

Actividad | #3 | Código en lenguaje c

introducción al desarrollo de software

Ingeniería en Desarrollo de Software



TUTOR: Sandra Lara Debóra

ALUMNO: Jose Antonio Colón Romero

FECHA: 07/04/2024

Indice

Introducción
Descripción
Justificación
Desarrollo
Conclusión
Referencias

Introducción

Tanto los algoritmos y diagramas de flujo son representaciones visuales de los pasos o procesos de un algoritmo o sistema. Se utilizan comúnmente en programación, ingeniería, negocios y otros campos para visualizar el flujo de control de un proceso o sistema. En un diagrama de flujo, cada paso o acción se representa con un símbolo específico, como una forma geométrica, y se conectan mediante flechas que indican la dirección del flujo. Esto ayuda a comprender y comunicar de manera clara y concisa cómo funciona un proceso o algoritmo, facilitando la identificación de problemas, la optimización de procesos y la colaboración entre diferentes equipos o personas. Podría decir que estos diagramas como mapas que muestran los pasos o acciones que se deben seguir para completar una tarea. Utilizan símbolos y flechas para mostrar el orden de las acciones y cómo se relacionan entre sí. Son útiles para entender y comunicar procesos de manera visual y fácil de seguir.

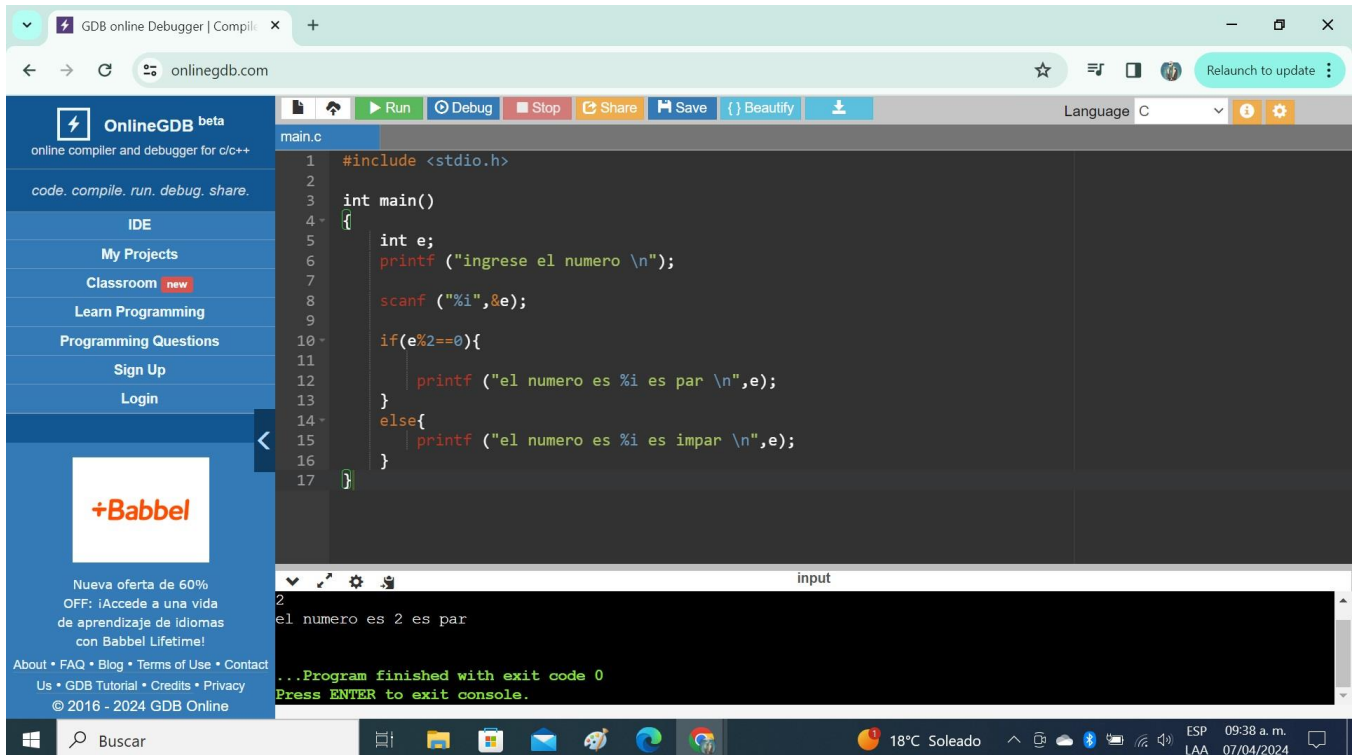
Descripción

Dentro de la actividad se está poniendo en contexto que la empresa MathTech va a necesitar a un ingeniero en desarrollo de software que sea capaz de realizar la tarea de programar tres tipos de calculadoras diferentes para implementar en los colegios y escuelas públicas, por lo tanto, la actividad solicita la realización de 3 calculadoras, la primera debe de identificar a todos los números primos que sean ingresados e indicarlos en un mensaje, por ello lleva el nombre de “primos”. La segunda de nombre á Par/Impar, su objetivo es que se ingresen 10 números, ya sean pares o impares entre los números enteros para determinar los pares e impares. Y una última de nombre á 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. Para los programas que se piden realizar se necesitará el algoritmo de los programas.

Justificación

La siguiente actividad va a dar los resultados finales de estas calculadoras que se requieren en la actividad con los programas cumpliendo su respectiva función para el usuario. Se demostrará la función de las calculadoras y su proceso. Son una manera eficiente de comunicar y mostrarnos procesos que pueden indicar algún error del que se pueda mejorar. Muchas veces inclusive es mas sencillo para las personas identificar los pasos del proceso simplemente de manera visual y así tienen una mejor oportunidad de realizar tareas y procesos de modo.

Desarrollo “Par/Impar”

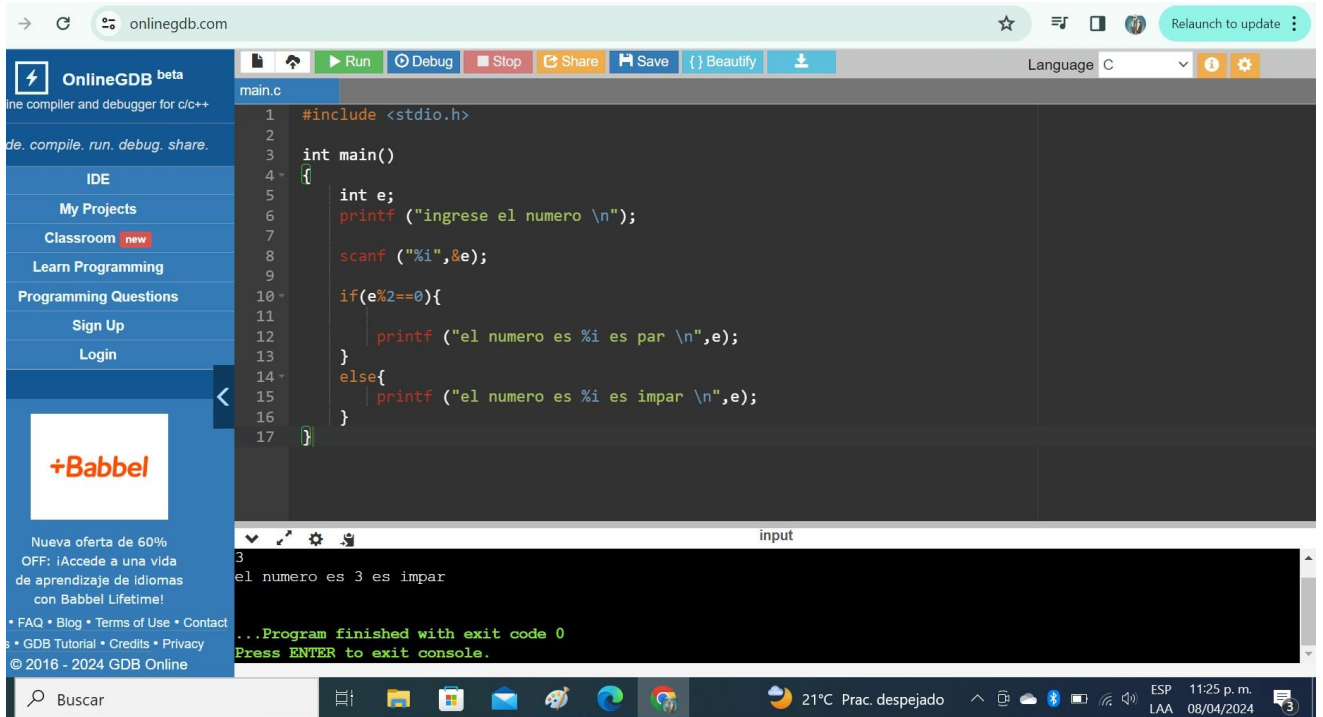


The screenshot shows the OnlineGDB web IDE interface. The left sidebar contains navigation links: IDE, My Projects, Classroom (new), Learn Programming, Programming Questions, Sign Up, and Login. The main editor area displays a C program named 'main.c' with the following code:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int e;
6     printf ("ingrese el numero \n");
7
8     scanf ("%i",&e);
9
10    if(e%2==0){
11        printf ("el numero es %i es par \n",e);
12    }
13    else{
14        printf ("el numero es %i es impar \n",e);
15    }
16 }
17
```

The console output shows the program running successfully with the input '2' and the output 'el numero es 2 es par'. The console also displays the message '...Program finished with exit code 0' and 'Press ENTER to exit console.'.

En la siguiente captura de la calculadora Par/Impar se introdujo un número par.



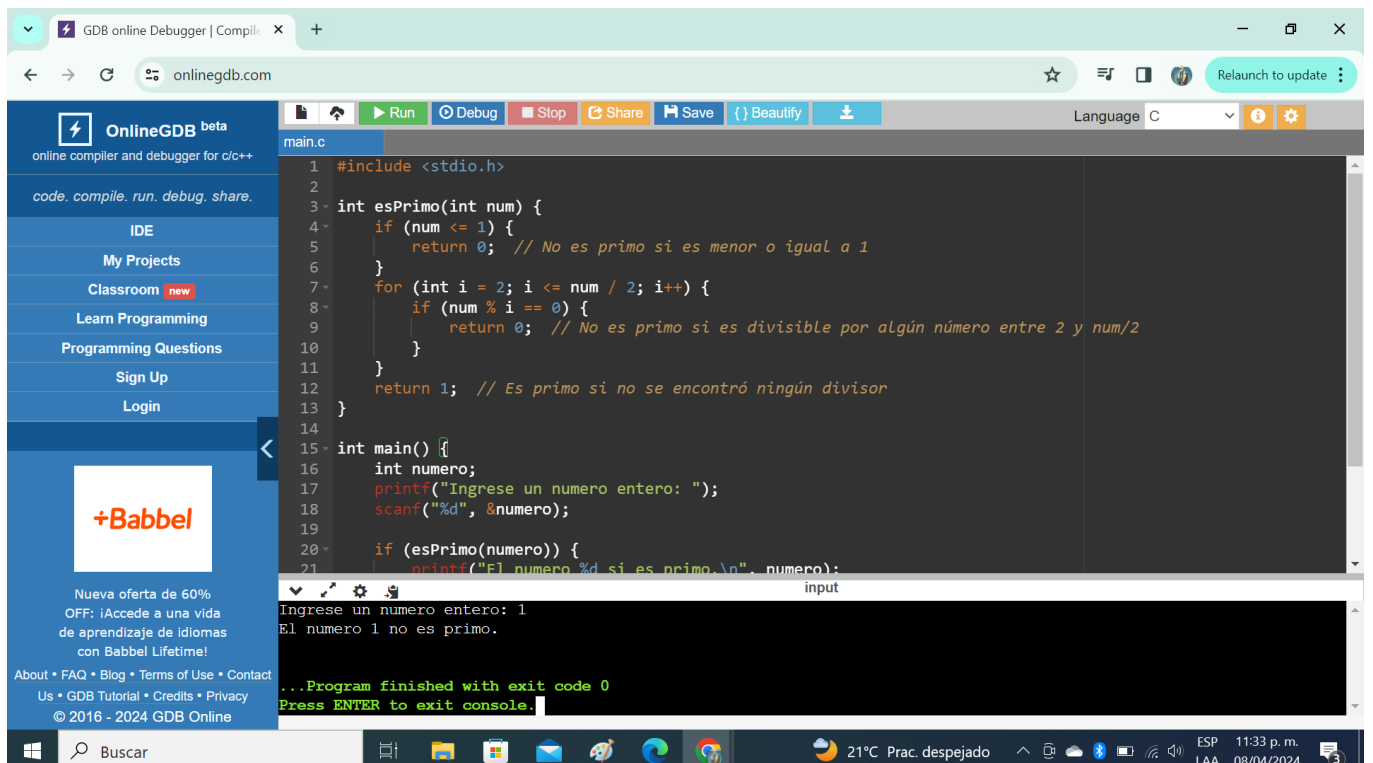
The screenshot shows the OnlineGDB IDE interface. On the left, there's a sidebar with navigation links: IDE, My Projects, Classroom (marked 'new'), Learn Programming, Programming Questions, Sign Up, and Login. Below these is a Babbel advertisement. The main area displays a C program in 'main.c' that checks if a number is even or odd. The code is as follows:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int e;
6     printf("ingrese el numero \n");
7
8     scanf ("%i",&e);
9
10    if(e%2==0){
11        printf ("el numero es %i es par \n",e);
12    }
13    else{
14        printf ("el numero es %i es impar \n",e);
15    }
16 }
17
```

The console output shows the program execution: 'el numero es 3 es impar' and '...Program finished with exit code 0'. The input field at the bottom shows '3'.

En esta captura se introduce el número “3” y se nos arroja el resultado.

Desarrollo “Primos”

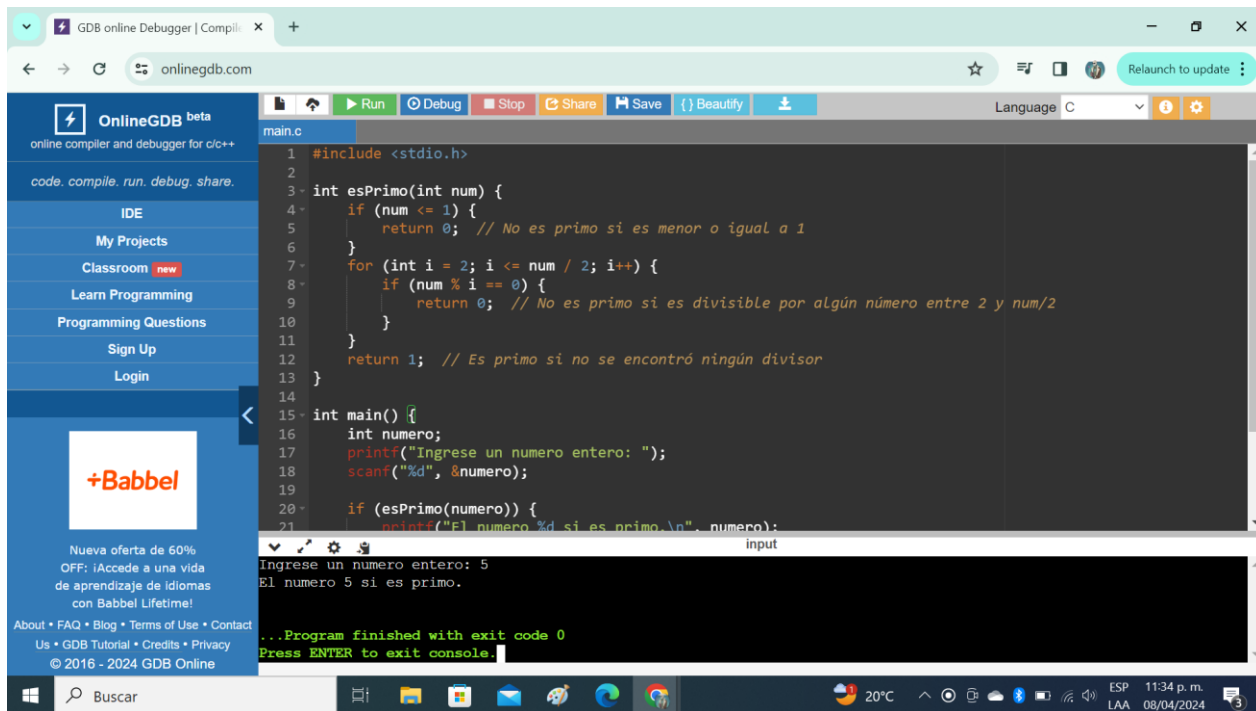


The screenshot shows the OnlineGDB IDE interface with a C program that checks if a number is prime. The code is as follows:

```
1 #include <stdio.h>
2
3 int esPrimo(int num) {
4     if (num <= 1) {
5         return 0; // No es primo si es menor o igual a 1
6     }
7     for (int i = 2; i <= num / 2; i++) {
8         if (num % i == 0) {
9             return 0; // No es primo si es divisible por algún número entre 2 y num/2
10        }
11    }
12    return 1; // Es primo si no se encontró ningún divisor
13 }
14
15 int main() {
16     int numero;
17     printf("Ingrese un numero entero: ");
18     scanf("%d", &numero);
19
20     if (esPrimo(numero)) {
21         printf("El numero %d si es primo.\n", numero);
22     }
23 }
```

The console output shows the program execution: 'Ingrese un numero entero: 1' and 'El numero 1 no es primo.' The input field at the bottom shows '1'.

En este caso se muestra el número “1” ingresado el cual nos dice que no es primo.



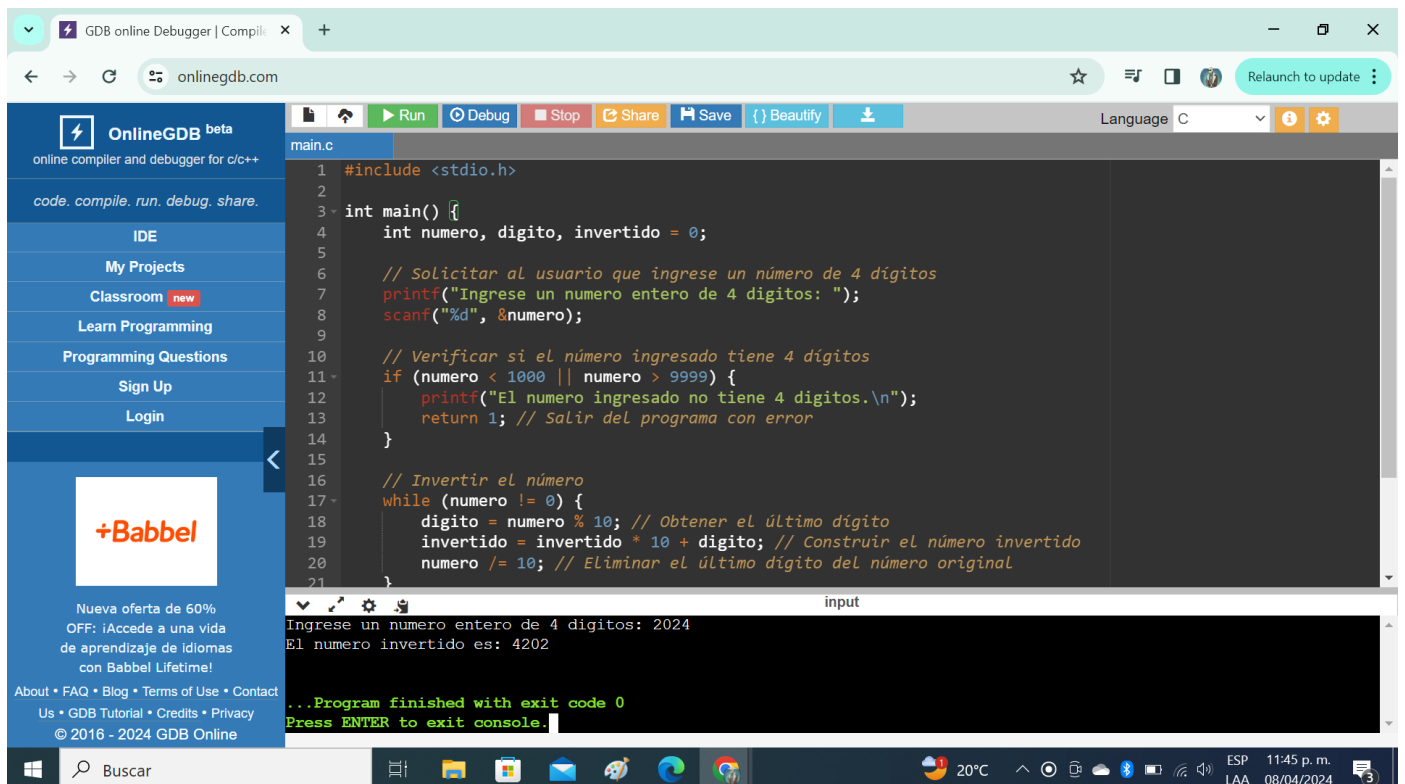
The screenshot shows the OnlineGDB interface with a C program that checks if a number is prime. The code is as follows:

```
1 #include <stdio.h>
2
3 int esPrimo(int num) {
4     if (num <= 1) {
5         return 0; // No es primo si es menor o igual a 1
6     }
7     for (int i = 2; i <= num / 2; i++) {
8         if (num % i == 0) {
9             return 0; // No es primo si es divisible por algún número entre 2 y num/2
10        }
11    }
12    return 1; // Es primo si no se encontró ningún divisor
13 }
14
15 int main() {
16     int numero;
17     printf("Ingrese un numero entero: ");
18     scanf("%d", &numero);
19
20     if (esPrimo(numero)) {
21         printf("El numero %d si es primo.\n", numero);
22     }
23 }
```

The console output shows the user entering 5 and the program responding: "El numero 5 si es primo." The program finished with exit code 0.

Se ingresa el número “5” el cuál sí es un número primo

Desarrollo “Al Revés”.



The screenshot shows the OnlineGDB interface with a C program that reverses a 4-digit number. The code is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int numero, digito, invertido = 0;
5
6     // Solicitar al usuario que ingrese un número de 4 dígitos
7     printf("Ingrese un numero entero de 4 digitos: ");
8     scanf("%d", &numero);
9
10    // Verificar si el número ingresado tiene 4 dígitos
11    if (numero < 1000 || numero > 9999) {
12        printf("El numero ingresado no tiene 4 digitos.\n");
13        return 1; // Salir del programa con error
14    }
15
16    // Invertir el número
17    while (numero != 0) {
18        digito = numero % 10; // Obtener el último dígito
19        invertido = invertido * 10 + digito; // Construir el número invertido
20        numero /= 10; // Eliminar el último dígito del número original
21    }
22 }
```

The console output shows the user entering 2024 and the program responding: "El numero invertido es: 4202". The program finished with exit code 0.

En este caso se introduce el número 2024 y nos da como resultado el número revés.

Conclusión

En esta actividad se realiza el proceso de cómo funcionan las calculadoras que se requieren en la actividad programadas en lenguaje C. Programar en C es importante por su eficiencia, portabilidad, flexibilidad y su papel fundamental en el desarrollo de sistemas operativos y otros lenguajes de programación.

Referencias:

<https://www.onlinegdb.com/>