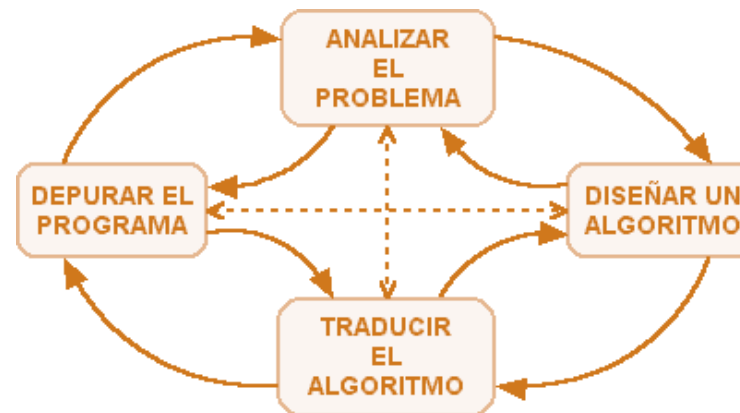


# ALGORITMOS Y PROGRAMACIÓN



# Índice

- Solución de Problemas
- Fases para realizar un Programa



- Lenguajes de Programación



# Qué es un Problema

Es una situación en la que se trata de alcanzar una **meta** y para lograrla se deben hallar unos **medios** y unas **estrategias**

## Elementos:

- Estado inicial
- Meta (submetas): descomponer problema
- Recursos
- Dominio

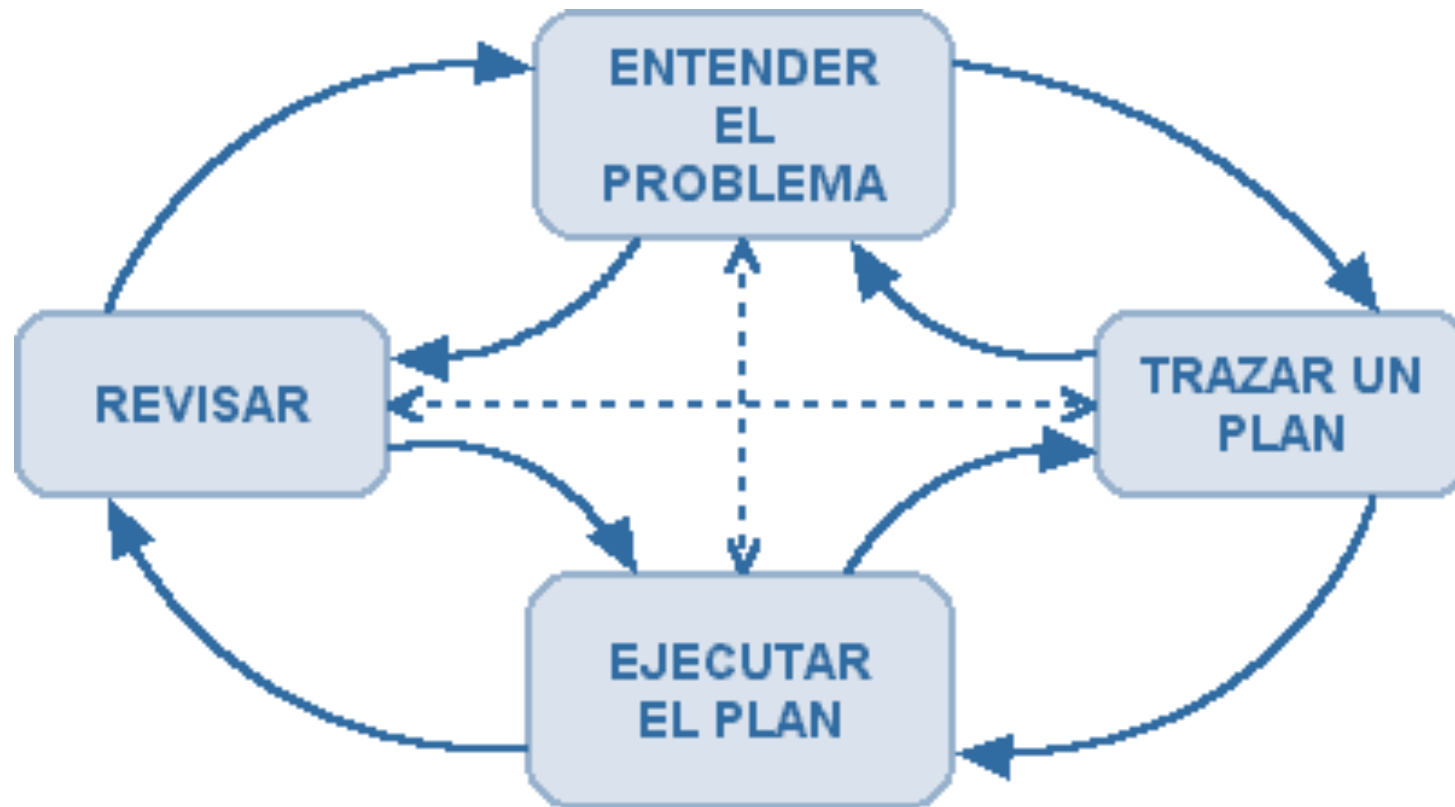


# Estrategias

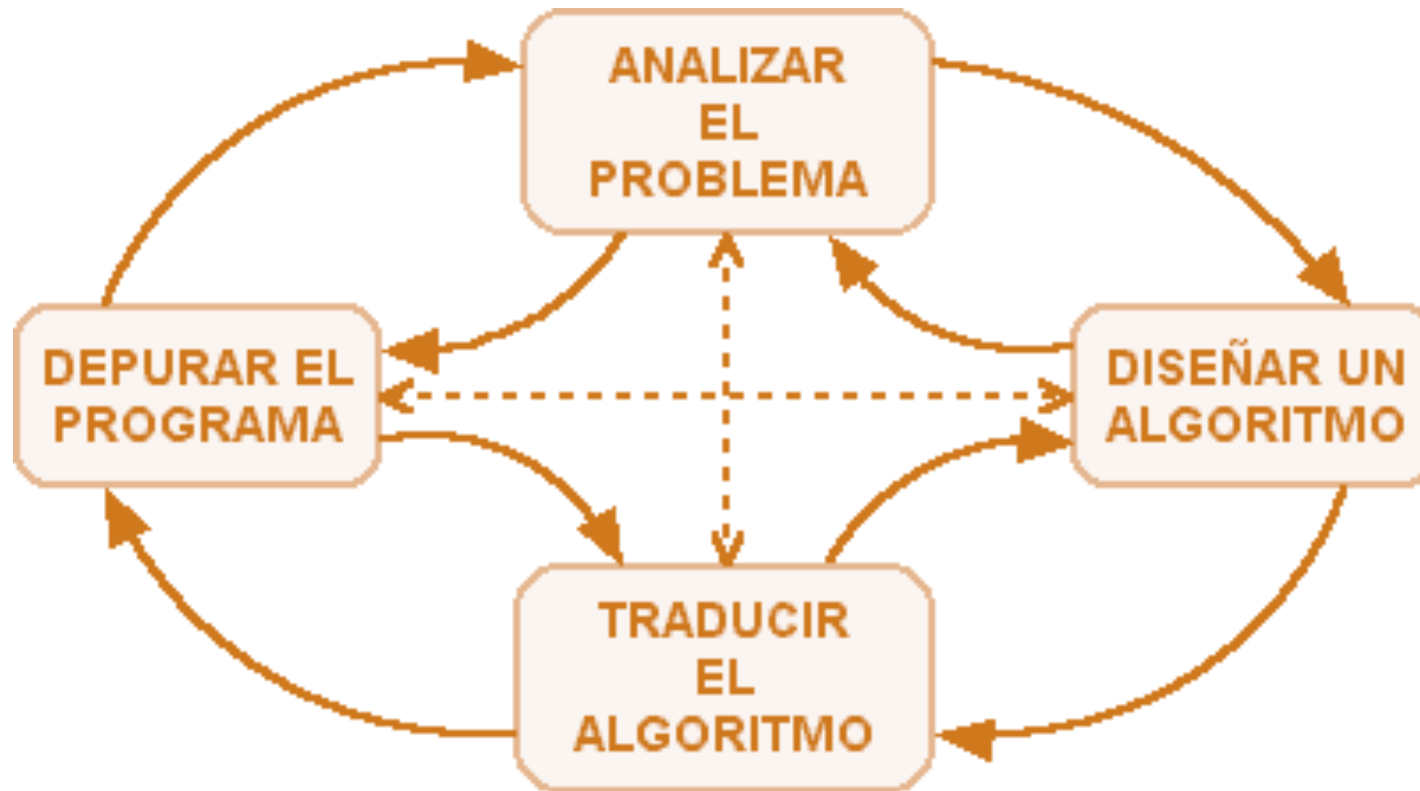
- Ensayo y error
- Iluminación
- **Heurística**
- **Algoritmos**
- Razonamiento Analógico
- Lluvia de ideas
- Pensamiento lateral



# Método Resolver Problema



# Fases Realizar un Programa



# Analizar Problema

- Formular el problema
- Precisar los resultados esperados: metas y submetas
- Búsqueda de información
- Identificar datos disponibles: estado inicial y restricciones
- Descomponer el problema en subproblemas



# Creatividad

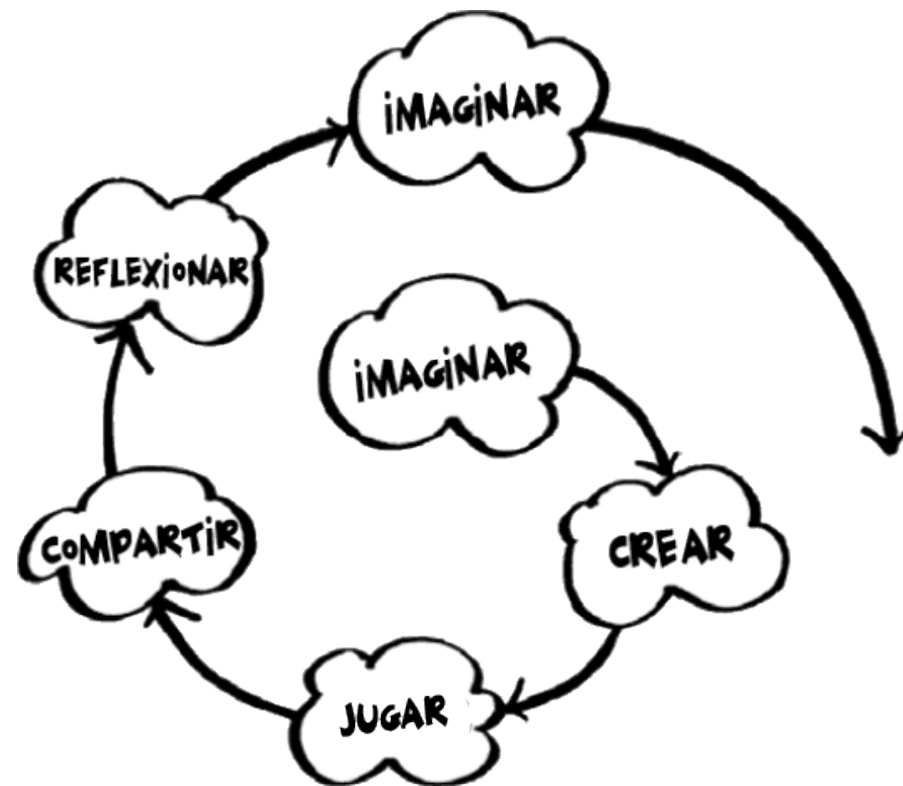
Es la capacidad de inventar algo nuevo, de relacionar algo conocido de forma innovadora o de apartarse de los esquemas de pensamiento y conducta habituales

Personas creativas: buen humor; confianza en sí mismos; flexibilidad y adaptabilidad; alta capacidad de asociación; sensibilidad; curiosidad intelectual; percepción y observación agudas; iniciativa para tomar riesgos; imaginación; expresividad; capacidad crítica; entusiasmo; y, tenacidad





# Espiral de pensamiento Creativo



# Diseñar un Algoritmo

- Establecer procesos:
  - Que procesos necesito
  - Como se interrelacionan
- Resolver los subproblemas aislados
- Crear Algoritmos pseudocódigo
- Diagrama de flujo

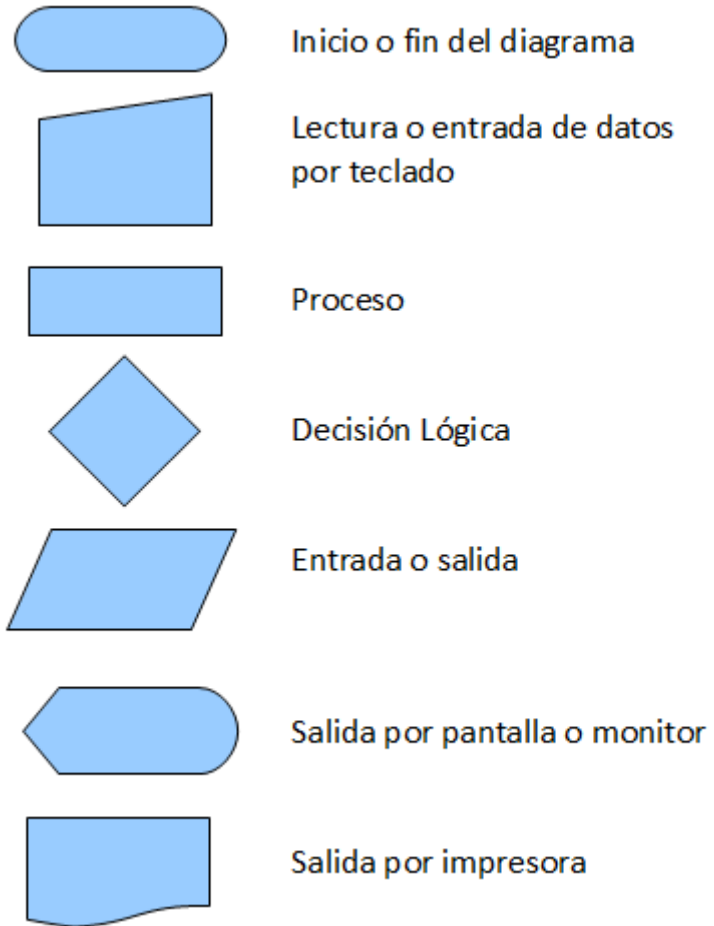


# Que es un Algoritmo

- ¿Que es?  
Secuencia operaciones lógicas
- Estructuras
  - Secuencial
  - Iterativa
  - Condicional



# Simbología Diagrama de Flujo



# Algoritmo Pseudocódigo-Diagrama Flujo

Paso 1: Inicio

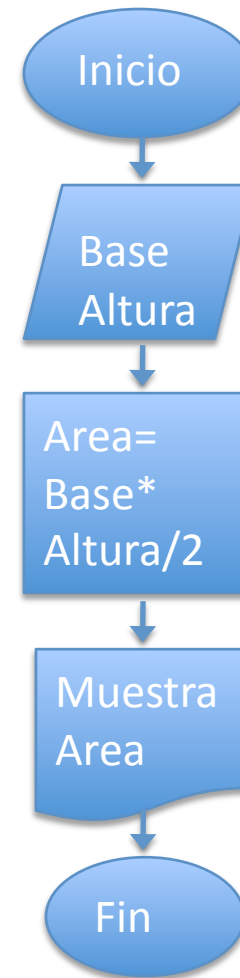
Paso 2: Asignar el número 3 a la constante "base"

Paso 3: Asignar el número 4 a la constante "altura"

Paso 4: Guardar en la variable "área" el resultado de  $\text{base} * \text{altura} / \text{div}$

Paso 5: Imprimir el valor de la variable "área"

Paso 6: Final



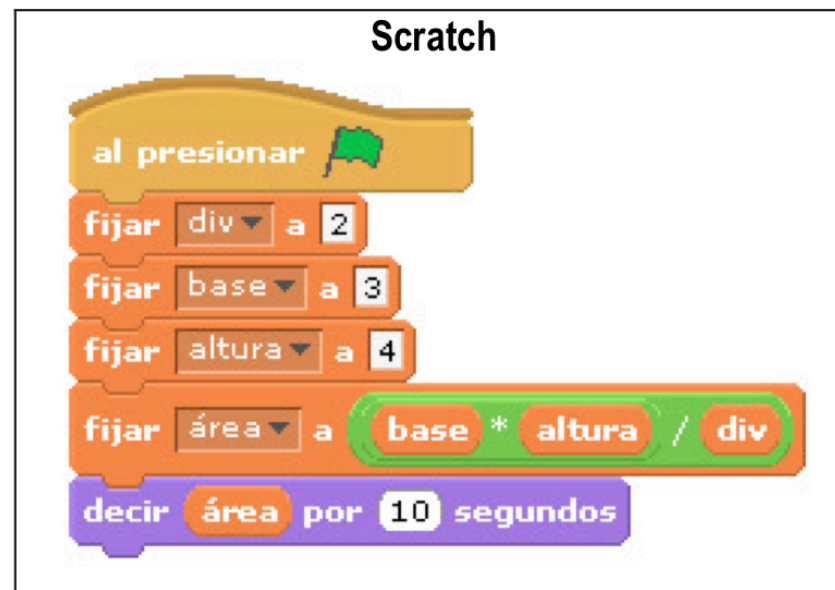
# Diagrama de Flujo

- ¿Qué es? Es una representación gráfica de un algoritmo
- Ventajas
  - Forzar la identificación de todos los pasos de una solución de forma clara y lógica
  - Establecer una **visión** amplia y objetiva de la **solución**
  - Verificar si se han tenido en cuenta todas las posibilidades
  - **Comprobar** si hay procedimientos **duplicados**
  - **Representar gráficamente** una solución (es más simple hacerlo con gráficas que mediante palabras)
  - Facilitar a otras personas la **comprensión** de la secuencia lógica de la solución planteada



# Traducir Algoritmo

- Ejemplo con Scratch



# Depurar Programa

- **Dificultad** para elaborar procedimientos perfectos en los **primeros intentos**
- La **dificultad** aumenta a medida que los **problemas** se vuelven más **complejos**
- Los **resultados** se deben probar y **validar** (revisión)
- Fallo **Sintaxis**
- Proceso de **deconstrucción**





# Lenguajes de Programación

- Tipos
- Conceptos Básicos
- Estructuras básicas



# Tipos de Lenguajes de Programación

- Según nivel
  - Lenguajes de bajo nivel
    - Lenguaje máquina: 0-1s
    - Lenguaje ensamblador: Números-letras
  - Lenguajes alto nivel
- Código / Bloques

```
Blink
/*
  Blink
  Turns on an LED on for one second, then off for one second,

  This example code is in the public domain.
  */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);            // wait for a second
}
```



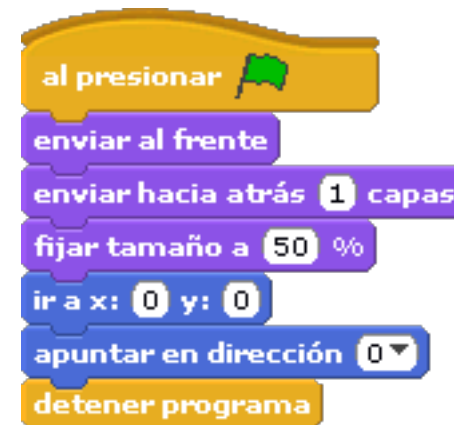
# Conceptos Básicos

- Tipos de datos
  - Números: *0; 25;*
  - Letras: *"Cual es tu nombre"*
  - Cadenas: *números, letras*
- Variables: *velocidad*
- Contadores: *velocidad = velocidad + 1*
- Operadores / Operadores Lógicos



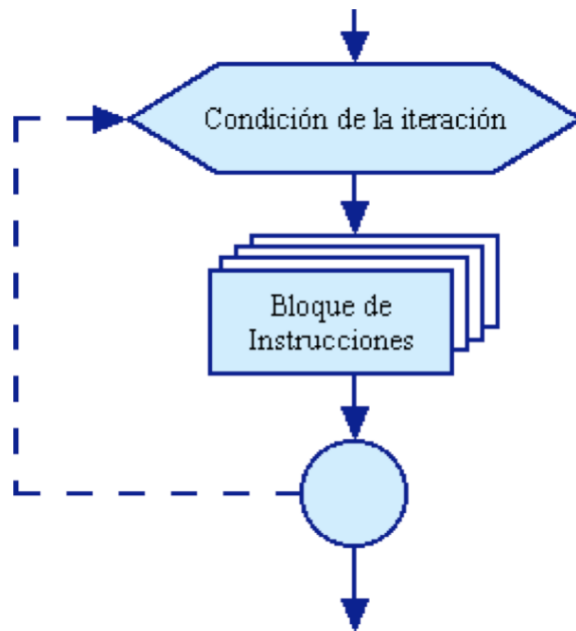
# Estructuras Básicas

## Secuencial



# Estructuras Básicas

## Iterativa



# Estructuras Básicas

Condición simple      Condición doble

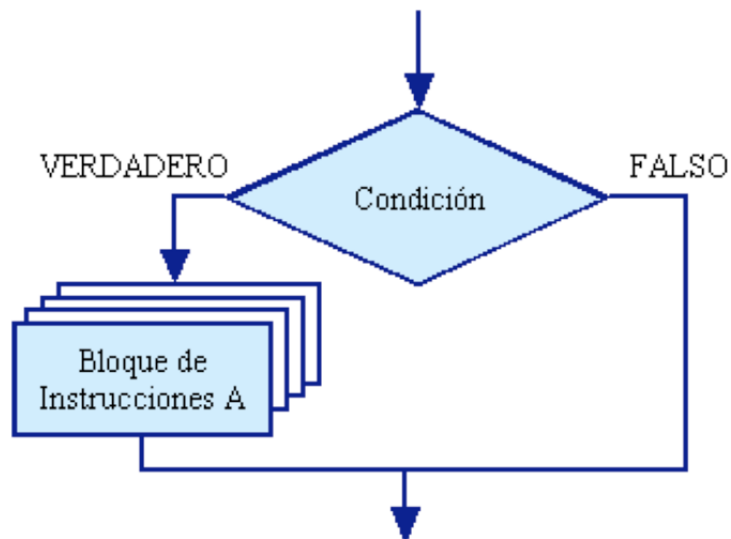
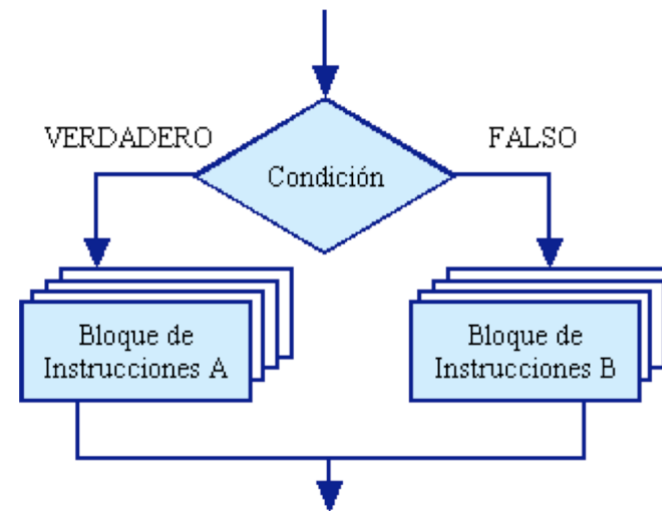
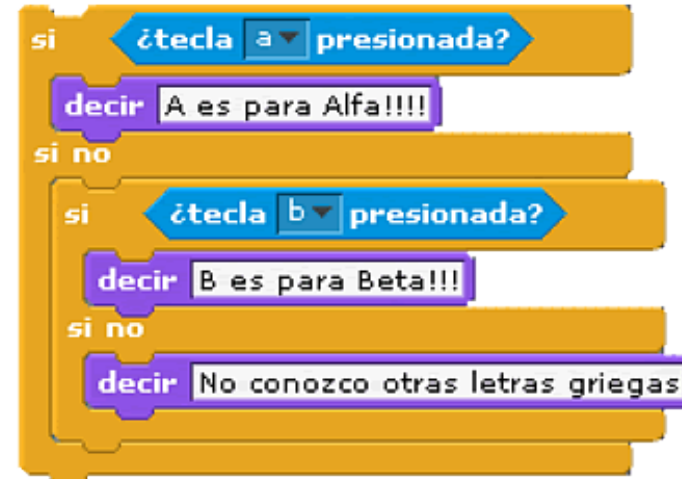


Ilustración 3-9: Modelo de estructura condicional simple.



# Estructuras Básicas

Condicional selección simple      Condicional selección doble anidado



# Bibliografía

Presentación Basada en el Material:  
“Algoritmos y Programación” de Juan  
Carlos López García. Eduteka

