

Actividad | 3 | Códigos en Lenguaje C

Introducción al Desarrollo del Software

Ingeniería en Desarrollo de Software



TUTOR: Sandra Luz Lara Dévora

ALUMNO: Adriana Esteban López

FECHA: 24 de marzo de 2024

ÍNDICE

Introducción	03
Descripción	05
Justificación	07
Desarrollo	11
Conclusión	17

INTRODUCCIÓN

¿Qué es el desarrollo de software?

El desarrollo de software se refiere a un conjunto de actividades informáticas dedicadas al proceso de creación, diseño, despliegue y compatibilidad de software.

El software en sí es el conjunto de instrucciones o programas que le dicen a una computadora qué hacer. Es independiente del hardware y hace que las computadoras sean programables. Hay tres tipos básicos:

1. Software del sistema para proporcionar funciones básicas como sistemas operativos, administración de discos, servicios, administración de hardware y otras necesidades operacionales.
2. Software de programación para brindar a los programadores herramientas como editores de texto, compiladores, enlazadores, depuradores y otras herramientas para crear código.
3. Software de aplicación (aplicaciones o apps) para ayudar a los usuarios a realizar tareas. Las suites de productividad de Office, el software de gestión de datos, los reproductores multimedia y los programas de seguridad son algunos ejemplos. Aplicaciones también se refiere a aplicaciones web y móviles como las que se utilizan para comprar en Amazon.com, socializar en Facebook o publicar imágenes en Instagram.
4. Un posible cuarto tipo es el software integrado. El software de sistemas integrado se utiliza para controlar máquinas y dispositivos que normalmente no se consideran computadoras, como redes de telecomunicaciones, automóviles, robots industriales y más. Estos dispositivos, y su software, se pueden conectar como parte del Internet de las Cosas (IoT).²

Los programadores, o codificadores, escriben el código fuente para programar computadoras para realizar tareas específicas como fusionar bases de datos, procesar pedidos en línea, enrutar comunicaciones, realizar búsquedas o mostrar texto y gráficos. Los programadores suelen interpretar

las instrucciones de los desarrolladores e ingenieros de software y utilizan lenguajes de programación como C.

El lenguaje C es un lenguaje estructurado, en el mismo sentido que lo son otros lenguajes de programación tales como el lenguaje Pascal, el Ada o el Modula-2, pero no es estructurado por bloques, o sea, no es posible declarar subrutinas (pequeños trozos de programa) dentro de otras subrutinas, a diferencia de como sucede con otros lenguajes estructurados tales como el Pascal. Además, el lenguaje C no es rígido en la comprobación de tipos de datos, permitiendo fácilmente la conversión entre diferentes tipos de datos y la asignación entre tipos de datos diferentes, por ejemplo, la expresión siguiente es válida en C:

```
float a; /Declaro una variable para numeros reales/  
  
int b; /Declaro otra variable para numero enteros/  
  
b=a; /Asigno a la variable para entera el numero real/
```

Todo programa de C consta, básicamente, de un conjunto de funciones, y una función llamada main, la cual es la primera que se ejecuta al comenzar el programa, llamándose desde ella al resto de funciones que compongan nuestro programa.

A lo largo del desarrollo de esta actividad, estaremos haciendo uso de un lenguaje de Programación en C para crear el pseudocódigo de los problemas de las actividades anteriores, en las cuales ya se realizó el algoritmo y el diagrama de flujo.

DESCRIPCIÓN

Contextualización:

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:

- 1) 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.
- 2) 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.
- 3) 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. Para comenzar con el desarrollo de estos programas, se solicita que se sigan los siguientes pasos para la elaboración de dichos programas:

Para cada problema matemático, se deberá:

1. Realizar el algoritmo de cada programa
2. Realizar su diagrama de flujo por cada uno

3. Codificarlo finalmente en lenguaje C

Ya se ha realizado el algoritmo de cada programa, ahora se deberá plasmarlo en un diagrama de flujo para comprender de mejor manera.

Actividad: Desarrollar los 3 programas matemáticos especificados que se han estado analizando en las dos actividades pasadas. Para ello, utilizar un compilador online (ver sección de Recursos). Sin embargo, es posible utilizar el compilador que se prefiera:

1. El primer programa (Números Primos) deberá imprimir (mostrar en pantalla) lo siguiente:
“Ingresa un número”, que representará el número ingresado. Si se trata de un número primo, se deberá mostrar: “Este número es primo”. En caso contrario, se mostrará: “Este número no es primo”. Por ejemplo:

```
Output
/tmp/ww8z9d9L42.o
Ingresa un número:
17
Este número es primo
```

```
Output
/tmp/ww8z9d9L42.o
Ingresa un número:
10
Este número no es primo
```

2. El segundo programa (Números Pares e Impares) deberá pedir que se ingrese un número. Si el número ingresado resulta ser un par, se deberá mostrar lo siguiente: “Este número es par”. En caso contrario: “Este número es impar”. Por ejemplo:

```
Output
/tmp/ww8z9d9L42.o
Ingresa un número:
12
Este número es par
```

```
Output
/tmp/ww8z9d9L42.o
Ingresa un número:
7
Este número es impar
```

3. El tercer programa (Números Invertidos) deberá ingresar un número de 4 dígitos. Después, se debe mostrar lo siguiente: ◇.

```
Output
/tmp/ww8z9d9L42.o
Ingresa un número de 4 dígitos:
9129
El número al revés es: 9219
```

JUSTIFICACIÓN

El lenguaje C es un lenguaje de nivel medio bajo, así que no requiere de muchos recursos, pero ofrece una buena respuesta y una ejecución rápida de programas, por lo que es ideal para quienes empiezan a programar. Además, mientras se va desarrollando el pseudocódigo se puede ir visualizando el funcionamiento de los programas mientras aún los estamos escribiendo en C. Por lo tanto, se considera que estos programas resultan más eficientes que los escritos en otros lenguajes.

En esta ocasión en particular, para la elaboración de los pseudocódigos que se están solicitando se estará utilizando PSEInt que es una herramienta mediante la cual con un simple e intuitivo pseudolenguaje en español (complementado con un editor de diagramas de flujo), proporciona un entorno de trabajo ideal para programadores principiantes.

La unidad más pequeña de programación en C se conoce como token C. Los tokens son palabras clave, identificadores, constantes, variables o cualquier otro símbolo que tenga algún significado en el lenguaje C.

Un programa escrito en C es un juego de tokens. La sintaxis de C determina cómo combinar estos tokens para escribir el código.

Cada bloque de código se encierra en corchetes. La base de cada bloque es la función principal **main()**.

Un programa puede contener varias funciones, cada una con su juego de parámetros.

Al final de cada operador se pone un punto y coma y si se olvida ponerlo, generalmente el compilador mostrará un mensaje de error para que sea corregido.

Tomando los diagramas de flujo de la actividad 2, estaremos realizando el pseudocódigo de cada uno:

Diagrama de Flujo: Número Primos

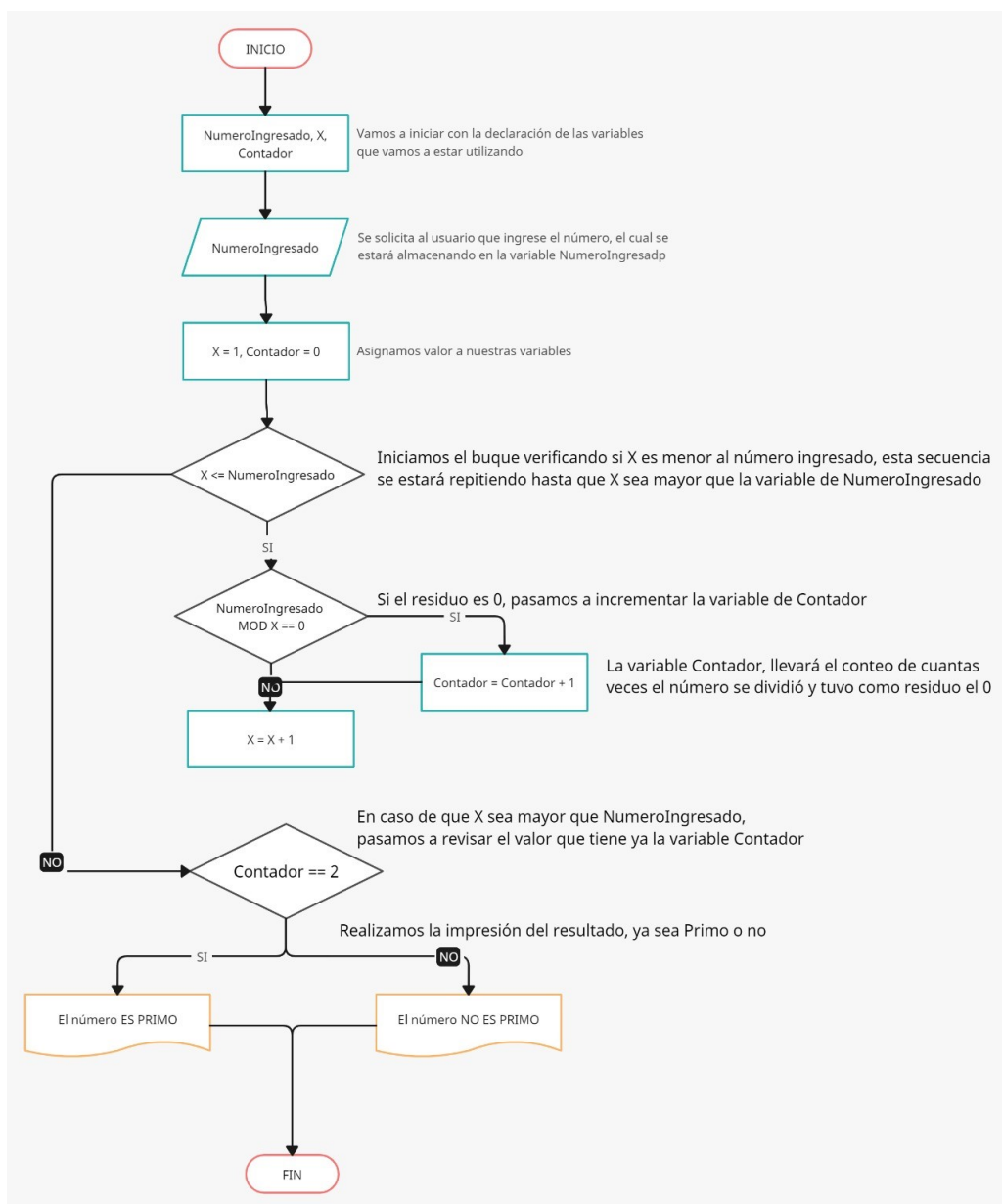


Diagrama de Flujo: Número Par e Impar

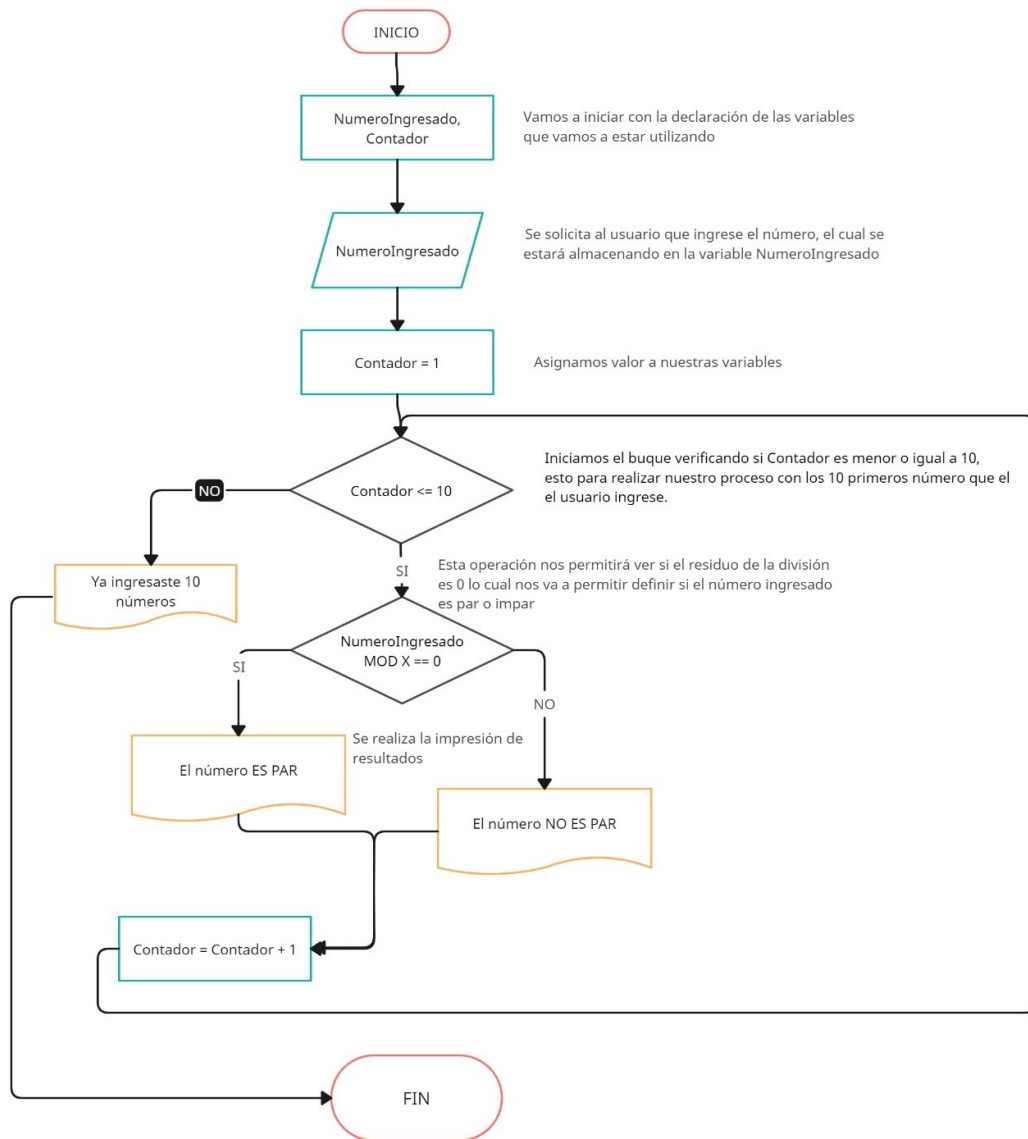
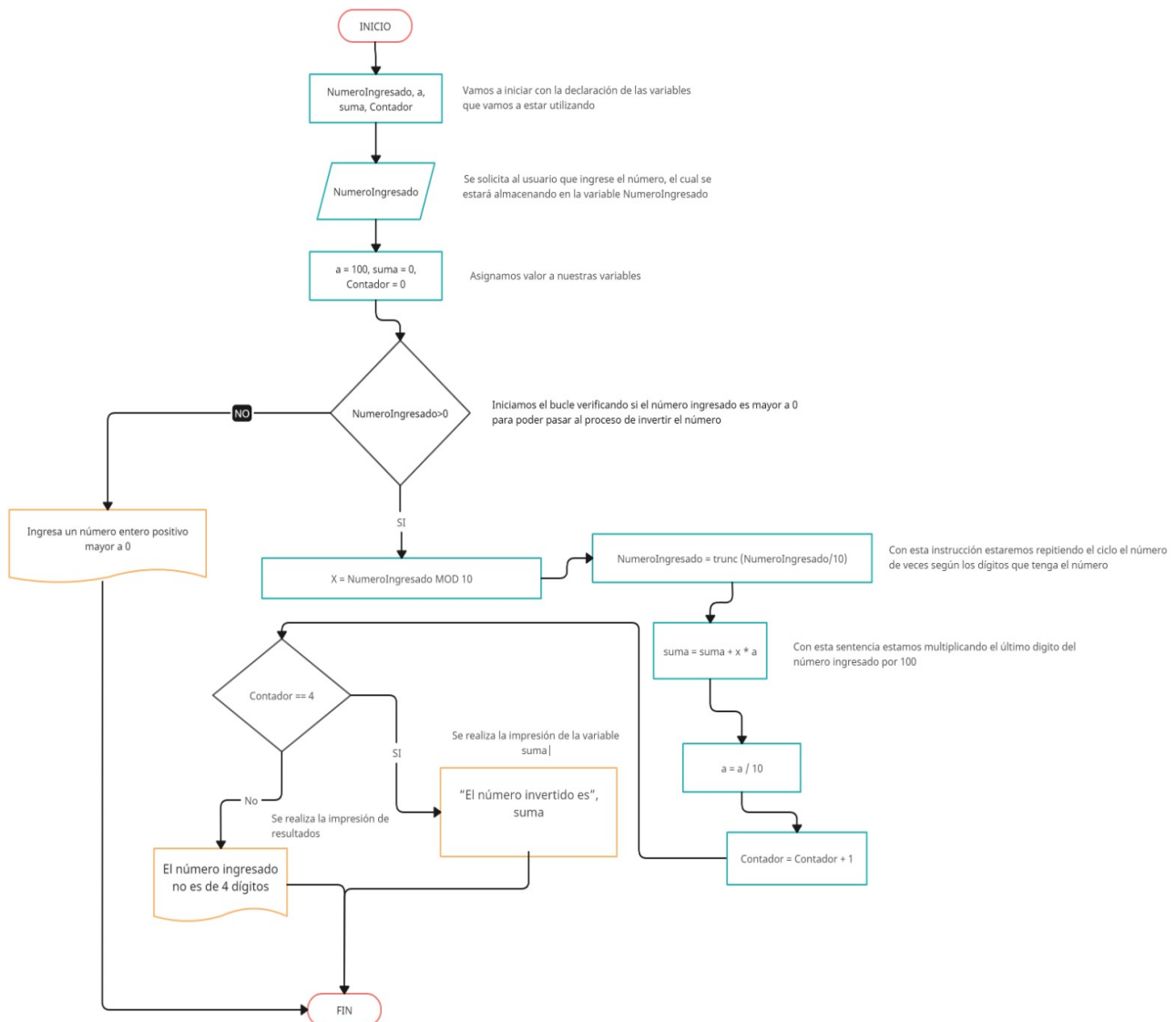


Diagrama de Flujo: Imprimir un número de 4 dígitos al revés



DESARROLLO

Números Primos: El 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”.

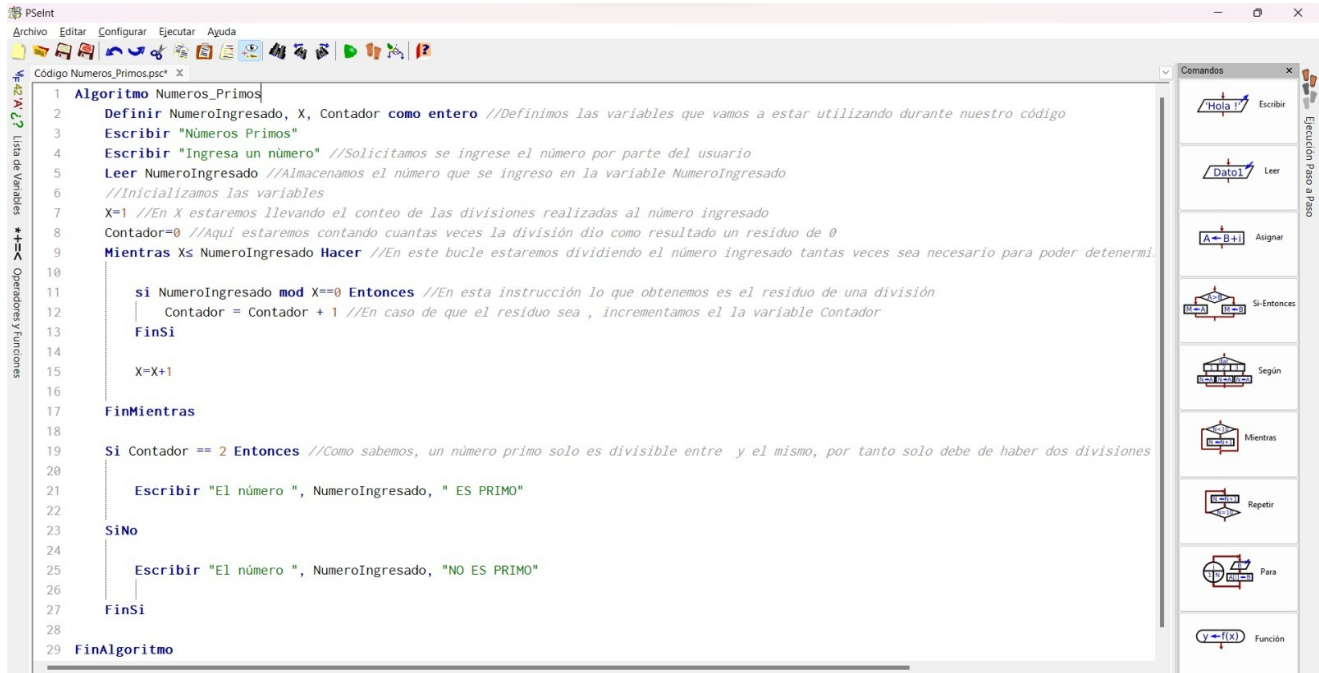
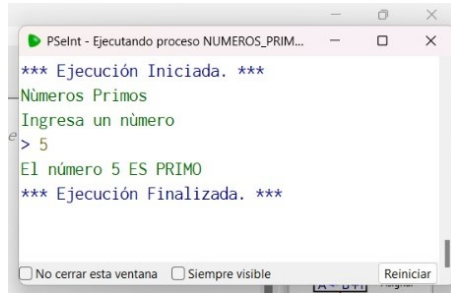


Imagen 1

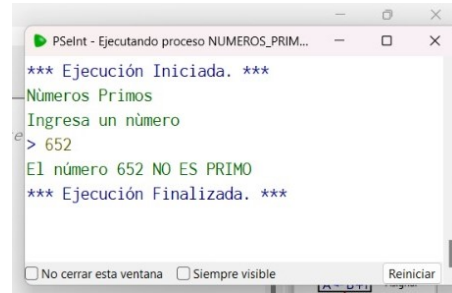
En la *Imagen 1* se puede observar el pseudocódigo que se escribió en el Lenguaje de Programación PSEint, en el cual se puede ver que se utiliza una sintaxis simple y en español, así como también permite el ir comentando cada línea de código para así saber cuál es la función.

En esta programación, lo que se hizo fue utilizar la función MOD, de la cual el resultado que obtenemos es el residuo de una división, el cual en el caso de números primos debe de ser 0 al dividir el número que ingrese el usuario entre 1 y el mismo; de lo contrario no cumple con la condición para ser número primo.

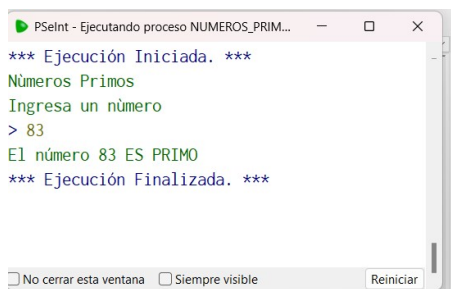
Por tanto, lo que hace el programa es que, una vez realizadas las divisiones necesarias, lleva un conteo de cuántas veces el residuo fue 0 para que si fue en 2 ocasiones nos indica que el número es Primo y en caso de que no, nos estará indicando que no es un número Primo; en las siguientes imágenes se observa la ejecución del mismo:



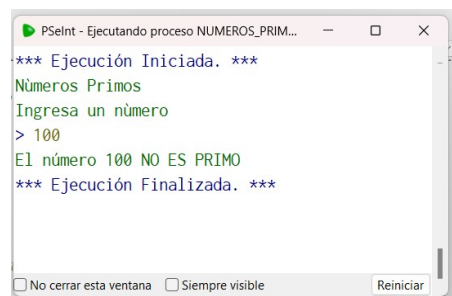
```
*** Ejecución Iniciada. ***
Números Primos
Ingresa un número
> 5
El número 5 ES PRIMO
*** Ejecución Finalizada. ***
```



```
*** Ejecución Iniciada. ***
Números Primos
Ingresa un número
> 652
El número 652 NO ES PRIMO
*** Ejecución Finalizada. ***
```



```
*** Ejecución Iniciada. ***
Números Primos
Ingresa un número
> 83
El número 83 ES PRIMO
*** Ejecución Finalizada. ***
```



```
*** Ejecución Iniciada. ***
Números Primos
Ingresa un número
> 100
El número 100 NO ES PRIMO
*** Ejecución Finalizada. ***
```

Número Par e Impar: El 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, en las siguientes imágenes se observa la ejecución del mismo

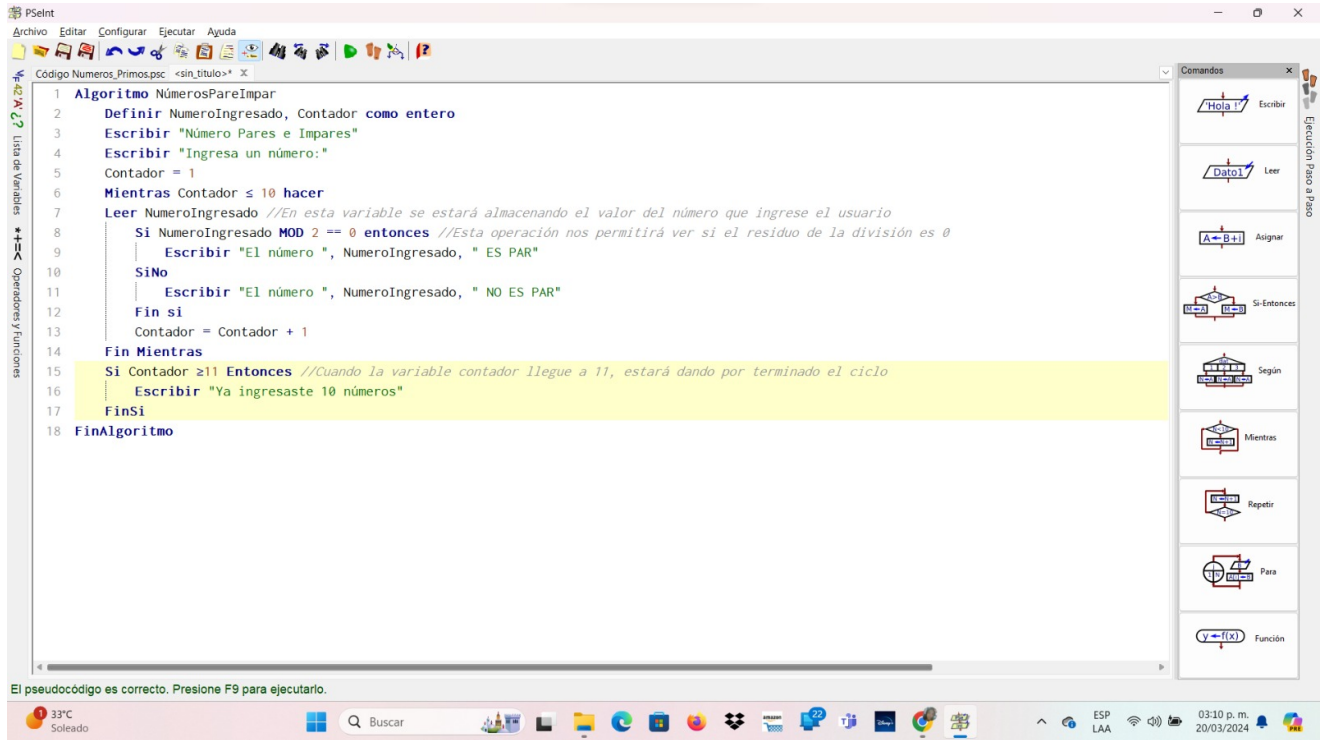



Imagen 2

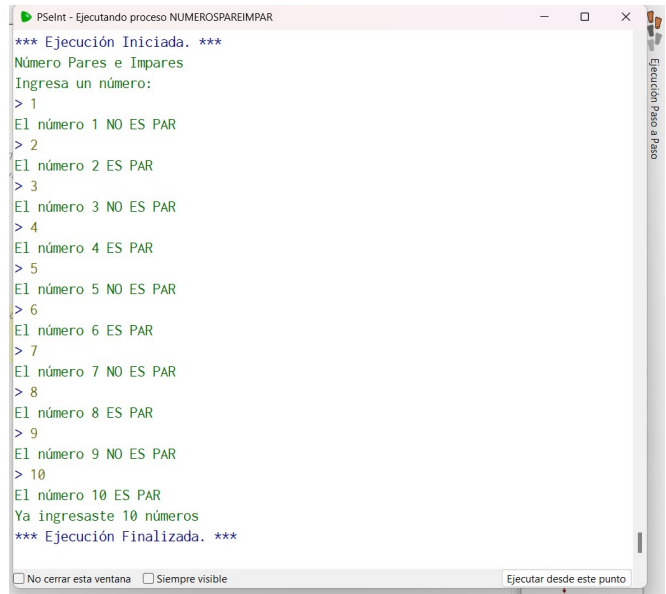
En la **Imagen 2** se muestra el código en el cual se pide que el usuario ingrese un número y el programa va a determinar si es Par o Impar, y esta misma acción la debe de realizar con 10 números; lo que se hizo fue utilizar un ciclo MIENTRAS que nos va a permitir hacer el análisis de los números que el usuario ingresa durante 10 veces, en cuanto el usuario ingrese el número por 11va vez saldrá el mensaje “Ya ingresaste 10 números”

Para poder determinar si un número es par o impar, utilizamos la función MOD, la cual nos va a devolver el residuo de una división; y como sabemos un número par al ser dividido entre 2, da como residuo 0.

A continuación, se muestra la ejecución del programa:



```
*** Ejecución Iniciada. ***
Número Pares e Impares
Ingresa un número:
> 15
El número 15 NO ES PAR
> 85
El número 85 NO ES PAR
> 36
El número 36 ES PAR
> 5
El número 5 NO ES PAR
> 7
El número 7 NO ES PAR
> 10
El número 10 ES PAR
> 65
El número 65 NO ES PAR
> 20
El número 20 ES PAR
> 89
El número 89 NO ES PAR
> 35
El número 35 NO ES PAR
Ya ingresaste 10 números
*** Ejecución Finalizada. ***
```



```
*** Ejecución Iniciada. ***
Número Pares e Impares
Ingresa un número:
> 1
El número 1 NO ES PAR
> 2
El número 2 ES PAR
> 3
El número 3 NO ES PAR
> 4
El número 4 ES PAR
> 5
El número 5 NO ES PAR
> 6
El número 6 ES PAR
> 7
El número 7 NO ES PAR
> 8
El número 8 ES PAR
> 9
El número 9 NO ES PAR
> 10
El número 10 ES PAR
Ya ingresaste 10 números
*** Ejecución Finalizada. ***
```

Número al revés: el 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  Algoritmo Número_Invertido
2  Definir NumeroIngresado,X,Contador como entero
3  Definir a,suma como real
4  Escribir "NÚMERO AL REVES"
5  Escribir "Ingresa un número de 4 dígitos"
6  Leer NumeroIngresado //Esta será nuestra variable NumeroIngresado, en la que se estará almacenando el valor del número que ingrese el usuario
7  a = 1000
8  suma = 0
9  Contador = 0 //Utilizaremos esta variable para revisar si el número que se ingresó consta de 4 dígitos
10 Mientras NumeroIngresado > 0 Hacer
11     X = NumeroIngresado MOD 10
12     //Con esta instrucción estaremos repitiendo el ciclo el número de veces según los dígitos que tenga el número
13     NumeroIngresado = trunc (NumeroIngresado/10)
14     suma = suma + X * a // Con esta instrucción estamos multiplicando el último dígito de nuestro número por 1000, para así pasarlo de unidades a miles;
15     //y así sucesivamente con cada dígito del número, es decir, lo vamos pasando de unidades a miles, de decenas a centenas, de centenas a decenas
16     //y miles a unidades
17     a = a / 10 //inicializamos esta variable en 1000, ya que el número a invertir es de 4 dígitos y conforme va avanzando el ciclo lo vamos decrementando
18     // para ir sacando los demás valores del número ingresado
19     Contador = Contador + 1//en este Contador llevamos la cuenta de las veces que se repite el ciclo, para así determinar si el número ingresado
20     // es un número de 4 dígitos
21 FinMientras
22 Si Contador == 4 entonces
23     Escribir "El número de 4 cifras inverso es: ",suma
24 SiNo
25     Escribir "El número NO ES DE 4 CIFRAS"
26 FinSi
27 FinAlgoritmo
28

```

La ejecución ha finalizado sin errores.

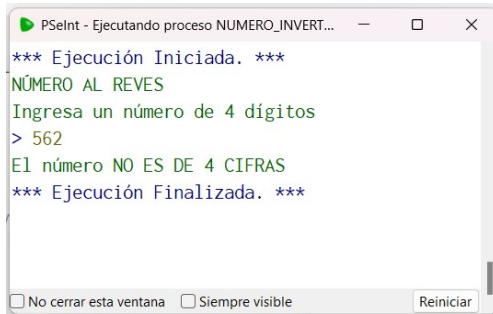
Para este caso, una vez que el usuario ingrese el número de 4 dígitos, lo tendremos que “descomponer” en unidades, decenas, centenas y miles; lo cual estaremos realizando a través divisiones, sumas y multiplicaciones dentro de la programación; además de ello se está haciendo uso de las funciones **MOD** y **trunc**.

Para la solución de este problema si inicio con la división del número que el usuario ingresa a través de la función **MOD 10** (divide el número entre 10) para irlo separando y que conforme lo valla dividiendo estará almacenando ese valor obtenido en la variable **Suma**, ahora para garantizar que este proceso se estará realizando las veces que sea necesario, se utiliza la función **trunc**.

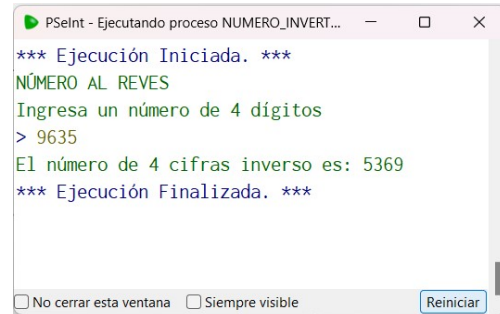
Con la función **trunc** lo que se hace es ir decrementando el número que ingreso el usuario para que el ciclo se repita tantas veces según los dígitos que tenga el número que ingresa el usuario, en este caso tendrá que ser 4 veces

Dentro de la variable **Suma**, vamos lo que va obteniendo en la función MOD para que se valla formando el número al revés; así mismo nos vamos apoyando en demás variables como la **a** que inicializamos con el valor 1000 debido a que tendremos un número de 4 dígitos; la cual dentro del mismo ciclo vamos decrementando para ir haciendo la operación correspondiente

A continuación, se muestra la ejecución del programa:



```
*** Ejecución Iniciada. ***
NÚMERO AL REVES
Ingresa un número de 4 dígitos
> 562
El número NO ES DE 4 CIFRAS
*** Ejecución Finalizada. ***
```



```
*** Ejecución Iniciada. ***
NÚMERO AL REVES
Ingresa un número de 4 dígitos
> 9635
El número de 4 cifras inverso es: 5369
*** Ejecución Finalizada. ***
```


CONCLUSIÓN

¿Por qué la programación en C es importante?

El lenguaje de programación C es muy eficiente ya que su código ofrece una estructura clara y, por tanto, facilita la creación de aplicaciones de una forma rápida y potente.

Asimismo, al contar con un número limitado de palabras clave, el aprendizaje del mismo es más rápido.

La programación en C ha sido la madre de otros lenguajes de programación, y por esa razón, puede llegar a ser parecida su sintaxis, como por ejemplo los lenguajes C++, Objective-C, C#, PHP o JavaScript; es por ello que tiene tanta importancia aprender a programar en C, ya que nos ayudará a los que iniciamos en la programación comprender con más facilidad otros lenguajes de programación.

El lenguaje C es eficiente, seguro y optimizado, utilizando un lenguaje de máquina que permite instrucciones muy optimizadas.

A lo largo de esta actividad se dio solución mediante la programación de problemas relativamente sencillos; sin embargo la programación tiene infinidad de aplicaciones, desde cuestiones básicas como las resueltas en esta actividad a más complejas como la Inteligencia Artificial y la programación en un lenguaje C es el inicio para todo programador en desarrollo.