

# Sesión 1: La lógica de la Programación

Jose Rodney Menezes De la Cruz.



[rodney.menezes@pucp.edu.pe](mailto:rodney.menezes@pucp.edu.pe)

Taller: Rstudio Aplicado a Finanzas  
Colegio de Economistas de Loreto

CELOR, 2019

# Índice

1. Principios Básicos de Programación.
2. ¿Qué necesito para programar?
3. Las matemáticas en la programación.
4. Los operadores matemáticos en programación.
5. Las Metodologías en la Programación.
6. Fundamentos de Programación.
  - Conceptos Elementales.
  - Tipo de datos y variables.
  - Estructuras If, Else y Switch.
  - Operadores Aritméticos, Relacionales y Lógicos.
  - Bucles, Arrays y Matrices.
  - Algoritmo, Diagramas de flujo y Pseudocódigo.



# LA LÓGICA DE PROGRAMACIÓN





# Principios Básicos De Programación



Programar **NO** es cosa  
del otro mundo

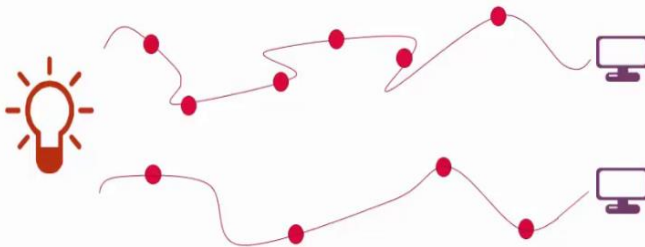


# Principios Básicos De Programación

## Principios Básicos De Programación

### La Lógica

Tener idea del camino más simple y más fácil, que escogemos para lograr nuestro objetivo.





## Principios Básicos De Programación



### El Programa

Un programa es un conjunto de instrucciones que al ser ejecutadas logran un objetivo.

```

        $resource_id' => $resource_details['role_id'];
    );
    $this->rule_exists( $resource_details['id'], $role_id );
    ( $access == false ) {
        // Remove the rule as there is currently no need for it
        $details['access'] = false;
        $this->sql->delete( 'acl_rules', $details );
    } else {
        // Update the rule with the new access value
        $this->sql->update( 'acl_rules', array( 'access' => $access ) );
    }
    foreach( $this->rules as $key=>$rule ) {
        if ( $details['role_id'] == $rule['role_id'] && $details['access'] == false ) {
            if ( $access == false ) {
                unset( $this->rules[ $key ] );
            } else {
                $this->rules[ $key ]['access'] = $access;
            }
        }
    }
}

```

# Principios Básicos De Programación



## La Programación

Una serie de pasos o instrucciones para dar una solución al problema.





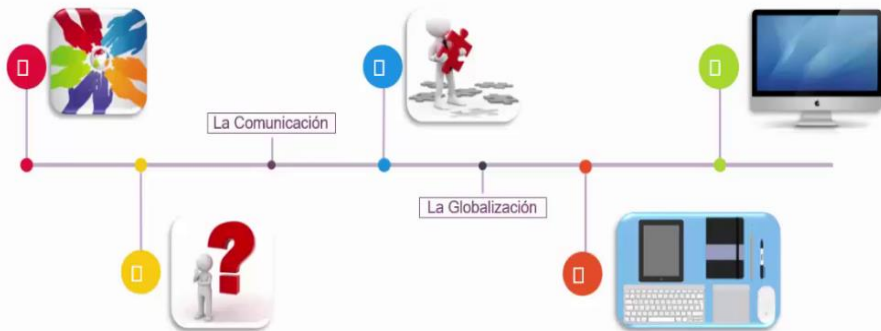
¿Qué necesito para  
programar?



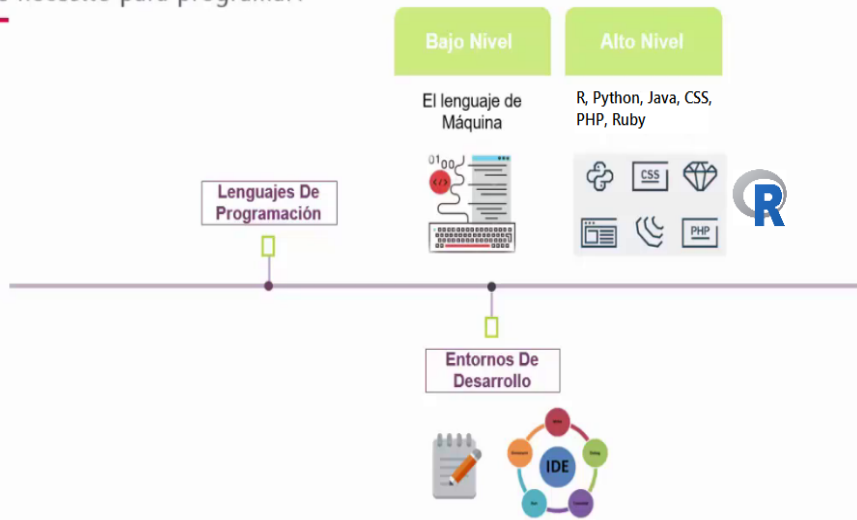
# Mentalidad Abierta



## ¿Qué necesito para programar?



## ¿Qué necesito para programar?





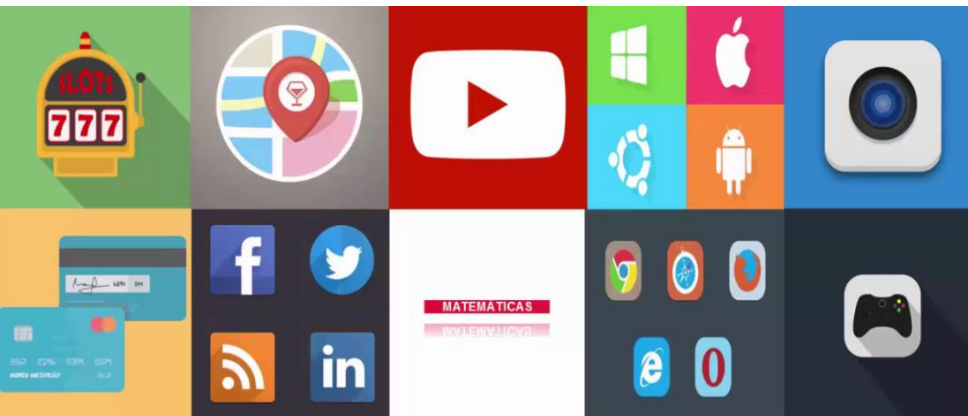
# ¿Necesitamos las matemáticas para programar?



## ¿Necesito las matemáticas para programar?







## La importancia de las Matemáticas



# Los operadores matemáticos en programación



## Los operadores matemáticos en programación



### Operador

Son signos que sirven para relacionar:

Constantes

Variables

Expresiones

## Los operadores matemáticos en programación



## Los operadores matemáticos en programación



### Reglas al programar expresiones matemáticas



Una expresión ha de ser lineal

## Los operadores matemáticos en programación



### Reglas al programar expresiones matemáticas



#### El orden jerárquico

Potencia	^
Multiplicación y División	* /
Suma y Resta	+ -

$$A = b + c * d - e$$

## Los operadores matemáticos en programación



### Reglas al programar expresiones matemáticas



#### El uso de paréntesis

$$A = (b + c) * d$$

## Los operadores matemáticos en programación



### Reglas al programar expresiones matemáticas



Los paréntesis interiores son prioritarios

$$A = ( (b + c) * 4 ) * 5 + 3$$



## Los operadores matemáticos en programación

$$x = \frac{2+8}{1+5}$$

$$2+8/1+5 \quad \text{X}$$

$$(2+8)/(1+5)$$

## Los operadores matemáticos en programación

$$x = \frac{2 * 8 - 1}{3 + \frac{10}{2^3}} \quad \begin{array}{l} ((2 * 8) - 1) / (3 + (10 / 2^3)) \\ (2 * 8 - 1) / (3 + 10 / 2^3) \end{array}$$



# Las metodologías en programación



## Las metodologías en programación

### Metodología

Una serie de pasos para llegar a un objetivo planteado.



## Las metodologías en programación

---

OBJETIVO

El problema

ALGORITMO

La solución

DEPLOY

Despliegue

TESTING

Probar



# FUNDAMENTOS DE PROGRAMACIÓN





# Conceptos Elementales



## Conceptos Elementales



### Atributo

Característica de una propiedad que diferencia a un objeto del otro.



### Dato (Valores)

Características del atributo.



## Conceptos Elementales



ATRIBUTO	VALORES
Nombre	Javier
Edad	24
Profesión	Ingeniero Informático
Gustos	Fútbol y Programación

## Conceptos Elementales



Registros

## Conceptos Elementales



## Conceptos Elementales

Nombre	Edad	Profesión	Gustos
Javier	24	Ingeniero	Fútbol
Pedro	23	Arquitecto	Natación
Gabriela	24	Profesora	Baloncesto
Ana	25	Directora	Leer

## Conceptos Elementales

## Archivo

Atributos

	Nombre	Edad	Profesión	Gustos
Datos	Javier	24	Ingeniero	Fútbol
	Pedro	23	Arquitecto	Natación
	Gabriela	24	Profesora	Baloncesto
		25		Leer



Registros



# Tipos de Datos



## Tipos de Datos



### Tipo de dato

Característica del dato que le dice al ordenador que dato va a procesar.

## Tipos de Datos



### Entero



Sin decimales



Rango de negativo a positivo

int

-86

1

-37

259

$$7 / 2 = 3.5$$

$$7 / 2 = 3$$



$$1.5 + 4.2 = 5.7$$

$$1.5 + 4.2 = 5$$



## Tipos de Datos

### Flotante

-  Números con parte decimal
-  Rango de negativo a positivo

float

-7.9      -3.701      2.59      -1.0      4 / 2 = 2.0      8 + 1 = 9.0

## Tipos de Datos



### Caracter



Cualquier símbolo o letra

**char**

**string**

? a ç ! @

**ROJO**

**ROJO**

Cadena de caracteres

## Tipos de Datos



 Poseen dos valores: TRUE o FALSE



# Variables y Constantes



# Variables y Constantes



## Variables

Lo que varía o cambia.

Son lugares de almacenamiento que pueden contener información de distintos tipos  
Por ejemplo: texto, números, letras etc

## DATOS

# Variables y Constantes

## Variables

### Numéricas

Valores numéricos  
positivos o negativos

### Alfanuméricas

Letras números y  
caracteres especiales

### Lógicas

Tipo de datos  
Booleanos

## Tipos de Datos



### Constantes

Un dato numérico o alfanumérico que no cambia durante la ejecución de un programa.

 $\pi$ 

3.1415

## Tipos de Datos



### Expresiones

Combinaciones de constantes, variables, símbolos de operación, paréntesis y nombres de funciones.

$$A + (b + 2) / d - 9$$





# Estructuras IF ELSE y SWITCH

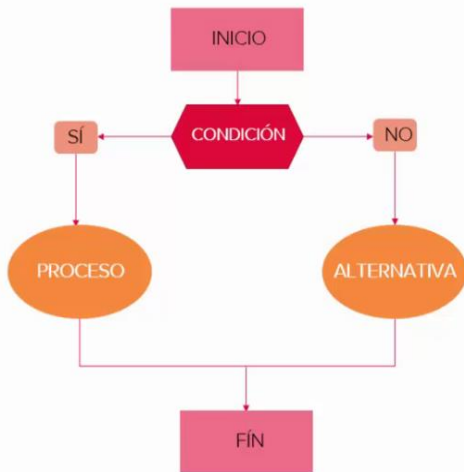


## Estructuras If Else y Switch

### IF ELSE

IF → SI

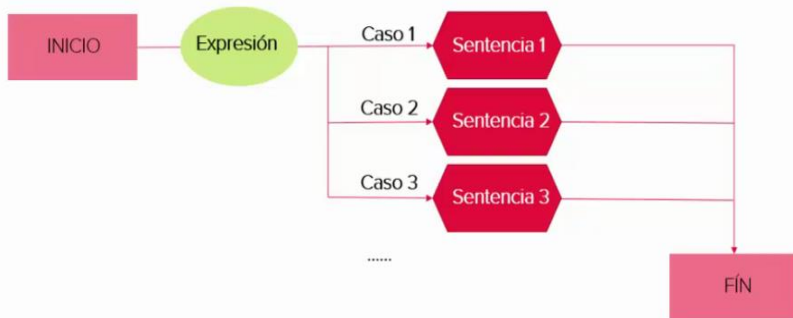
ELSE → SINO



## Estructuras If Else y Switch



### SWITCH





# Operadores Aritméticos



## Operadores Aritméticos



### Aritméticos

Operador	Uso
+	Adición- Suma los valores.
-	Resta- Resta los valores.
*	Multiplicación- Multiplica los valores.
/	División - Divide los valores.
%	Módulo - Devuelve el resto.
++	Incremento - Aumenta el valor en 1.
--	Decremento - Disminuye el valor por 1.



# Operadores Relacionales



## Operadores Relacionales



### Relacionales

Operador	Uso
<code>==</code>	Comprueba si dos valores son iguales.
<code>!=</code>	Comprueba si dos valores no son iguales.
<code>&gt;</code>	Comprueba si el valor izquierdo es mayor.
<code>&lt;</code>	Comprueba si el valor izquierdo es menor.
<code>&gt;=</code>	Comprueba si el valor valor izquierdo es mayor o igual que el derecho.
<code>&lt;=</code>	Comprueba si el valor valor izquierdo es menor o igual que el derecho.



# Operadores Lógicos





## Operadores Lógicos

**AND &&**

**Elemento 1**

**Elemento 2**

Verdadero

AND

Verdadero

Verdadero

Verdadero

AND

Falso

Falso

Falso

AND

Verdadero

Falso

Falso

AND

Falso

Falso

## Operadores Lógicos

OR ||

## Elemento 1

## Elemento 2

Verdadero

OR

Verdadero

Verdadero

Verdadero

OR

Falso

Verdadero

Falso

OR

Verdadero

Verdadero

Falso

OR

Falso

Falso

## Operadores Lógicos

**NOT!**

NOT

Verdadero



Falso

NOT

Falso



Verdadero

## Sesión 1

Navigation icons: back, forward, search, and other controls.

CELOR, 2019

60 / 82



# Bucles, Arrays y Matrices



## Bucles, Arrays y Matrices



### Bucle

Los bucles o también llamados CICLOS se utilizan para ejecutar un conjunto de instrucciones varias veces basándose siempre en una condición que decidirá si se sigue repitiendo o no.

## Bucles, Arrays y Matrices



### Bucle WHILE

Se utiliza cuando queremos repetir la ejecución de unas sentencias, un número indefinido de veces, siempre que se cumpla una condición.

```
x=5;  
while ( x > 0 ) {x * 2;}  
  
System.out.println("Ahora x vale " +x);
```

## Bucles, Arrays y Matrices



### Bucle DO WHILE

Se utiliza generalmente cuando no sabemos cuantas veces se habrá de ejecutar el bucle.

```
x=5;  
  
do ( x * 2 )  
  
while ( x > 0)  
  
System.out.println("Ahora x vale " +x);
```

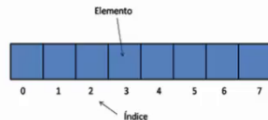


## Bucles, Arrays y Matrices



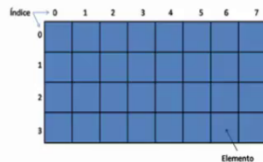
### Array

Un conjunto de datos que se almacenan en memoria de manera seguida con el mismo nombre.



### Matrices

Son arreglos con dos dimensiones y su manejo es igual.





# ¿Qué es un Algoritmo?



## ¿Qué es un Algoritmo?



### Algoritmo

Una serie de pasos o instrucciones finitas y ordenadas que sirven para dar solución al problema.

Inicio

-  
-  
-  
-  
-  
-  
-

Fin

## ¿Qué es un Algoritmo?

 Algoritmo

 Cualitativo

 Cuntitativo

## ¿Qué es un Algoritmo?



### **Cualitativo**

Aquello en el que NO intervienen cálculos numéricos.



La SECUENCIA de acciones



La DECISIÓN de acciones



El CICLO de acciones

## ¿Qué es un Algoritmo?

### □ La secuencia de acciones:

Seguir una secuencia y orden.

INICIO

- 1- Marcar el número.
- 2- Presionar el botón de llamada.
- 3- Hablar
- 4- Terminar la llamada.
- 5- Guardar el número en la lista de contactos.

FIN

## ¿Qué es un Algoritmo?

### La decisión de acciones:

Decidir el camino lógico.

INICIO

Si (if)

(condición)

Entonces

(condición)

Sino (else)

(condición)

FÍN

INICIO

1- Marcar el número.

2- Si (if)

Se escuchan varios tonos (el número está ocupado)

3- Entonces

Colgar

4- Sino (else)

Hablar

Terminar la llamada

Guardar el número

FÍN

## ¿Qué es un Algoritmo?

### El ciclo de acciones:

Estructuras que sirven para repetir.

Mientras / Hasta

Do While

Repetir / Hasta

Repeat While

INICIO

1- Marcar el número.

2- Mientras (while)

Se escuchan varios tonos (el número está ocupado)

Entonces

3-Repetir (repeat)

4- Sino (else)

Hablar

Terminar la llamada

Guardar el número

FÍN



## ¿Qué es un Algoritmo?



### Cuantitativo

Aquello en el que SÍ intervienen cálculos numéricos.



ENTRADA



PROCESO



SALIDA

## ¿Qué es un Algoritmo?



### La Entrada

Introducir los datos.

Leer 3 números  
N1, N2, N3



### El Proceso

Indicar los pasos.

Sumar 3 números  
 $N1 + N2 + N3$



### La Salida

Muestra de los resultados.

Resultado de la suma  
N1, N2, N3



# Diagramas de Flujo Y El Pseudocódigo



## Diagramas de flujo y el Pseudocódigo

# ALGORITMO

**CICLO**



Repetición necesaria.

**DECISIÓN**



Tomar el camino lógico.

**SECUENCIA**



Ejecución de acciones en una orden.

## Diagramas de flujo y el Pseudocódigo



### Pseudocódigo

Una forma de representar el algoritmo de manera que lo podamos entender más fácilmente.

Secuencia

Decisiones

Ciclo

## Diagramas de flujo y el Pseudocódigo

### INICIO

Edad: Entero (int)

Escribe ¿Cuántos años tienes? (string)

Lee Edad

Si (if) Edad  $\geq$  18 entonces

    Muestra "Eres mayor de Edad."

Sino (else)

    Muestra "Eres menor de Edad."

### FÍN

## Diagramas de flujo y el Pseudocódigo

### INICIO

Edad: Entero (int)

Escribe ¿Cuántos años tienes? (string)

Lee Edad

Si (if) Edad  $\geq$  18 entonces

    Muestra "Eres mayor de Edad."

Sino (else)

    Muestra "Eres menor de Edad."

FÍN

### INICIO

Introducir la edad

Preguntar los años que tiene la persona

Leer el dato introducido

Si el dato introducido (la edad) es 18 o más de 18 (años)

    Escribir que eres mayor de edad

Mientras que si tiene menos de 18 años

    Escribir que eres menor de edad

FÍN

## Diagramas de flujo y el Pseudocódigo



### Diagramas de Flujo

Una forma de representar el algoritmo de manera más gráfica.

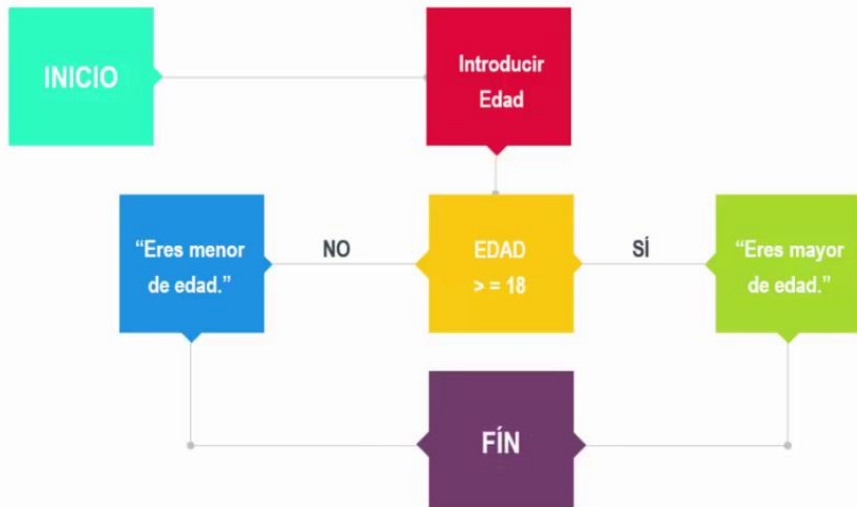
Secuencia

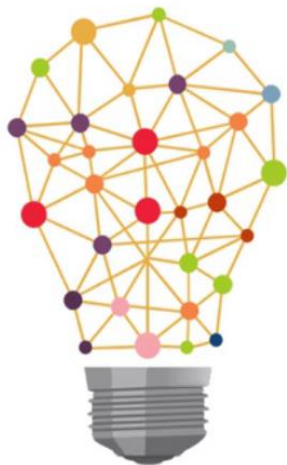
Decisión

Ciclo



## Diagramas de flujo y el Pseudocódigo





# LA LÓGICA DE PROGRAMACIÓN

