

## Actividad | #2| Diagramas de Flujo

Ingeniería en Desarrollo de Software



TUTOR: Sandra Lara Devora

ALUMNO: Edgar Enrique Cuamea Ochoa

FECHA: 27 de julio del 2022

## Indice

Introducción. ....	3
Descripción. ....	4
Justificación. ....	5
Desarrollo.....	6
Diagrama algoritmo primos. ....	6
Diagrama algoritmo números pares. ....	8
Diagrama algoritmo número invertido.....	10
Conclusión. ....	11
Referencias.....	12

### **Introducción.**

En esta segunda actividad retomaremos lo visto en la actividad anterior y veremos como convertir el pseudocódigo que hicimos anteriormente a diferentes diagramas de flujo, un diagrama para la calculadora de números primos, otro para la calculadora de números pares y el ultimo para la calculadora de los números inversos donde se vera la función que realiza el pseudocódigo visto en la actividad anterior pero de una manera mas clara en su funcionamiento ya que utilizaremos el lenguaje natural y no pseudocódigo y símbolos para una mejor comprensión, así como diseñar estos diagramas de flujo para explicar el algoritmo que previamente escribimos en el programa PSeInt, por lo que utilizaremos la pagina web de Figma para diseñar los diagramas de flujo de las calculadoras anteriormente mencionadas, ya que esta pagina es sencilla de utilizar y además es de uso libre por lo que la utilizaremos para hacer visualmente mas atractiva el algoritmo de las calculadoras así como facilitar la comprensión de estas.

### **Descripción.**

Ya que realizamos nuestro pseudocódigo en el programa psetnt lo utilizaremos para realizar nuestro diagrama de flujo en figma, por lo que nos será mas fácil el realizar nuestro diagrama de flujo para comprender mas sobre el pseudocódigo que realizamos en la anterior actividad, así que solo convertiremos el pseudocódigo a nuestro diagrama, utilizaremos figuras y flechas para marcar inicios y fin de cada algoritmo y como cada flecha nos mandara a otra nueva instrucción si la pregunta o variable dentro de un recuadro sea verdadera o falsa con el fin de hacer mas atractiva visualmente el algoritmo así como comprender mas sobre las instrucciones que hicimos en el pseudocódigo, ya que al no el lenguaje de programación facilita la comprensión de este pero si no estas familiarizado con la programación, un diagrama de flujo sera mas sencillo de entender que un pseudocódigo y al ser mas atractiva visualmente sera mas comprensible para quien lo lee.

### **Justificación.**

Dicho anteriormente utilizaremos la pagina web de figma ya que esta es sencilla de utilizar agregando que es de uso libre para hacer los diagramas de flujo de nuestras calculadoras, así también utilizaremos como base los pseudocódigos escritos en el programa pslnt para convertirlos en diagrama de flujo ya que este también cuenta con opciones para convertir el pseudocódigo en diagrama de flujo opte por utilizar figma pues utilizare un lenguaje que sea mas comprensible para todo aquel ajeno al desarrollo de software para facilitar su comprensión del código simplemente siguiendo los caminos que trazaremos en los diagramas de flujo, también utilizaremos el uso de ciclos y como diferentes ciclos pueden alterar el uso de las variables e incluso el resultado esperado haciendo que el código sea mas comprensible con el uso de los diagramas de flujo para cada una de las calculadoras anteriormente mencionadas ya que cada una tiene un funcionamiento distinto pero con un inicio y fin iguales entre si para facilitar el uso de este sin errores o fallos en el algoritmo.

### Desarrollo.

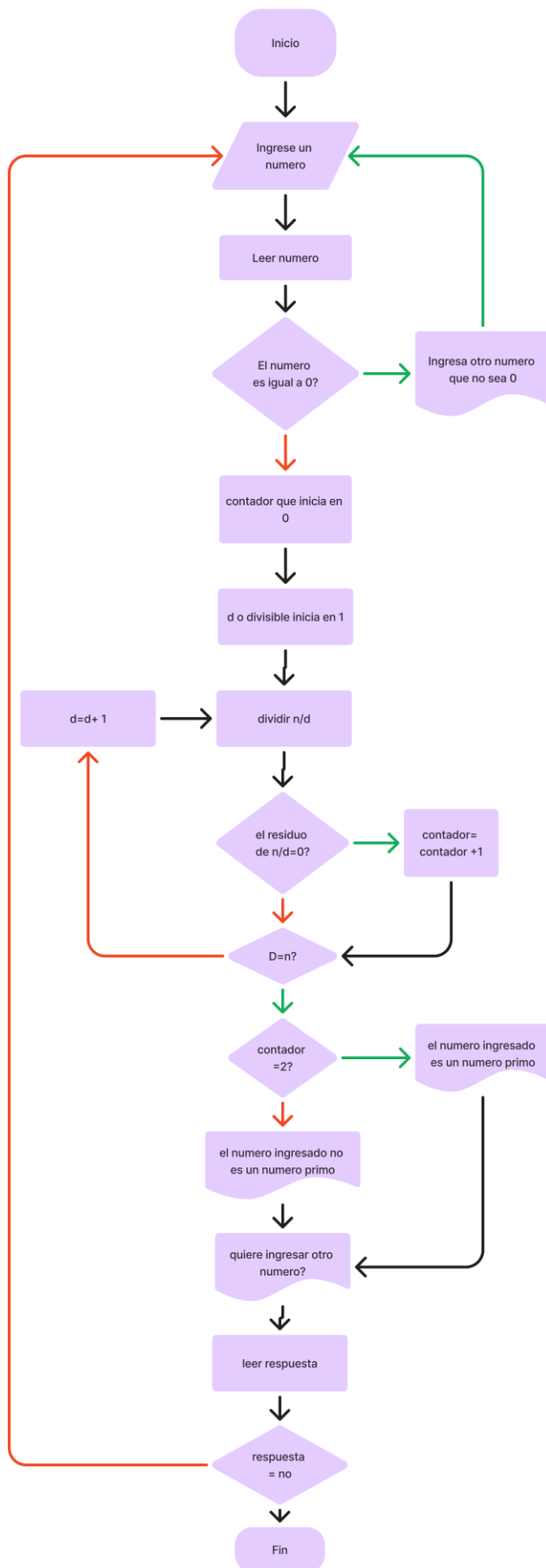
Para el primer diagrama de flujo estaremos utilizando el pseudocódigo de el primer algoritmo que realizamos, en este caso el diferenciar un numero si es número primo o no, para el segundo diagrama utilizaremos el algoritmo que diferencia un numero par de uno impar y para el tercer y último algoritmo haremos que los números ingresados los devuelva de manera invertida por lo que necesitamos definir las funciones de nuestro diagrama que serán las siguientes:



Tenemos un inicio y un fin para el algoritmo, un diamante para la decisión, un trapezico para la entrada de datos y el rectangulo que lo utilizare como proceso, ya sea utilizarlo como contador o para la funcion “para” que utilizamos en el algoritmo lo utilizaremos como bucle.

### Diagrama algoritmo primos.

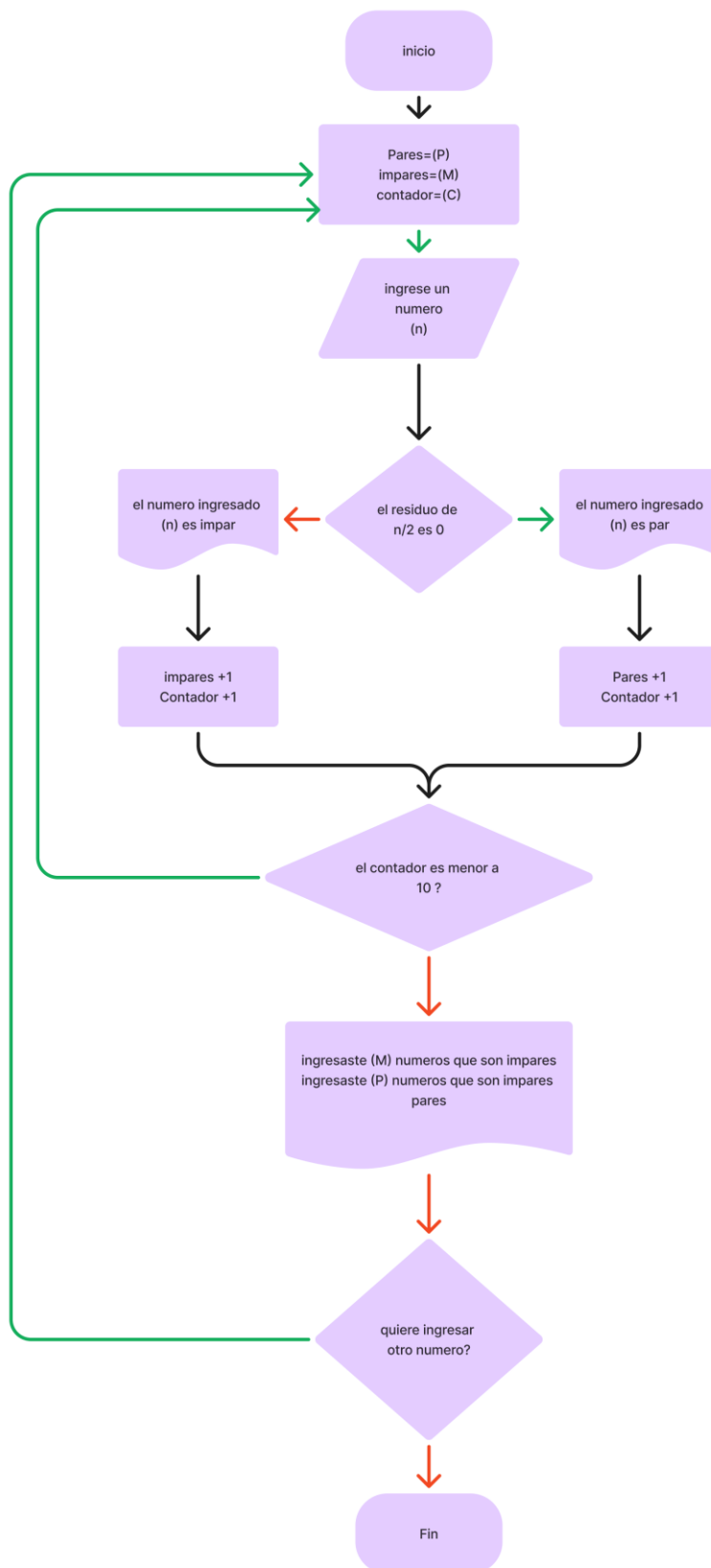
Para el primer algoritmo necesitamos diferenciar entre un numero primo o no por lo que se realizan una serie de operaciones donde solo el numero que ingresamos solo se divide en 1 y por si mismo sin dejar residuos entonces es un numero primo por lo que nos quedara de la siguiente forma:



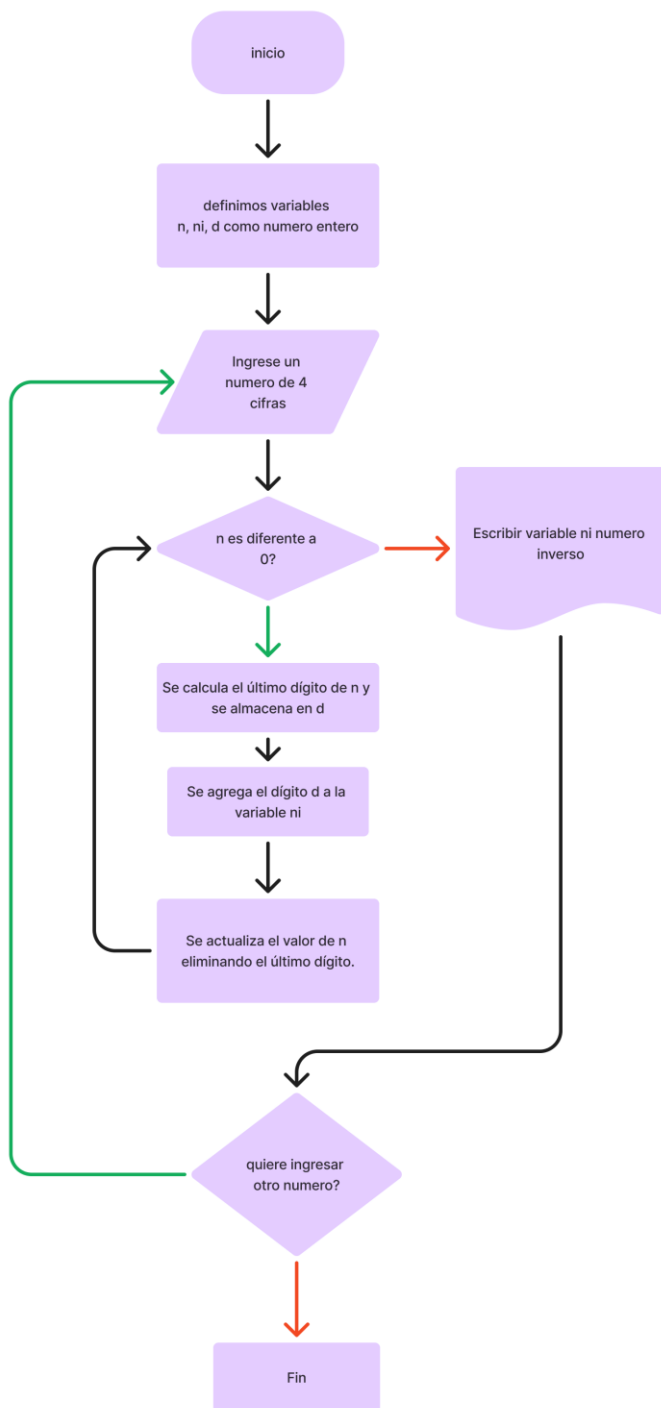
**Diagrama algoritmo números pares.**

El objetivo de este algoritmo es identificar de 10 números que ingresemos los pares y los impares por lo que si ingresamos el numero 4 dirá que el numero es par y si ingresamos el 5 diga que es impar por lo que al final nos diga de los 10 números que ingresamos cuantos de ellos son par y cuantos son impar por lo que el diagrama de flujo es el siguiente:





# Diagrama algoritmo número invertido.



El objetivo de esta calculadora es ingresar 4 números y que nos devuelva los números de forma invertida así que si ingresamos 1234, esta devolverá 4321 por lo que se definen 3 variables, n para el numero ingresado, ni para el numero invertido y d para almacenar el ultimo digito de la variable n por lo que se actualiza la variable n para eliminar el ultimo digito y el numero de la variable d pasa a guardarse a la variable ni y volver al ciclo, cuando se hayan procesado los 4 números que ingresemos, obtendremos la variable ni con los números invertidos.

### **Conclusión.**

Como vimos en los diagramas de flujo de cada una de las calculadoras es más fácil diseñar el algoritmo de forma visual ya que se da a entender mas sobre lo que hace en las variables ya que en la última calculadora, el residuo de  $n/10$  es lo que le da valor a la variable d pero se elimina el numero entero solo guardando el decimal o el residuo por lo que tiene diferentes formas de expresar el mismo algoritmo por lo que es más fácil entenderlo con el diagrama de flujo ayudándonos a saber que compuertas lógicas podremos utilizar para un funcionamiento optimo de las calculadoras así como hacerlo de manera mas eficiente eliminando las variables innecesarias para un buque que no necesitamos, así que haciéndolo de manera automática en programas o haciendo las operaciones manuales se dará el mismo resultado ya que nuestro algoritmo funciona de manera óptima y sin errores.

### Referencias.

*Untitled.* (s. f.). Figma. <https://www.figma.com/board/Bm4fmIFUkjxDVIKklZqYSQ/Untitled?node-id=0-1&t=9CV2c2hrdpuES7D4-0>

Hackemi. (s. f.-b). *GitHub - Hackemi/Calculadoras: Actividades entregadas en PDF de la introduccion al desarrollo de software.* GitHub. <https://github.com/Hackemi/Calculadoras>