



Ensayo

Nombre del Alumno: **Vania Natali Santizo Morales**

Nombre del tema: **Ensayo**

Parcial: **I Parcial**

Nombre de la Materia: **Programación lógica**

Nombre del profesor: **Juan José Ojeda Trujillo**

Nombre de la Licenciatura: **Ingeniería en Sistemas Computacionales**

Cuatrimestre: **8 Cuatrimestre**

Introducción

La programación es como aprender a cocinar: hay muchas recetas y técnicas, pero lo importante es saber qué ingredientes usar y cómo combinarlos. En este ensayo, vamos a explorar dos estilos de programación: la lógica y la funcional. Estos estilos son como dos formas diferentes de pensar en cómo resolver problemas con código. Además hablaremos de conceptos básicos como funciones, tipos de datos y listas, que son herramientas esenciales para cualquier programador. Si alguna vez te has preguntado cómo las computadoras entienden lo que les decimos este es un buen punto de partida para descubrirlo.

Unidad 1: Conceptos fundamentales de programación lógica y funcional

1.1. Estilos de programación

En el mundo de la programación, hay varios estilos o formas de escribir código. La programación lógica se basa en la lógica matemática y se enfoca en resolver problemas mediante reglas y relaciones. Por ejemplo, si decimos "si A es verdadero y B es verdadero, entonces C es verdadero", estamos usando lógica. Por otro lado, la programación funcional se basa en el uso de funciones matemáticas. Aquí, todo se resuelve mediante funciones que toman datos de entrada y producen resultados. Es como una receta: pones ingredientes (datos) y obtienes un pastel (resultado).

1.2. Evaluación de expresiones

Cuando programamos, las expresiones son como las frases que le decimos a la computadora. Por ejemplo, "2 + 3" es una expresión que la computadora evalúa para obtener el resultado "5". En la programación funcional, las expresiones son muy importantes porque todo se basa en ellas. La computadora las resuelve paso a paso, como si estuviera siguiendo una serie de instrucciones.

1.3. Definición de funciones

Las funciones son como pequeñas máquinas que hacen algo específico. Por ejemplo, una función puede sumar dos números o convertir una palabra en mayúsculas. En la programación funcional, las funciones son las estrellas del show. Se definen de manera clara y luego se usan una y otra vez para resolver problemas.

1.4. Disciplina de tipos

Los tipos son como etiquetas que le decimos a la computadora qué tipo de dato estamos usando. Por ejemplo, un número es de tipo "entero" y una palabra es de tipo "texto". La disciplina de tipos es importante porque nos ayuda a evitar errores. Si intentamos sumar un número con una palabra, la computadora nos dirá que algo está mal.

1.5. Tipos de datos

Los tipos de datos son las categorías en las que clasificamos la información. Los más comunes son los números (enteros, decimales), los textos (cadenas de caracteres) y los booleanos (verdadero o falso). Saber qué tipo de dato estamos manejando es clave para escribir programas que funcionen bien.

Unidad 2: Programación funcional

2.1. El tipo de datos

En la programación funcional, los tipos de datos son aún más importantes porque todo gira en torno a cómo se manipulan los datos. Por ejemplo, si tenemos una lista de números, debemos saber cómo trabajar con ella: sumar sus elementos, filtrar los que cumplen una condición, etc.

2.2. Funciones

Las funciones en la programación funcional son como bloques de construcción. Pueden ser simples, como una función que suma dos números, o complejas, como una función que procesa una lista completa. Lo interesante es que las funciones pueden pasarse como argumentos a otras funciones, lo que las hace muy versátiles.

2.3. Intervalos

Los intervalos son secuencias de números o valores. Por ejemplo, un intervalo puede ser los números del 1 al 10. En programación funcional, los intervalos son útiles para generar listas o para repetir operaciones sobre un rango de valores.

2.4. Operadores

Los operadores son símbolos que realizan acciones sobre los datos. Por ejemplo, el signo "+" suma dos números, y el signo ">" compara si un número es mayor que otro. En la programación funcional, los operadores se usan mucho para manipular datos dentro de las funciones.

2.5. Aplicaciones de las listas

Las listas son colecciones de datos. En la programación funcional, las listas son muy importantes porque permiten almacenar y manipular muchos datos a la vez. Por ejemplo, podemos usar una lista para guardar las edades de un grupo de personas y luego calcular el promedio de esas edades.

2.6. Árboles

Los árboles son estructuras de datos más complejas. Imagina un árbol genealógico: cada persona tiene padres, que a su vez tienen otros padres, y así sucesivamente. En programación, los árboles se usan para representar datos jerárquicos, como la estructura de un archivo en una computadora.

Conclusión

La programación lógica y funcional son dos formas diferentes de resolver problemas con código. La programación lógica se basa en reglas y relaciones, mientras que la programación funcional se enfoca en el uso de funciones y la manipulación de datos. Ambos estilos tienen sus propias herramientas, como las listas, los árboles y los operadores que nos ayudan a escribir programas eficientes y fáciles de entender. Aprender estos conceptos es como aprender un nuevo idioma: al principio puede parecer complicado, pero con práctica, todo empieza a tener sentido. Lo mejor es que nos permite crear cosas increíbles con solo unas cuantas líneas de código.