

Definiciones Fundamentales

Analista Programador



Profesor

Matías Verges

Definiciones Fundamentales

1

Instrucciones

Son las órdenes que se utilizan para indicar a la computadora qué acción realizar.

3

Algoritmo

Es una serie de pasos ordenados y definidos que se siguen para resolver un problema.

2

Lenguaje de programación

Es el lenguaje que se usa para indicarle a la computadora qué es lo que tiene que hacer. Es un conjunto de reglas, símbolos y estructuras específicas para cada lenguaje.

4

Programa

Es un conjunto de instrucciones escritas en un lenguaje de programación que, al ser ejecutadas por una computadora, permiten realizar una tarea específica.

Definiciones Fundamentales

5

Compilar

Es el proceso de convertir el código fuente de un programa escrito en un lenguaje de programación, en un archivo ejecutable que puede ser utilizado por la computadora.

6

Depurar

Es el proceso de encontrar y corregir errores en el código de un programa.

7

IDE

Un IDE (Integrated Development Environment o Entorno de Desarrollo Integrado) es un software que proporciona un conjunto de herramientas y funcionalidades para desarrollar software de manera más eficiente. Es una herramienta para ayudar a los programadores a escribir, depurar y compilar código.

Diagramas de Flujo

Los diagramas de flujo en programación son una herramienta esencial, especialmente para quienes están iniciándose en este campo.

Son representaciones visuales que muestran el flujo lógico de un programa a través de pasos secuenciales. Estos pasos se simbolizan mediante bloques o formas geométricas conectadas por flechas que guían la dirección del flujo del programa.

Diagramas de Flujo



Inicio / Final

Representa el inicio y el final de un proceso



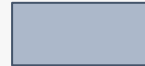
Línea de flujo

Indica el orden de la ejecución de las operaciones. La flecha indica la siguiente instrucción.



Entrada / Salida

Representa la lectura de datos en la entrada(input) y la impresión de datos en la salida(output)



Entrada / Salida

Representa la lectura de datos en la entrada(input) y la impresión de datos en la salida(output)



Decisión

Nos permite analizar una situación, con base en los valores verdadero y falso

Importancia

Claridad: Ayudan a entender claramente lo que hace el programa, desglosando los procesos complicados en pasos más simples y manejables.

Depuración: Facilitan la identificación de errores lógicos, permitiendo revisar y corregir el flujo antes de escribir el código.

Comunicación: Son útiles para explicar el diseño del programa a otros, ya que proporcionan una representación visual que puede ser más fácil de entender que el código escrito.

Cómo utilizarlos

Define el problema: Entiende claramente qué se supone que debe hacer tu programa.

Identifica los principales pasos: Desglosa el problema en tareas principales y represéntalas en tu diagrama.

Utiliza símbolos estándar: Emplea los símbolos convencionales para representar diferentes tipos de acciones y decisiones.

Cómo utilizarlos

Conecta los bloques: Usa flechas para mostrar el flujo de operaciones de principio a fin.

Valida el flujo: Revisa el diagrama para asegurarte de que representa correctamente la lógica que deseas implementar.

A tener en cuenta

Los diagramas de flujo son una excelente manera de visualizar la lógica de tus programas, especialmente cuando estás aprendiendo a programar. Te ayudan a estructurar tus pensamientos y a comunicar tus ideas de manera efectiva, lo cual es una habilidad crucial en programación.

Recordar

Estos símbolos están estandarizados y ampliamente reconocidos en la mayoría de los campos donde se utilizan los diagramas de flujo, lo que permite que personas de diferentes disciplinas o incluso diferentes partes del mundo comprendan la lógica presentada sin necesidad de explicaciones verbales detalladas.

Utilizar estos símbolos correctamente en un diagrama de flujo ayuda a clarificar el diseño del programa, facilita la depuración del código y mejora la comunicación entre los desarrolladores o entre el programador y los usuarios finales.

Consejos

Comienza con diagramas simples y aumenta gradualmente su complejidad.

Usa herramientas de software para dibujar diagramas de flujo, lo que facilita la edición y mejora la claridad.

<https://app.diagrams.net/>

Practica creando diagramas para problemas pequeños y, a medida que te sientas cómodo, avanza hacia problemas más complejos.

Ejercicios

Diseña los diagramas de flujo que describen la solución a cada uno de los siguientes problemas

Hacer un café con azúcar

Regar plantas cuando la tierra esté seca

Patear un penal

Cruzar la calle

Cruzar la calle en una esquina con semáforo

Cruzar la calle donde hay una sebra señalizada

Cocinar pasta

Vestirse apropiado para el clima

Cocinar el almuerzo

Deshacerse de ropa

Repartir los costos del asado con amigos

Elegir un tema para escuchar

Hacer ejercicio

Calcular la suma de dos números

Calcular la resta de dos números

Calculadora básica

Conversión de temperatura

Calcular la suma de 5 números

Algoritmo

Conjunto ordenado y finito de reglas y procedimientos que describen la solución de todos los problemas de un mismo tipo, independientemente de los datos

Formas de representar los algoritmos

Existen muchas formas de representar algoritmos, tres de las más conocidas son:

- Lenguaje natural
- Diagramas de flujo
- Pseudocódigo

Representar algoritmos

Lenguaje Natural

Una forma de expresar nuestros algoritmos es mediante el Lenguaje Natural. Ejemplos de este tipo de algoritmos son las recetas de cocina o los manuales de instrucciones. La gran ventaja de este tipo de algoritmos es la facilidad de comprensión

Desventajas

- El lenguaje natural no es universal. Estos algoritmos un chino no los comprendería.
- El lenguaje natural es ambiguo y eso da a lugar a errores
- Lo que para una persona puede ser una instrucción sencilla, para otra no

Pseudocódigo

El pseudocódigo es una forma de expresar los distintos pasos que va a tener un algoritmo de la forma similar a un lenguaje de programación.

- Permite especificar la estructura de un algoritmo independientemente del lenguaje.
- Indica la lógica que debe cumplir el programa.

Inicio

acción 1

acción 2

acción 3

acción 4

...

acción n

Fin

Sentencias de control

// Secuencia

Se ejecuta cada instrucción una después de la otra, desde arriba hacia abajo, secuencialmente

```
// Secuencia
```

```
acción 1
```

```
acción 2
```

```
acción 3
```

```
...
```

```
acción n
```

// Repetición o bucles

En un bucle hay una o varias instrucciones que se repiten dependiendo de una condición

// Repetición o bucles

```
Mientras <condición> hacer
    acción 1
    acción 2
    ...
    acción 3
FinMientras
```


// Repetición o bucles

Repetir

acción 1

acción 2

...

acción n

Hasta <condición>

// Repetición o bucles

Para cada <elemento> en <lista>

 Evaluar elemento

 Imprimir elemento

FinMientras

// Repetición o bucles

contador = 1

Repetir

 contador = contador + 1

 acción 1

Hasta <condición>

Imprimir contador

```
// Decisión
```

```
Si <condición>
```

```
    Acción 1
```

```
    Acción 2
```

```
    ...
```

```
    Acción n
```

```
FinSi
```

```
// Decisión
```

```
Si <condición>
```

```
    Acción 1
```

```
    Acción 2
```

```
Sino
```

```
    Acción 3
```

```
FinSi
```

// Ejemplo 1 - Cruzar la calle

Inicio

Me acerco al cordón

Si vienen autos

Espero

FinSi

Cruzo la calle

Fin

// Ejemplo 2 - Pagar facturas

Inicio

 Agarrar facturas

 Mientras tenga dinero

 Pagar factura

 Marcar factura como paga

 FinMientras

Fin

// Ejemplo 3 - Pagar Facturas

Inicio

Agarrar facturas

Leer totalEnLaCuenta

Mientras totalEnLaCuenta > 0

 Pagar factura

 totalEnLaCuenta = totalEnLaCuenta - totalFactura

 Marcar factura como paga

FinMientras

Fin

// Ejemplo 4 - Me alcanza?

Inicio

 Agarrar facturas

 Leer totalEnCuenta

 Para cada factura en facturas

 totalEnCuenta = totalEnCuenta - factura.total

 FinPara

 Si totalEnCuenta > 0

 Imprimir "Me alcanza"

 Sino

 Imprimir "No me alcanza"

 FinSi

Fin

Ejercicio: Pasar los diagramas de flujo anteriores a pseudocódigo

Validación de contraseña

Acceso denegado al 3er intento fallido!

Cálculo de la media de una serie de números

El último número es un cero

Determinación de un año bisiesto

Muestra la tabla de multiplicar del número introducido por el usuario.

Búsqueda lineal

En qué índice del arreglo está el número buscado?

Inversión de una cadena de texto

Cálculo de interés compuesto

Calcular el interés compuesto con base en el capital inicial, la tasa de interés y el número de periodos.

Verificación de un número primo

Fibonacci

Generar los primeros N números de la secuencia de Fibonacci.

Ordenamiento de burbuja

Para resolver en clase

Contador de vocales

Detección de palíndromos

Suma de dígitos de un número

Cálculo de comisión

Calcular la comisión de un vendedor, basada en sus ventas mensuales.

Cantidad de meses entre dos fechas del mismo año

Promedio de calificaciones

Validación de mayoría de edad

Clasificación de triángulos

Es equilátero, isósceles o escaleno?

Calculator área de un cuadrado

Calcular perímetro de un hexágono

Imprime pirámide de N pisos

Calculadora de IMC (Índice de Masa Corporal)

Detección de secuencias en un arreglo

Conteo regresivo

Imprime los números entre x y cero

Suma de números pares hasta N

Listado de números impares

Imprime números impares desde 0 a x

Validación de email

Tiene arroba?

Conversión de segundos a horas, minutos y segundos

Cálculo de descuentos por cantidad

Menos de \$1,000 no tiene descuento. Mas de \$1,000 2% descuento. Mas de \$10,000 5%

Análisis de un número (positivo, negativo, cero)

Encuentra el número repetido en un arreglo

Materiales de referencia

Diseñar diagramas de flujo:

<https://app.diagrams.net/>

Muchas gracias
por su atención!

