

## **DISEÑO DE CONJUNTO DE INSTRUCCIONES Y MICROARQUITECTURA ARQUITECTURA DEL COMPUTADOR**

### **Presentado por:**

Carlos Andrei Villamil Rueda cod: 100284616

Deyssi Catalina Arango Bermúdez cod: 100271305

Nelson Alonso Camacho Camacho cod: 100310957

Kevin Geovanny Camargo Bautista cod: 2011025769

Luis Felipe Álvarez Ospina cod: 100260436

**Tutor: Blanco Ignacio**

*Septiembre de 2022*

---

## **CONTENIDO**

### **1. Objetivos**

1.1 Objetivo General

1.2 Objetivos Específicos

### **2. Desarrollo**

2.1 Definición

2.2 Código

## **1. OBJETIVOS**

## 1.1 OBJETIVO GENERAL

Se debe diseñar el conjunto de instrucciones y la microarquitectura de un procesador que permita determinar si un número es primo o no.

## 1.2 OBJETIVOS ESPECÍFICOS

- Crear un pseudocódigo o código en lenguaje de alto nivel que determine si un número es primo o no.
- Definir qué instrucciones de bajo nivel son requeridas y diseñar el Conjunto de Instrucciones (ISA).
- Definir la estructura de hardware mínima para poder ejecutar todas las instrucciones definidas en el conjunto de instrucciones.
- Simular la microarquitectura del hardware.

## 2. DESARROLLO

### 2.1 DEFINICIÓN

Como sabemos, los números primos son los números naturales mayores que uno que tiene únicamente dos divisores positivos distintos: él mismo y el uno. Para la implementación del código propuesto se utilizó el lenguaje de programación de alto nivel **Python** para validar si un número es primo o no.

Para validar si el número ingresado es primo, procedemos a evaluar las siguientes condiciones:

- Validar que el número sea mayor a cero.
- Validar que el número no sea uno: Debido a que el número uno no se considera primo.
- Validar que el número sea dos: Debido a que es el único número par que es primo.
- Validar que el número no sea par mayor a dos: Ya que cualquier número par será divisible por el número dos u otro número.
- Validar que el número no sea divisible por ningún otro número impar hasta la mitad de este: Debido a que solo los números impares mayores o iguales a 3 pueden ser primos, se procede a verificar si el número ingresado por el usuario es divisible por estos. Esta condición solo se verifica hasta la mitad del número ingresado por el usuario buscando optimizar el procesamiento de estos números.

### 2.2 CÓDIGO

```

In [5]: #creamos una función para imprimir en pantalla al usuario si el número ingresado no es
def mostrar_no_es_primo(num_in):
    print(f"El número", num_in, "no es primo")
    return

#creamos una función para imprimir en pantalla al usuario si el número ingresado es pr
def mostrar_es_primo(num_in):
    print(f"El número", num_in, "es primo")
    return

"""con esta función podemos validar mediante las condiciones dadas en la definición de
algoritmo si el número es primo o no"""
def validar_primo():
    #definimos una variable para guardar el número ingresado al sistema
    num_in = int(input("Ingrese un número positivo: "))
    #validamos que el número sea mayor a cero.
    if num_in == 0:
        mostrar_no_es_primo(num_in)
        return
    #validamos que el número no sea uno: Debido a que el número uno no se considera pr
    if num_in == 1:
        mostrar_no_es_primo(num_in)
        return
    #validamos que el número sea dos: Debido a que es el único número par que es primo
    if num_in == 2:
        mostrar_es_primo(num_in)
        return
    """Validar que el número no sea par mayor a dos:
    Ya que cualquier número par será divisible por el número dos u otro número.
    esto lo verificamos con el operador mod % """
    if num_in % 2 == 0:
        mostrar_no_es_primo(num_in)
        return
    #validamos que el número no sea divisible por ningún otro número impar hasta L
    mitad = int(num_in/2)
    for i in range(3, mitad):
        if num_in % i == 0:
            #si hay número divisor, mostramos cual es en pantalla
            print(f"El número divisor es:", i)
            mostrar_no_es_primo(num_in)
            return
    mostrar_es_primo(num_in)
    return

#hacemos el llamado de la función
if __name__ == '__main__':
    validar_primo()

```

Ingrese un número positivo: 787  
 El número 787 es primo

Ejemplos con evaluación de otros números para probar el algoritmo:

```
Ingrese un número positivo: 87987987987987
El número divisor es: 3
El número 87987987987987 no es primo

Process finished with exit code 0
```

```
Ingrese un número positivo: 15
El número divisor es: 3
El número 15 no es primo

Process finished with exit code 0
```

```
Ingrese un número positivo: 7
El número 7 es primo

Process finished with exit code 0
```