



# Estructura de Datos y Algoritmos

Sesión 1

Fundamentos de Programación

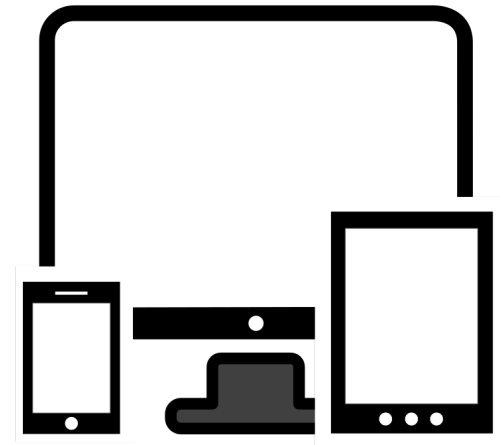
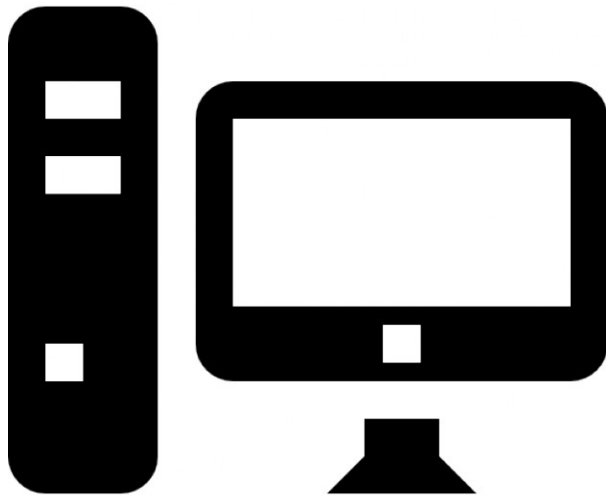


## Agenda

- Introducción
- Algoritmos
- Tipos de Datos
- Selectivos
- Repetitivos
- Arreglos

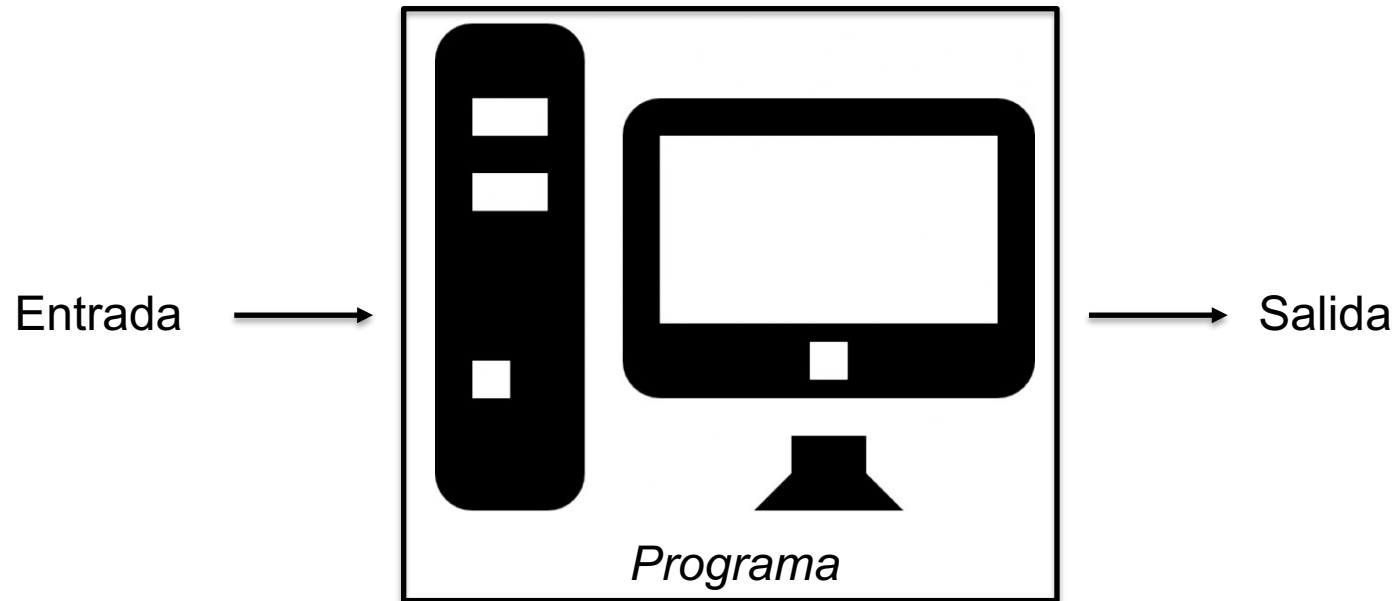


## El computador



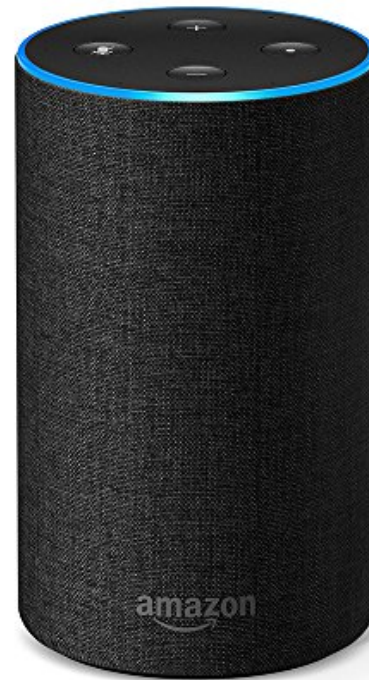


## El computador





## El computador





## El computador





## ¿Qué es un algoritmo?

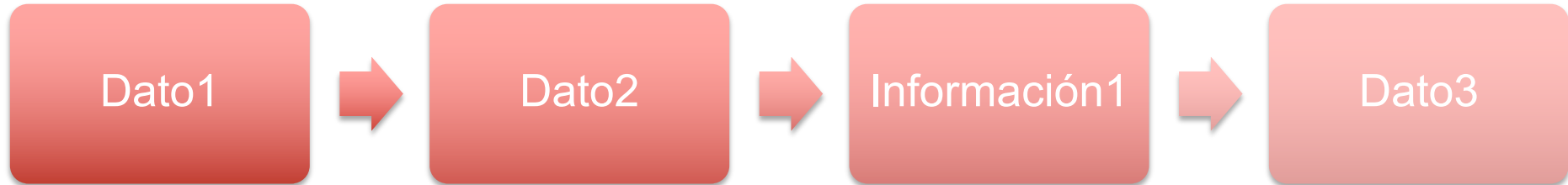
Secuencia de pasos lógicos y ordenados que se siguen para solucionar un problema determinado.





## ¿Qué es una estructura de datos?

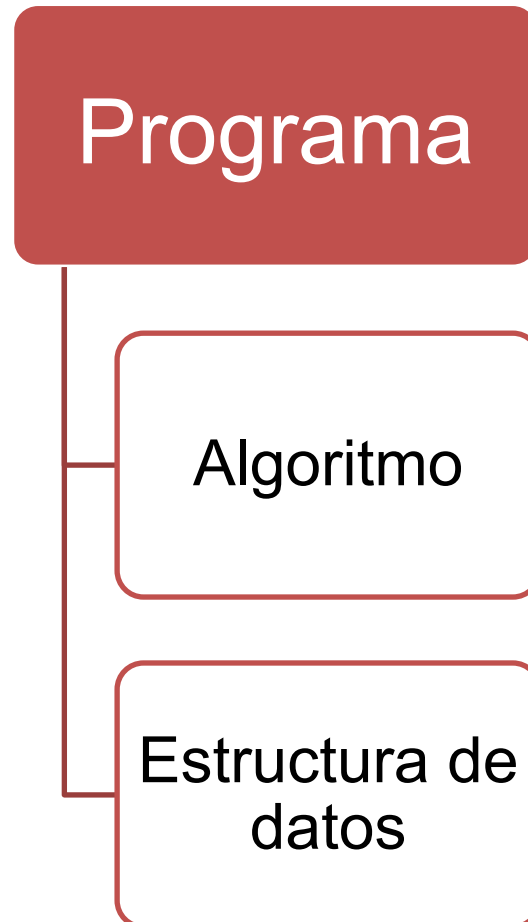
Disposición de los datos y la información en la memoria del computador.







## ¿Qué es un programa?





## Características

**Preciso y Exacto:** Indicar el orden de realización de cada paso, claro y sin ambigüedad.

**Definido:** Si se sigue el algoritmo dos veces, se debe obtener el mismo resultado cada vez.

**Finito:** Si se sigue se debe terminar en un momento. Debe tener un número finito de pasos.

**Eficiente:** Cada instrucción puede ser verificada por una persona con una prueba manual que satisfaga los requerimientos planteados por el problema.



## Tipos de datos

NOMBRE	CONJUNTO DE VALORES	OPERACIONES
Enteros	Negativos y positivos sin decimal	Sumar, restar, dividir, multiplicar, residuo
Reales	Negativos y positivos, con decimal	Sumar, restar, dividir, multiplicar
Lógicos	Verdadero o Falso(1 o 0)	And, Or, Not
Caracteres	Letras, números, especiales, juntos forman una cadena	Concatenar



## Venta de pasajes

A continuación se presenta una lista de precios de una aerolínea desde Lima hacia:

- Arequipa:
  - Ejecutivo: S/. 600
  - Turista: S/. 300
- Tumbes
  - Ejecutivo: S/. 800
  - Turista: S/. 350

Determinar el precio de un pasaje dado un destino y un tipo de asiento.



## Estructura selectiva IF

```
if (evaluación) {  
    acción 1  
    acción 2  
    ...  
}  
[else{  
    acción 3  
    acción 4  
    ...}]
```



## Estructura selectiva IF

```
if (evaluación1) {  
    acción 1  
    ...  
}  
[else if (evaluación2) {  
    acción 2  
}]  
[else{  
    acción 3  
}]
```



## Estructura selectiva IF

### Comparación

*Evaluación1 == Evaluación2*

*Cuando es String: cadena.equals("")*

### Operadores lógicos

*(Eva1 == Eva2) && (Eva3 < Eva4)*

*(Eva1 == Eva2) || (Eva3 == Eva4)*



## Estructura selectiva IF

### Opciones de evaluación

*Evaluación1 == Evaluación2 (igualdad)*

*Evaluación1 > Evaluación2 (mayor)*

*Evaluación1 >= Evaluación2 (mayor igual)*

*Evaluación1 < Evaluación2 (menor)*

*Evaluación1 <= Evaluación2 (menor igual)*

*Evaluación1 != Evaluación2 (desigualdad)*





## Estructura selectiva IF

A	B	A && B	A    B
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	F



## Estructura selectiva SWITCH

```
Switch (variable) {  
    case valor1:  
        acción 1;  
        break;  
    case valor2:  
        acción 2;  
        break;  
    default:  
        acción 3  
}
```



## Número Capicua

- Desarrollar un programa para que dado un número de **4 cifras** obtener como resultado el número invertido de dicho número

4591  $\Rightarrow$  1954

- ¿El número es capicua?. Un número es capicua cuando es igual a su número invertido:

4591  $\Rightarrow$  ¿es capicua 1954?



## Estructura repetitiva FOR

```
for (int i = 0; i <= X; i++) {  
    acción 1  
    acción 2  
    ...  
}
```

Donde X es n-1 repeticiones solicitadas

Ejemplo: si se solicita 20 repeticiones, x sería 19.



## Número Capicua

- Desarrollar un programa para que dado un número de **n cifras** obtener como resultado el número invertido de dicho número

4591  $\Rightarrow$  1954

- ¿El número es capicua?. Un número es capicua cuando es igual a su número invertido:

4591  $\Rightarrow$  ¿es capicua 1954?



## Estructura repetitiva WHILE

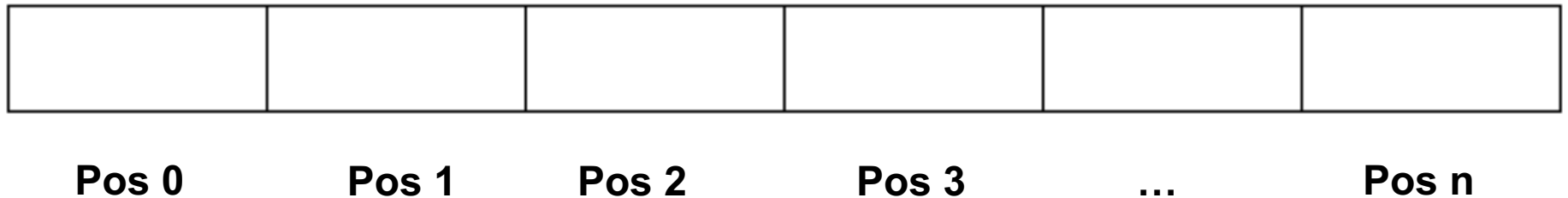
```
while (evaluación){  
    acción 1  
    acción 2  
    ...  
}
```

Un ciclo WHILE se repite ilimitadamente hasta que el resultado de la evaluación sea **FALSE**.



## Arreglos

- Un arreglo es una estructura de datos que permite almacenar una serie de valores de un tipo de dato específico.
- Usualmente, un arreglo se representa así:





## Arreglos

- Ejemplo: notas del curso de fundamentos de programación:
  - variable1 = PC1
  - variable2 = PC2
  - variable3 = Participación
  - variable4 = EF
- Con arreglos:
  - arreglo = [PC1, PC2, Part, EF]

<b>PC1</b>	<b>PC2</b>	<b>Part</b>	<b>EF</b>
------------	------------	-------------	-----------





## Arreglos

- Ejemplo: notas del curso de fundamentos de programación:
  - `variable1 = PC1 = arreglo[0]`
  - `variable2 = PC2 = arreglo[1]`
  - `variable3 = Participación = arreglo[2]`
  - `variable4 = EF = arreglo[3]`



## Arreglos

- Funciones principales
  - length: devuelve el tamaño del arreglo
- Elementos al arreglo
  - Primer elemento: arreglo[0] = 5;
  - Segundo elemento: arreglo[1] = 6;



## Arreglos

```
for (int i = 0; i < arreglo.length; i++) {  
    acción 1  
    acción 2  
    ...  
}
```



**Gracias...**

