



Fundamentos de programación, NRC:  
200274, martes y jueves de 11:00 a 1:00 PM,  
Diego Raúl Damián Torres 220097027,  
Resumen de funciones.

### **Funciones en Programación: Pilares Fundamentales de la Modularidad y Reutilización de Código**

En el mundo de la programación, las funciones son bloques de construcción esenciales que permiten organizar y estructurar el código de manera modular y eficiente. Desde los lenguajes de programación más básicos hasta los más avanzados, las funciones desempeñan un papel crucial en la creación de software robusto y mantenible. En esta exploración, vamos a sumergirnos en qué son las funciones, cómo se definen, su sintaxis, su importancia y algunas mejores prácticas para su uso.

#### **¿Qué es una Función en Programación?**

En términos simples, una función en programación es un bloque de código que realiza una tarea específica cuando se llama o se invoca desde otra parte del programa. Al igual que las funciones en matemáticas, las funciones en programación pueden tomar uno o más argumentos como entrada, realizar ciertas operaciones y devolver un resultado. Este resultado puede ser un valor, un objeto o incluso nada en absoluto.

#### **Sintaxis y Definición de Funciones**

La sintaxis para definir una función varía según el lenguaje de programación, pero en general sigue un patrón común. Por ejemplo, en Python, una función se define

usando la palabra clave **def**, seguida del nombre de la función y paréntesis que pueden contener los parámetros de entrada, y luego un bloque de código endentado que define lo que hace la función.

Ejemplo:

```
def suma(a, b):  
    resultado = a + b  
    return resultado
```

En este caso, hemos definido una función llamada **suma** que toma dos argumentos **a** y **b**, los suma y devuelve el resultado.

## Importancia de las Funciones en Programación

Las funciones son fundamentales en programación por varias razones:

1. **Modularidad:** Permiten dividir el código en unidades más pequeñas y manejables, lo que facilita su comprensión y mantenimiento.
2. **Reutilización de Código:** Una vez definida, una función puede ser llamada desde cualquier parte del programa, lo que evita la repetición de código y promueve la reutilización.
3. **Abstracción:** Las funciones permiten ocultar los detalles de implementación y proporcionar una interfaz clara para interactuar con el código.
4. **Organización:** Ayudan a organizar el código en bloques lógicos y cohesivos, lo que mejora la legibilidad y la estructura general del programa.

## Mejores Prácticas para el Uso de Funciones

Algunas mejores prácticas para el uso de funciones incluyen:

5. **Nombres Significativos:** Usa nombres descriptivos para tus funciones que indiquen claramente su propósito.
6. **Abstracción Adecuada:** Divide las funciones en unidades coherentes y con una única responsabilidad.
7. **Documentación:** Proporciona comentarios y documentación clara sobre el propósito y los parámetros de entrada y salida de cada función.
8. **Pruebas Unitarias:** Prueba cada función por separado para asegurarte de que funcione como se espera.
9. **Evitar Efectos Secundarios:** Las funciones deben evitar modificar variables globales y operar principalmente sobre sus argumentos.

## Referencias:

▷ [Que es una Función en Programación \(programacion.top\)](#)

<https://lenguajesdeprogramacion.net/diccionario/que-es-una-funcion-o-metodo-en-programacion/>

[FUNCIONES en programación. Definición y ejemplos!! \(apinem.com\)](#)