas. Finalmente, representan un paso hacia la democratización de la tecnología: al simplificar conceptos complejos, empoderan a los estudiantes para que no solo consuman tecnología, sino que la entiendan, la critiquen y la creen. En un futuro donde la programación será tan esencial como leer o escribir, iniciativas como estas son semillas para formar generaciones capaces de transformar el mundo con lógica y creatividad.

Referencias

Programiz. (s.f.). Learn to code for free. <https://www.programiz.com/>