GUIA NIVELACION DISEÑO DE ALGORITMOS

Introducción

El diseño de algoritmos es una habilidad fundamental en la informática y en la ingeniería de software. Los algoritmos son la base de la programación y de la resolución de problemas en computadoras. Esta guía de nivelación está diseñada para estudiantes que están comenzando con el diseño de algoritmos, proporcionando una comprensión desde los conceptos más básicos hasta técnicas de diseño más avanzadas.

Objetivos del Curso

- **Comprender qué es un algoritmo y su importancia.**
- **Aprender a diseñar y representar algoritmos mediante pseudocódigo y diagramas de flujo.**
- **Familiarizarse con estructuras de control y estructuras de datos básicas.**
- **Desarrollar habilidades para resolver problemas mediante la construcción de algoritmos eficientes.**

1. ¿Qué es un Algoritmo?

Un **algoritmo** es una secuencia finita de pasos bien definidos y ordenados que se utilizan para resolver un problema o realizar una tarea específica. Los algoritmos pueden ser representados de varias formas, como pseudocódigo, diagramas de flujo, o directamente en un lenguaje de programación.

Características de un Algoritmo

Profesor: Harol Hernan Torres Neuta
E-Mail: Harol.Torres@pi.edu.co
Politécnico Internacional – Sede Sur – Bogota 2024

- **Finitud**: Un algoritmo debe tener un número finito de pasos.
- **Definición**: Cada paso del algoritmo debe estar claramente definido y sin ambigüedades.
- **Entrada**: Un algoritmo puede tener cero o más entradas.
- **Salida**: Un algoritmo debe producir al menos una salida.
- **Efectividad**: Cada paso del algoritmo debe ser realizable en un tiempo finito.

2. Representación de Algoritmos

Pseudocódigo

El pseudocódigo es una forma de escribir algoritmos de manera similar a un lenguaje de programación, pero sin la rigurosidad de la sintaxis de un lenguaje específico. Es útil para planificar y comunicar algoritmos de forma clara y entendible.

Ejemplo de Pseudocódigo: Algoritmo para encontrar el máximo de dos números.

```
Algoritmo EncontrarMaximo
    Entrada: Número A, Número B
    Si A > B Entonces
        Maximo ← A
    Sino
        Maximo ← B
    Fin Si
    Salida: Maximo
Fin Algoritmo
```

Diagramas de Flujo

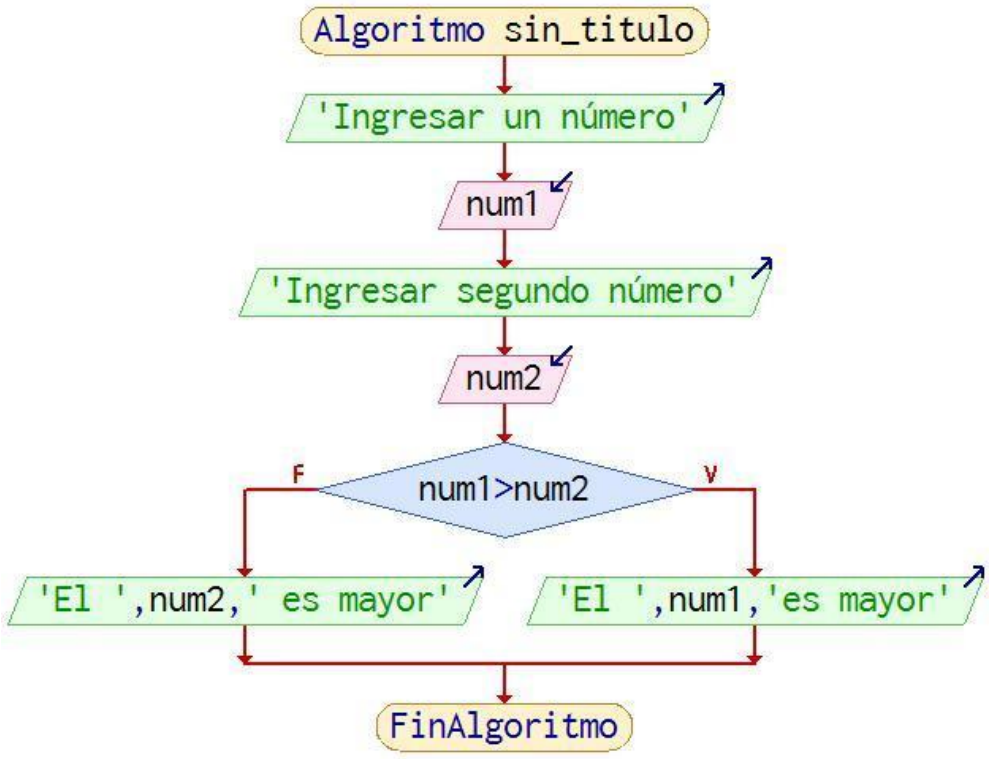
Un diagrama de flujo es una representación gráfica de un algoritmo. Utiliza símbolos estándar para representar diferentes tipos de acciones o pasos en un proceso.

Profesor: Harol Hernan Torres Neuta
E-Mail: Harol.Torres@pi.edu.co
Politécnico Internacional – Sede Sur – Bogota 2024

Símbolos Comunes en Diagramas de Flujo:

- **Óvalo**: Inicio y fin del algoritmo.
- **Rectángulo**: Proceso o instrucción.
- **Rombo**: Decisión o condición.
- **Flechas**: Indican el flujo del proceso.

Ejemplo de Diagrama de Flujo: Algoritmo para encontrar el máximo de dos números.



Profesor: Harol Hernan Torres Neuta
E-Mail: Harol.Torres@pi.edu.co
Politécnico Internacional – Sede Sur – Bogota 2024

3. Estructuras de Control

Las estructuras de control son la base del flujo de ejecución en un algoritmo. Existen tres tipos principales:

- **Secuencial**: Ejecución de instrucciones una tras otra.
- **Condicional (Decisión)**: Permite la ejecución de una instrucción o bloque de instrucciones basado en una condición (ej. if, else).
- **Iterativa (Repetición)**: Permite la repetición de un bloque de instrucciones hasta que se cumpla una condición (ej. while, for).

Ejemplos en Pseudocódigo

Estructura Secuencial:

```
Algoritmo Secuencial
    Leer A, B
    Suma ← A + B
    Mostrar Suma
Fin Algoritmo
```

Profesor: Harol Hernan Torres Neuta
E-Mail: Harol.Torres@pi.edu.co
Politécnico Internacional – Sede Sur – Bogota 2024

Estructura Condicional:

```
Algoritmo Condicional
    Leer A, B
    Si A > B Entonces
        Mostrar "A es mayor que B"
    Sino
        Mostrar "A no es mayor que B"
    Fin Si
Fin Algoritmo
```

Estructura Iterativa:

```
Algoritmo Iterativo
    Leer N
    Suma ← 0
    Para i ← 1 Hasta N Hacer
        Suma ← Suma + i
    Fin Para
    Mostrar Suma
Fin Algoritmo
```

Estructuras de Datos Básicas

Las estructuras de datos son formas de organizar y almacenar datos de manera eficiente para su uso y manipulación.

Profesor: Harol Hernan Torres Neuta
E-Mail: Harol.Torres@pi.edu.co
Politécnico Internacional – Sede Sur – Bogota 2024

- **Variables**: Almacenan valores individuales.
- **Arreglos (Vectores)**: Almacenan múltiples valores del mismo tipo.
- **Listas**: Colecciones de elementos que pueden variar en tamaño.

Ejemplos en Pseudocódigo

Uso de Variables:

```
Algoritmo UsoDeVariables
    Leer X
    Cuadrado ← X * X
    Mostrar Cuadrado
Fin Algoritmo
```

Uso de Arreglos:

```
Algoritmo UsoDeArreglos
    Declarar Arreglo[10]
    Para i ← 0 Hasta 9 Hacer
        Leer Arreglo[i]
    Fin Para
    Para i ← 0 Hasta 9 Hacer
        Mostrar Arreglo[i]
    Fin Para
Fin Algoritmo
```

Resolución de Problemas mediante Algoritmos

La capacidad de resolver problemas mediante el diseño de algoritmos implica seguir un enfoque estructurado:

1. **Comprender el Problema: Leer y entender el enunciado del problema.**
2. **Definir Entradas y Salidas: Identificar qué datos se necesitan y qué resultados se esperan.**
3. **Planificar el Algoritmo: Dividir el problema en pasos lógicos y escribir el pseudocódigo o dibujar el diagrama de flujo.**
4. **Implementar el Algoritmo: Traducir el pseudocódigo a un lenguaje de programación.**
5. **Probar y Depurar: Ejecutar el algoritmo con diferentes datos de prueba y corregir errores.**

El diseño de algoritmos es una habilidad crucial para cualquier programador o ingeniero de software. A través de la práctica constante y la resolución de diversos problemas, los estudiantes pueden desarrollar una comprensión profunda y habilidades avanzadas en el diseño y análisis de algoritmos. Esta guía de nivelación proporciona una base sólida para comenzar y progresar en esta área fundamental de la informática.

Profesor: Harol Hernan Torres Neuta
E-Mail: Harol.Torres@pi.edu.co
Politécnico Internacional – Sede Sur – Bogota 2024