



Ingeniería en Desarrollo de Software

Actividad 3 Código en Lenguaje C

Introducción al Desarrollo de Software

Ingeniería en Diseño de Software

Tutor: Sandra Lara

Alumna: Araceli Marlenee Gutiérrez Castillo

Fecha: 08 de abril del 2024

INDICE

INTRODUCCIÓN-----	2
DESCRIPCIÓN-----	2
JUSTIFICACIÓN -----	3
CÓDIGO PRIMOS -----	4
CÓDIGO PAR / IMPAR -----	5
CÓDIGO AL REVES -----	6
CONCLUSIONES-----	6

INTRODUCCIÓN

Para finalizar nuestras actividades debemos realizar los códigos correspondientes en el lenguaje C. la codificación es el proceso para traducir lo ya trabajado anteriormente, es decir, los algoritmos y los diagramas de flujo, este básicamente ya es el proceso donde las ideas se convierten en las instrucciones que la computadora ejecutara para poder darle solución a los problemas planteados. De esta manera se pueden resolver y automatizar tareas específicas. El código debe ser fácil de entender para cualquier persona hasta para el autor mismo en un futuro, debe tener las variables, funciones y clases muy claras y concisas para que pueda ser legible de manera óptima, este debe ejecutarse eficientemente y evitar tener repeticiones innecesarias, su organización debe ser lógica y coherente, ya que quizá en un futuro se puede utilizar todo o parte de este o ser compartido con otros programadores.

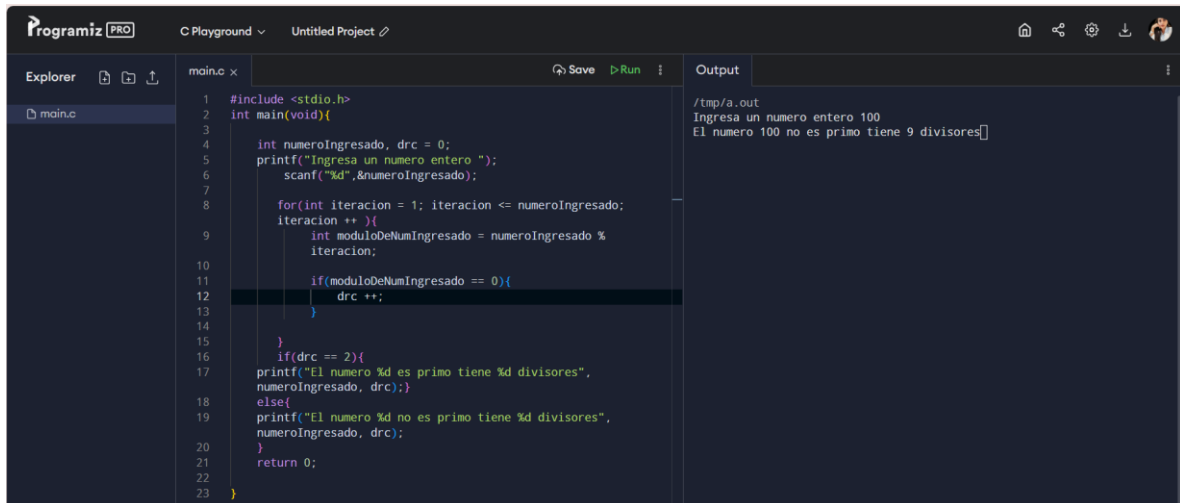
DESCRIPCIÓN

Se realizarán 3 códigos, uno por cada calculadora, se analizarán las distintas variables que están involucradas en nuestro problema; para esto evaluaremos las características que cada calculadora deberá tener para que funcione de manera correcta. Usaremos los algoritmos y los diagramas, estos nos ayudaran a visualizar la lógica que tendrá el código, nos ayudará a verificar que no existan ningún tipo de errores o fallas en la lógica del programa antes de comenzar a escribir el código. De esta manera podemos mejorar la calidad y eficiencia del código. Los códigos que realizaremos son en el lenguaje C que tiene eficiencia y flexibilidad, está organizado en funciones y cada una debe tener su nombre, la función principal es el punto de entrada en el código. Se usarán variables que son las que almacenaran los valores y estas pueden ir cambiando a lo largo del código, en estas es importante mencionar que se deben declarar que tipo de variable será la que se usara para que el código no tenga errores al momento de compilarse.

JUSTIFICACIÓN

El código es fundamental en la programación porque es el lenguaje que indica a la computadora las acciones que debe realizar. Es el proceso mediante el cual los algoritmos y los diagramas de flujo se convierten en un software funcional. Son las instrucciones que el programador da para que la computadora pueda realizar tareas específicas, resuelva problemas. Un código bien elaborado facilita la identificación de errores y solución de estos. El código debe ser eficiente en términos de uso de memoria y tiempo de ejecución, debe estar dividido en módulos o funciones que realicen tareas específicas y a su vez puedan ser reutilizables a futuro. De igual manera este debe ser fácil de actualizar a medida que vayan cambiando los requisitos del proyecto, para ello, es importante ir documentando adecuadamente el código y utilizar herramientas de control de versiones para que este pueda ser portátil a otras plataformas de sistemas operativos siempre y cuando sea posible.

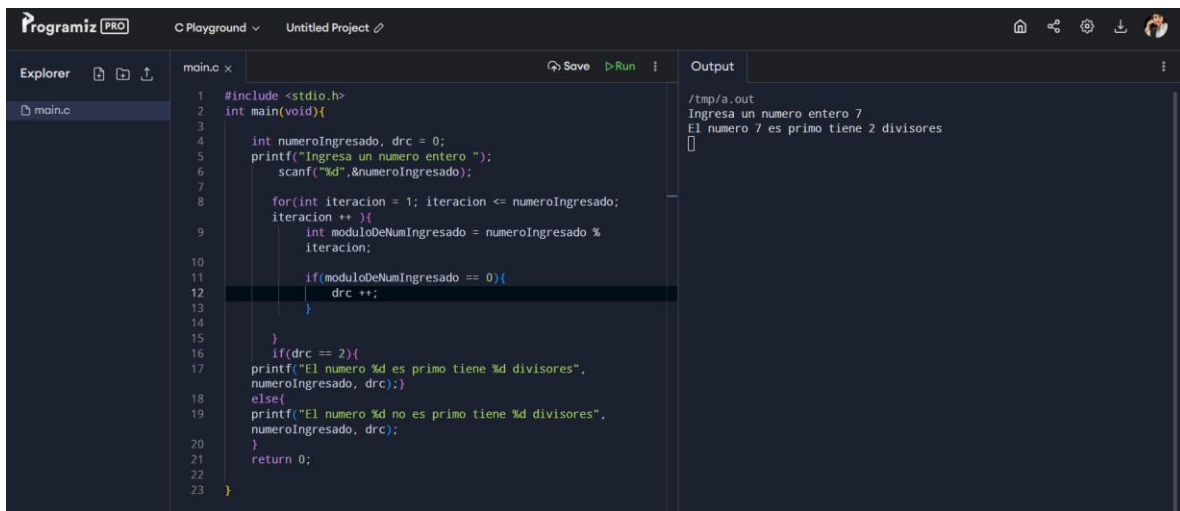
Figura número 1
Código de Números Primos



```
1 #include <stdio.h>
2 int main(void){
3
4     int numeroIngresado, drc = 0;
5     printf("Ingresa un numero entero ");
6     scanf("%d",&numeroIngresado);
7
8     for(int iteracion = 1; iteracion <= numeroIngresado;
9         iteracion ++ ){
10         int moduloDeNumIngresado = numeroIngresado %
            iteracion;
11
12         if(moduloDeNumIngresado == 0){
13             drc ++;
14         }
15     }
16     if(drc == 2){
17         printf("El numero %d es primo tiene %d divisores",
            numeroIngresado, drc);
18     }
19     else{
20         printf("El numero %d no es primo tiene %d divisores",
            numeroIngresado, drc);
21     }
22     return 0;
23 }
```

Output: /tmp/a.out
Ingresa un numero entero 100
El numero 100 no es primo tiene 9 divisores

La imagen nos muestra el código compilado de números primos, donde nos indica que el número ingresado No es primo.

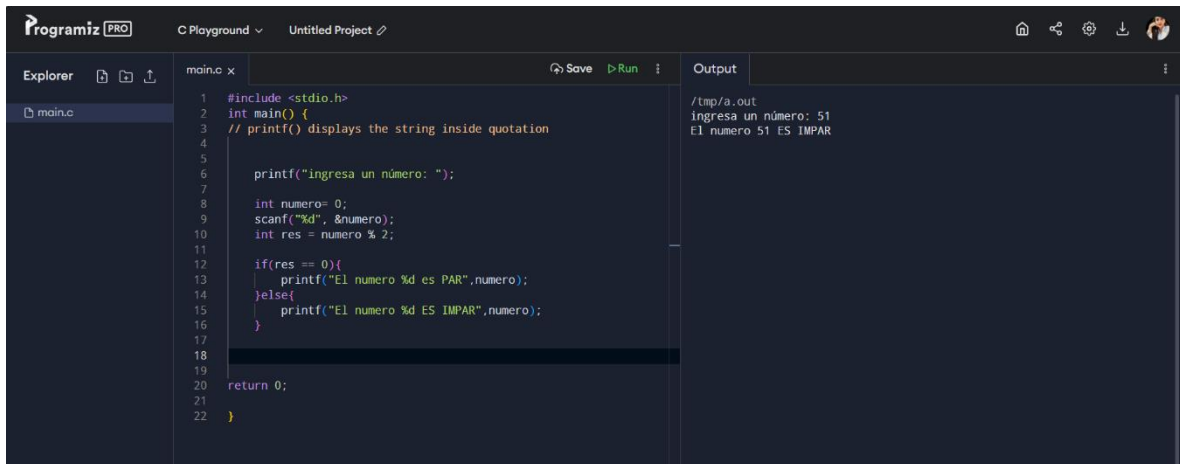


```
1 #include <stdio.h>
2 int main(void){
3
4     int numeroIngresado, drc = 0;
5     printf("Ingresa un numero entero ");
6     scanf("%d",&numeroIngresado);
7
8     for(int iteracion = 1; iteracion <= numeroIngresado;
9         iteracion ++ ){
10         int moduloDeNumIngresado = numeroIngresado %
            iteracion;
11
12         if(moduloDeNumIngresado == 0){
13             drc ++;
14         }
15     }
16     if(drc == 2){
17         printf("El numero %d es primo tiene %d divisores",
            numeroIngresado, drc);
18     }
19     else{
20         printf("El numero %d no es primo tiene %d divisores",
            numeroIngresado, drc);
21     }
22     return 0;
23 }
```

Output: /tmp/a.out
Ingresa un numero entero 7
El numero 7 es primo tiene 2 divisores

La imagen nos muestra el código compilado de números primos, donde nos indica que el número ingresado Si es primo.

Figura número 2
Código Par / Impar



```
#include <stdio.h>
int main() {
    // printf() displays the string inside quotation
    printf("Ingresa un número: ");

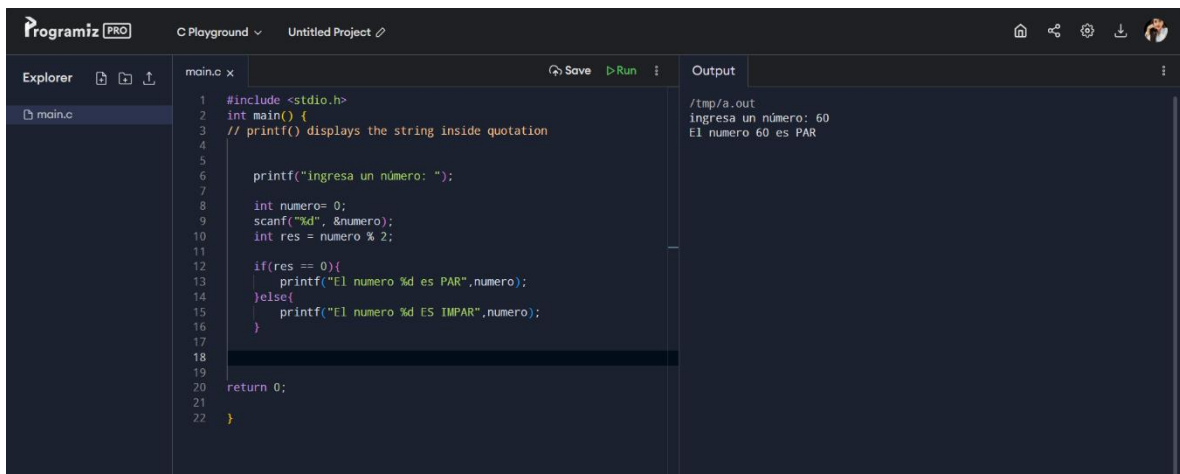
    int numero= 0;
    scanf("%d", &numero);
    int res = numero % 2;

    if(res == 0){
        printf("El numero %d es PAR",numero);
    }else{
        printf("El numero %d ES IMPAR",numero);
    }

    return 0;
}
```

Output: /tmp/a.out
Ingresa un número: 51
El numero 51 ES IMPAR

La imagen nos muestra el código compilado de números Pares e impares, donde nos indica que el número ingresado es un número Impar



```
#include <stdio.h>
int main() {
    // printf() displays the string inside quotation
    printf("Ingresa un número: ");

    int numero= 0;
    scanf("%d", &numero);
    int res = numero % 2;

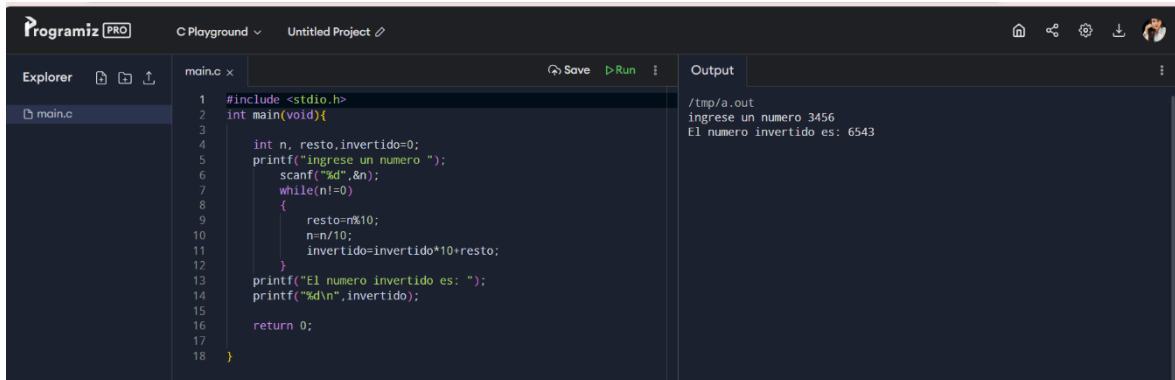
    if(res == 0){
        printf("El numero %d es PAR",numero);
    }else{
        printf("El numero %d ES IMPAR",numero);
    }

    return 0;
}
```

Output: /tmp/a.out
Ingresa un número: 60
El numero 60 es PAR

La imagen nos muestra el código compilado de números pares e impares, donde nos indica que el número ingresado es Par.

Figura número 3
Código Al revés



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

Output

```
/tmp/a.out
ingrese un numero 3456
El numero invertido es: 6543
```

La imagen nos muestra el código compilado de números al revés, donde nos dará por resultado el numero invertido del que fue ingresado. En este caso fue el número 3456 y al ejecutar el código nos arrojó por resultado 6543.

CONCLUSIÓN

El código es fundamental en la programación porque es el lenguaje que indica a la computadora las acciones que debe realizar. Es el proceso mediante el cual los algoritmos y los diagramas de flujo se convierten en un software funcional. Son las instrucciones que el programador da para que la computadora pueda realizar tareas específicas, resuelva problemas. El código debe ser fácil de entender para cualquier persona hasta para el autor mismo en un futuro, debe tener las variables, funciones y clases muy claras y concisas para que pueda ser legible de manera óptima, este debe ejecutarse eficientemente y evitar tener repeticiones innecesarias, su organización debe ser lógica y coherente, ya que quizá en un futuro se puede utilizar todo o parte de este o ser compartido con otros programadores. Con todo esto aprendido podemos llevarlo a implementar en nuestra vida cotidiana para poder desenvolvemos en nuestra área laboral y ayudar hacer en nuestro trabajo mejoras que puedan contribuir para llegar al éxito que buscamos.