01 DE AGOSTO DEL 2025

ACTIVIDAD # 3 CODIGO EN LENGUAJE C

INTRODUCCION AL DESARROLLO DE SOFTWARE

Ingeniería en desarrollo de Software

ALUMNO: GERMAN NAHUAT NAHUAT

academia global

Índice

Introducción

Ya hemos estado aprendiendo sobre algoritmos y los procesos de ir haciéndolo, al igual la forma grafica de hacerlo, representado mediante un diagrama de flujo en donde se representa de manera grafica todo el proceso a realizar para que un algoritmo de la solución planteada.

En esta actividad vamos a ir acoplando un algoritmo en u programa en lenguaje c, definiendo las variables y las constantes a utilizar en las algebras booleanas, asi podemos ver en que momento usar una variable global y una variable local, para que podamos ir dándole una estructura al lenguaje c. igual tenemos que aplicar lo aprendido sobre los tipos de datos si son simples o estructurados identificando los caracteres que se iran usando.

La creación de lenguaje c se vuelve sencillo siguiendo cada paso que hemos aprendido. La librería stdio.h permite ingresar valores de entrada/salida, igual hay que saber identificar las diferentes funciones de la librería previamente visto, llevaremos acabo el uso de los operadores y las expresiones que se han ido aprendido para la correcta creación de un lenguaje c.

Descripción

La empresa MathTech requiere 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:

- 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.
- 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.
- 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.

Ya hemos desarrollado los algoritmos y los diagramas de flujo de cada uno de los problemas que se nos plantea en la actividad, ahora es momento de hacer la codificación en lenguaje c de cada uno, veamos los pasos a seguir al igual corroborar que lo que haceos funcione como es planteado.

Justificación

La utilización de los algoritmos en la vida diaria es a menudo como se ha estado analizando a lo largo de esta materia, igual ya vino que los algoritmos es parte de nuestra vida y lo podemos simplificar en varias formas, ahora en el lenguaje de programación también se puede hacer mediante el método de lenguaje c, ya que atreves de ello podemos hacer varias programaciones y también estructurarlo de la mejor manera identificando cada factor a utilizar para que al momento de correr en un programa pueda dar el resultado programado.

La dominación de las funciones de stdio.h ayuda para que podamos realizar una buena programación, estas funciones son: GETCHAR, PUTCHAR, SCANF, PRINTF, GETS Y PUTS.

Estas funciones permiten transferir información de una computadora a los dispositivos de entrada/salida estándar como un teclado o un monitor.

Desarrollo

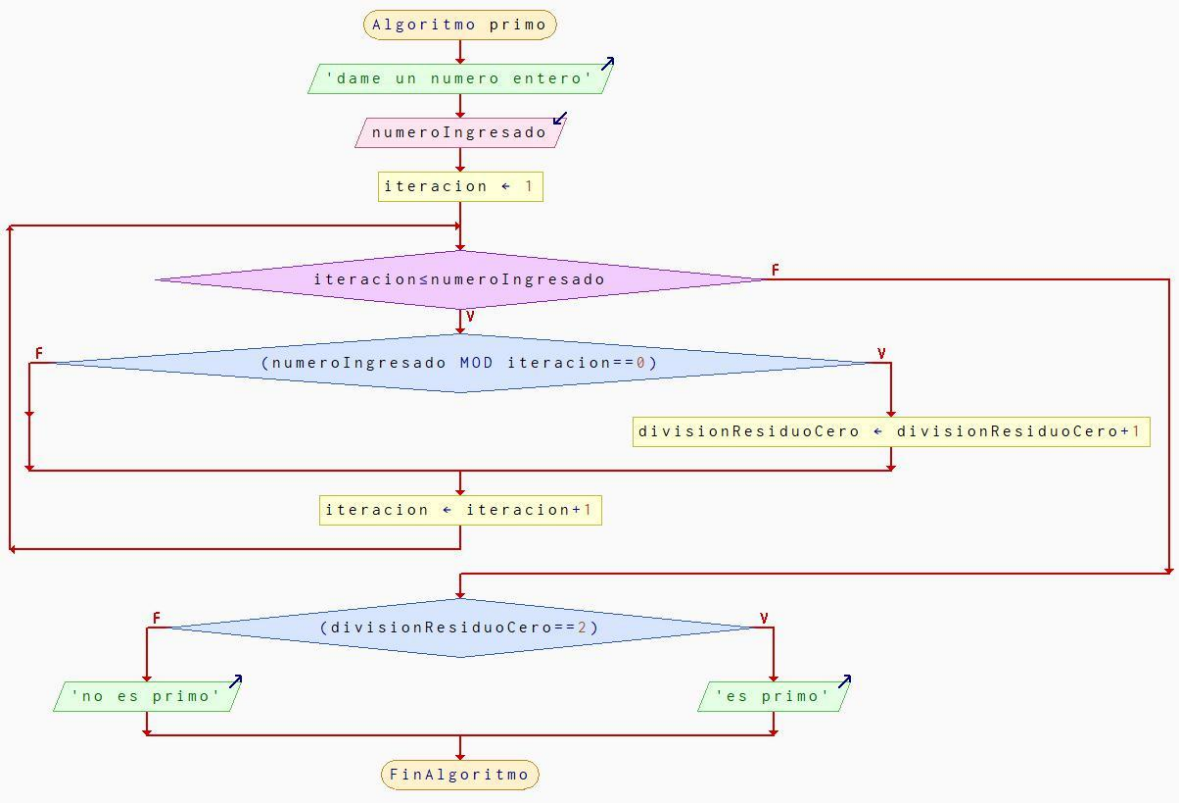
La primera calculadora deberá de llevar por nombre Primos, y su objetivo será identificar los números primos que se ingresen, por ejemplo, si el usuario ingresa el número 83, deberá imprimir el siguiente mensaje: “El número (número ingresado) si es primo”, en caso de que no sea primo se imprimirá el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

1. Definir algoritmo primo
2. Escribir número entero.
3. Leer número ingresado
4. Generar la viable de iteración.
5. Definir ciclo mientras.
6. Definir la condición de iteración.
7. Definir la división a realizar para determinar si es número primo
8. Si el residuo de la división es igual a 2 entonces es primo.
9. Si el residuo de la división es diferente de 2 entonces no es primo.
10. Fin del algoritmo

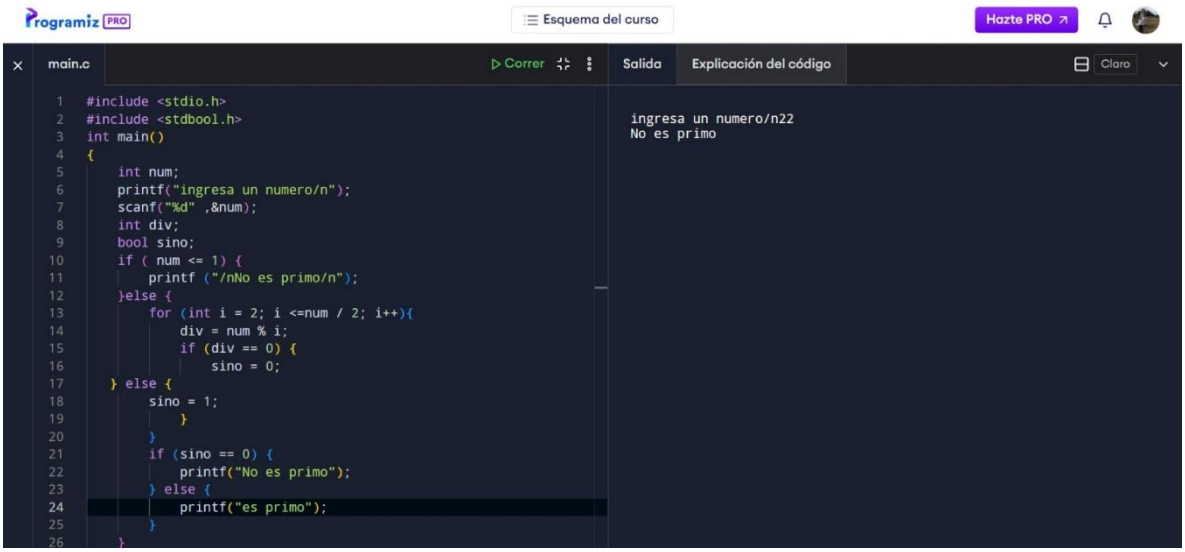
Veamos la secuencia de algoritmos ya ingresados en el programa que usamos.

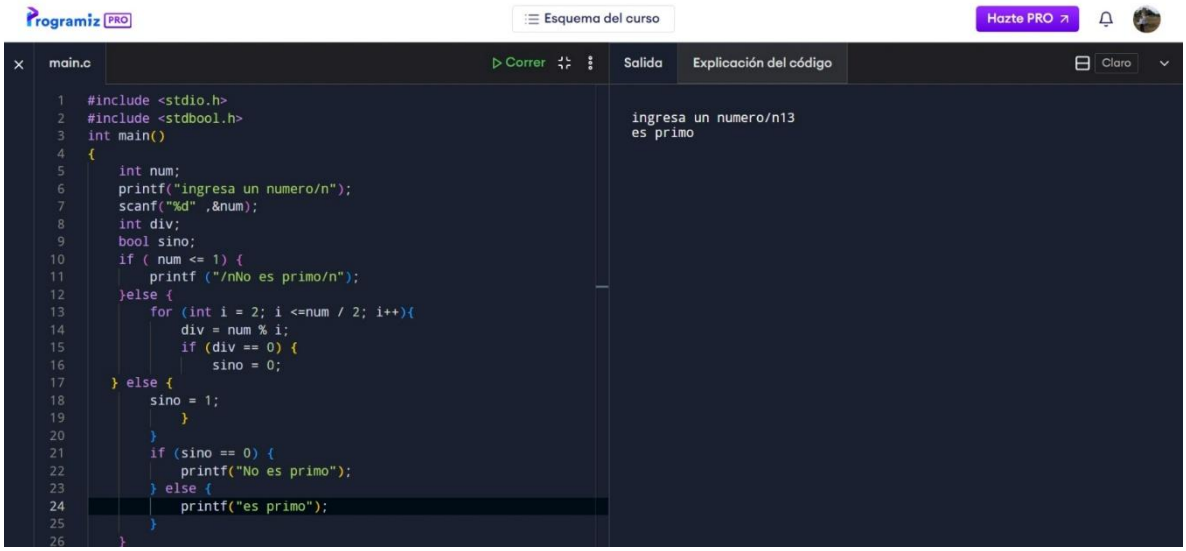
- Algoritmo primo
- Escribir “dame un numero entero”
- Leer numeroIngresado;
- 6teración=1;
- Mientras 6teración <= numeroIngresado Hacer
- si(numeroIngresado%iteracion==0) Entonces
- divisionResiduoCero=divisionResiduoCero+1;
- FinSi
- 6teración=6teración+1
- FinMientras
- si(divisionResiduoCero==2)Entonces
- Escribir “es primo”
- SiNo
- Escribir “no es primo”
- FinSi
- FinAlgoritmo

Este es la representación grafica de nuestro algoritmo para definir si un numero es Primo o No Primo.



En este diagrama se ve donde se ingresa el numero entero al igual en la secuencia se ve la iteración si es falso se pasa directamente a la división de residuo donde termina f si es NO PRIMO y verdadero si es PRIMO, pero si la iteración es verdadero entra al mod iteración donde el resultado ya sea falso o verdadero regresa de nuevo a la iteración hasta tener el resultado deseado.


Programiz PRO
Esquema del curso
Hazte PRO
main.c
Correr
Salida
Explicación del código
Claro
#include <stdio.h>
#include <stdbool.h>
int main()
{
int num;
printf("ingresa un numero/n");
scanf("%d" ,&num);
int div;
bool sino;
if (num <= 1) {
printf ("/nNo es primo/n");
}else {
for (int i = 2; i <=num / 2; i++){
div = num % i;
if (div == 0) {
sino = 0;
} else {
sino = 1;
}
}
if (sino == 0) {
printf("No es primo");
} else {
printf("es primo");
}
}
ingresa un numero/n22
No es primo


Programiz PRO
Esquema del curso
Hazte PRO
main.c
Correr
Salida
Explicación del código
Claro
#include <stdio.h>
#include <stdbool.h>
int main()
{
int num;
printf("ingresa un numero/n");
scanf("%d" ,&num);
int div;
bool sino;
if (num <= 1) {
printf ("/nNo es primo/n");
}else {
for (int i = 2; i <=num / 2; i++){
div = num % i;
if (div == 0) {
sino = 0;
} else {
sino = 1;
}
}
if (sino == 0) {
printf("No es primo");
} else {
printf("es primo");
}
}
ingresa un numero/n13
es primo

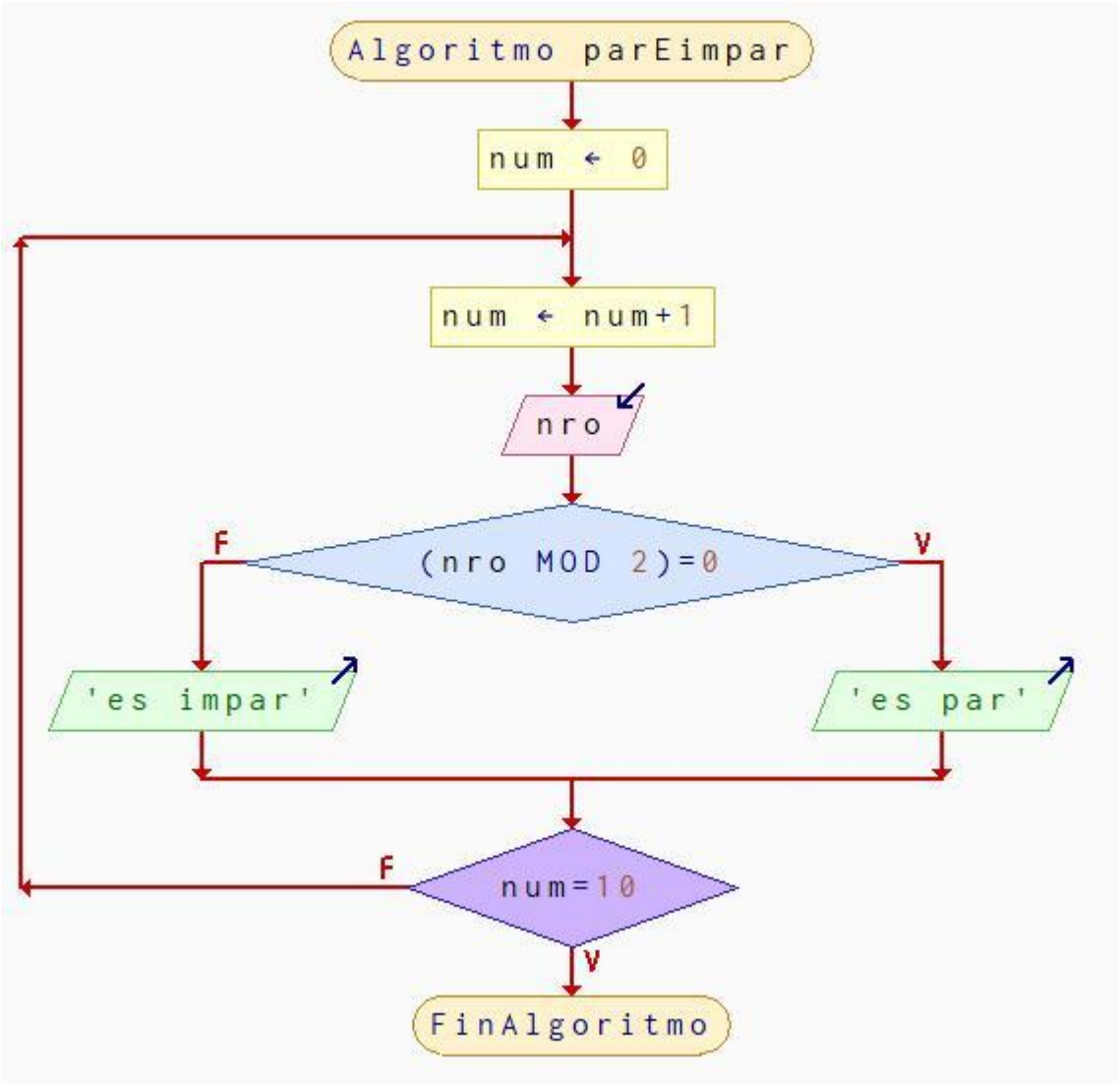
La segunda calculadora se llamará Par/Impar, su objetivo es que se ingresen 10 números, ya sean pares o impares, por ejemplo, si se ingresa el número 9, el programa deberá de indicar que es un número impar, pero si se trata del número 2, el programa deberá indicar que se trata de un número par. De 10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

1. Definir algoritmo
2. Establecer la variable num para los números
3. Establecer el ciclo de repeticiones
4. Establecer las iteraciones del ciclo
5. Leer numero
6. Guardar una variable, ya que hará la función a cualquier numero
7. Establecer la condición.
8. Si es divisible entre 2 y el residuo o resultado es igual a cero(0), ese numero es par.
9. Si se divide entre 2 y el residuo no es igual o mayor a cero(0), ese numero es impar.
10. Fin de algoritmo.

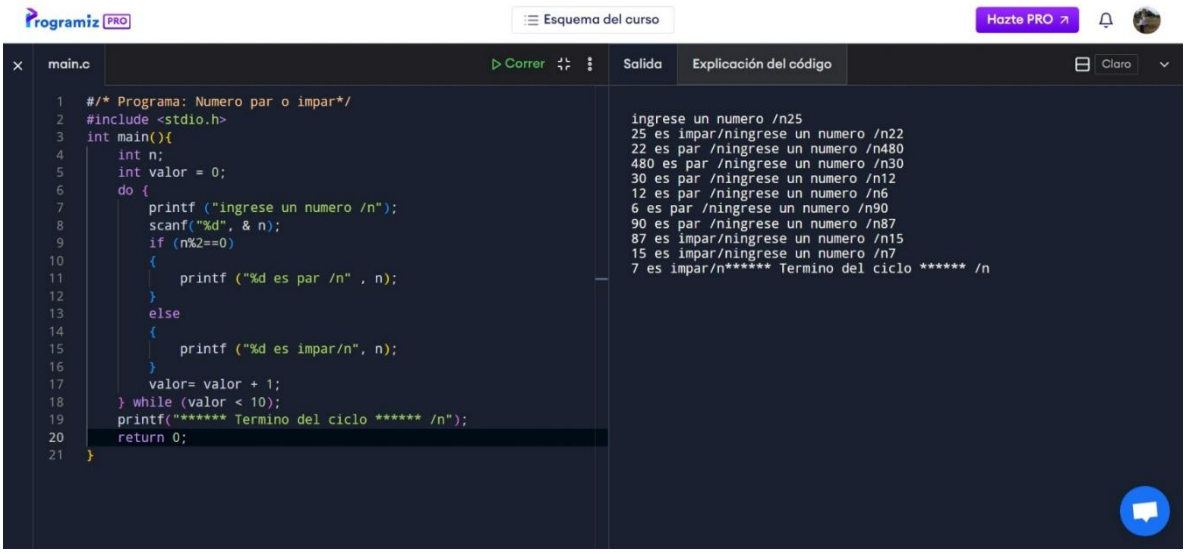
Veamos la secuencia de algoritmos ya ingresados en el programa que usamos.

- Algoritmo parEimpar
- num=0;
- Repetir
- num=num+1;
- leer nro;
- si (nro mod 2)=0 Entonces
- Escribir “es par”
- SiNo
- Escribir “es impar”
- FinSi
- Hasta Que num=10
- FinAlgoritmo

Esta es la secuencia grafica de nuestro algoritmo para definir un numero Par e Impar, igual vemos toda la secuencia para que nos pueda dar el resultado solicitado.



En este diagrama de flujo podemos identificar el seguimiento del algoritmo, como la variable num y el ciclo definido al momento de ingresar el número hace que el algoritmo define si es par o no es par un número siguiendo si es falso de IMPAR, pero si es verdadero es PAR, al igual si sale un error vuelve a regresar al inicio para hacer el mismo procedimiento.


Programiz PRO
Esquema del curso
Hazte PRO
main.c
Correr
Salida
Explicación del código
Claro
#/* Programa: Numero par o impar*/
#include <stdio.h>
int main(){
int n;
int valor = 0;
do {
printf ("ingrese un numero /n");
scanf("%d", & n);
if (n%2==0)
{
printf ("%d es par /n" , n);
}
else
{
printf ("%d es impar/n", n);
}
valor= valor + 1;
} while (valor < 10);
printf("****** Termino del ciclo ****** /n");
return 0;
}
ingrese un numero /n25
25 es impar/ningrese un numero /n22
22 es par /ningrese un numero /n480
480 es par /ningrese un numero /n30
30 es par /ningrese un numero /n12
12 es par /ningrese un numero /n6
6 es par /ningrese un numero /n90
90 es par /ningrese un numero /n87
87 es impar/ningrese un numero /n15
15 es impar/ningrese un numero /n7
7 es impar/n****** Termino del ciclo ****** /n

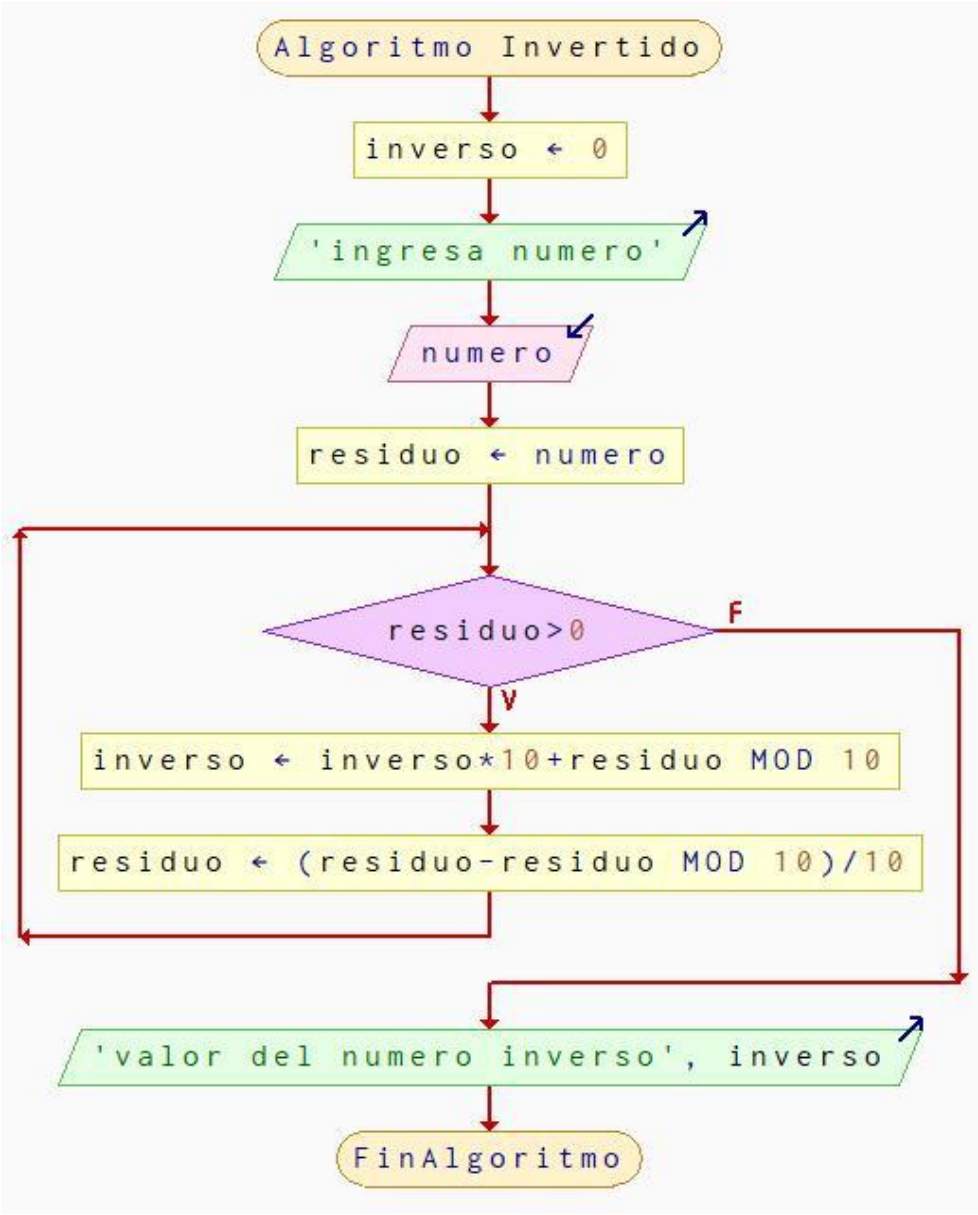
El último programa se llamará Al Revés, su objetivo es que el usuario ingrese un número de 4 dígitos y que sea un número entero, y este programa se encargará de regresar los números al revés o invertidos. Por ejemplo, si se ingresa el número 7631, el programa matemático deberá regresar 1367.

1. Definir algoritmo invertido
2. Asignación inversa a cero.
3. Escribir ingresar número.
4. Leer número.
5. Definimos secuencia.
6. Residuo del número mayor que cero.
7. Definimos ciclo
8. Ingresamos inverso por 10 más residuo mod 10
9. Residuo menos mod 10 dividido entre 10
10. Fin de ciclo
11. Escribir valor inverso
12. Fin del algoritmo.

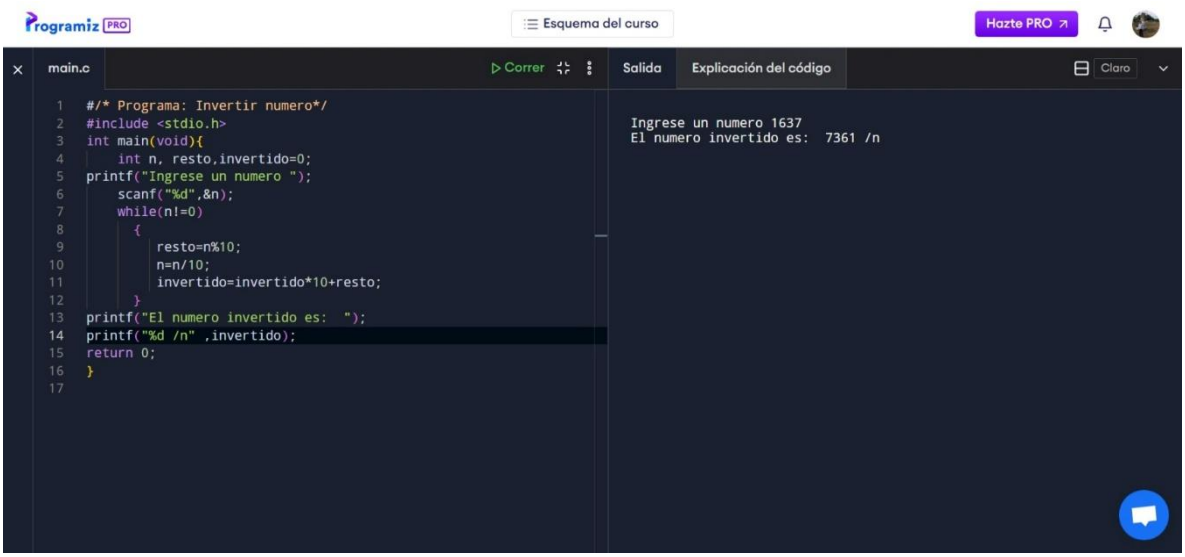
Veamos la secuencia de algoritmos ya ingresados en el programa que usamos.

- Algoritmo Invertido
- inverso <-0
- Escribir “ingresa numero”
- Leer numero
- residuo <- numero
- mientras residuo > 0 Hacer
- inverso <- inverso * 10 + residuo mod 10;
- residuo <- (residuo – residuo mod 10) / 10;
- FinMientras
- Escribir “valor del numero inverso”,inverso;
- FinAlgoritmo

Nuestro algoritmo de numeración invertido se muestra de la siguiente forma, contemplando paso a paso el algoritmo señalado para poder brindar el resultado programado.



En este algoritmo invertido vemos como al ingresar el número, desde el algoritmo realiza la división donde el residuo determina si es falso da el número inverso al solicitado, pero en caso de ser verdadero ingresa a las operaciones donde el residuo cumple con lo solicitado y regresa de nuevo para determinar el resultado y poder cumplir con el algoritmo.


Programiz PRO
Esquema del curso
Hazte PRO
main.c
Correr
Salida
Explicación del código
Claro
#/* Programa: Invertir numero*/
#include <stdio.h>
int main(void){
int n, resto,invertido=0;
printf("Ingrese un numero ");
scanf("%d",&n);
while(n!=0)
{
resto=n%10;
n=n/10;
invertido=invertido*10+resto;
}
printf("El numero invertido es: ");
printf("%d /n" ,invertido);
return 0;
}
Ingrese un numero 1637
El numero invertido es: 7361 /n

Conclusión

Esta materia ayudo a conocer el lenguaje c para programación, al igual que aprendimos que los algoritmos es parte de nuestra vida diaria, tanto en actividades y como es que un algoritmo participa en nuestra vida de manera diaria, igual analizamos que los algoritmos son finitos, tienen un inicio y un final, y se hace de manera ordenada, lo podemos representar gráficamente donde podemos ver los pasos que se realiza las veces que sea necesario para que obtengamos la solución a un problema planteado como en el transcurso de la actividad presentada y que nuestra introducción al desarrollo de software es mediante un lenguaje c, que es básico y nos ayudara a comprender los lenguajes de programación que veremos en materias posteriores.

La estructuración de cada uno de los pasos para un algoritmo nos lleva a que nuestro diagrama de flujo sea mas entendible y con ello permita presentar una información y que las personas logren entenderlo para poder desarrollar la programación en lenguaje c.

Enlace Git Hub

https://github.com/German-Nahuat/Repositorio.git