



**Argentina  
programa  
4.0**

# Introducción a Algoritmos y Java

---

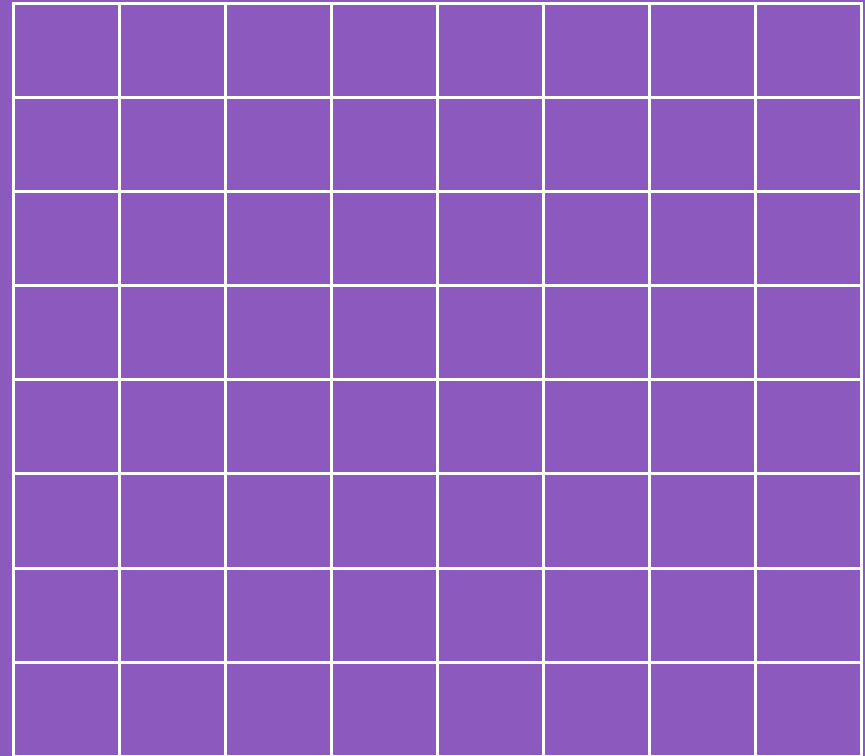
“Desarrollador Java Inicial”

# Agenda



- Concepto de Algoritmo
- Java - Características
- Sintaxis básica
  - Tipos Primitivos
  - Control de flujo
  - Vectores

# Algoritmo



# Concepto de algoritmo



Una de las tantas definiciones: instrucciones para resolver un cálculo o un problema abstracto, es decir, que un número finito de pasos convierten los datos de un problema (entrada) en una solución (salida)

Un algoritmo es una **serie de pasos organizados, que describe el proceso que se debe seguir, para dar solución a un problema específico.**

Un **algoritmo** es la descripción de cómo resolver un problema. Esta descripción se tiene que dar de manera finita (es decir, tiene que terminar) y ordenada.



# Otro ejemplo: Categoría Monotributo



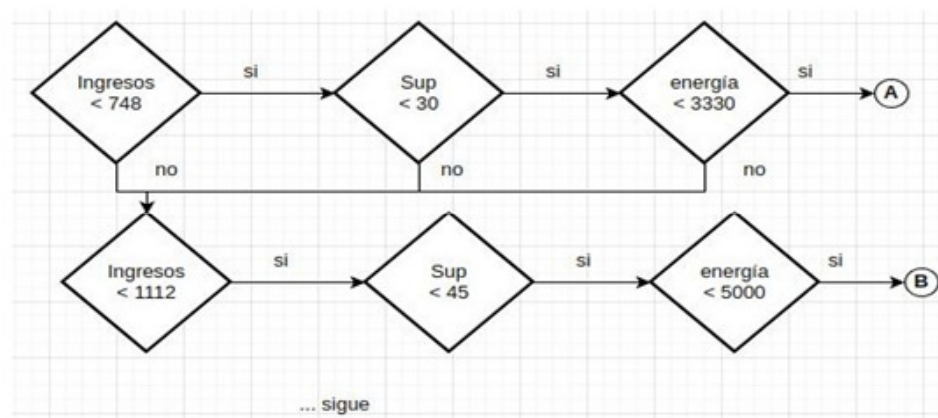
Problema: Determinar qué categoría del monotributo corresponde a una determinada persona

Entrada: Ingresos, superficie, energía eléctrica

Salida: Categoría

Categ.	Ingresos brutos	Sup. Afectada hasta:	Energía eléctrica consumida anualmente, hasta:
A	\$748.382,07	30 m2	3330 Kw
B	\$1.112.459,83	45 m2	5000 Kw
C	\$1.557.443,75	60 m2	6700 Kw
D	\$1.934.273,04	85 m2	10000 Kw
E	\$2.277.684,56	110 m2	13000 Kw

Determine a qué categoría pertenece cada ejemplo.  
¿Se anima a escribir el procedimiento?

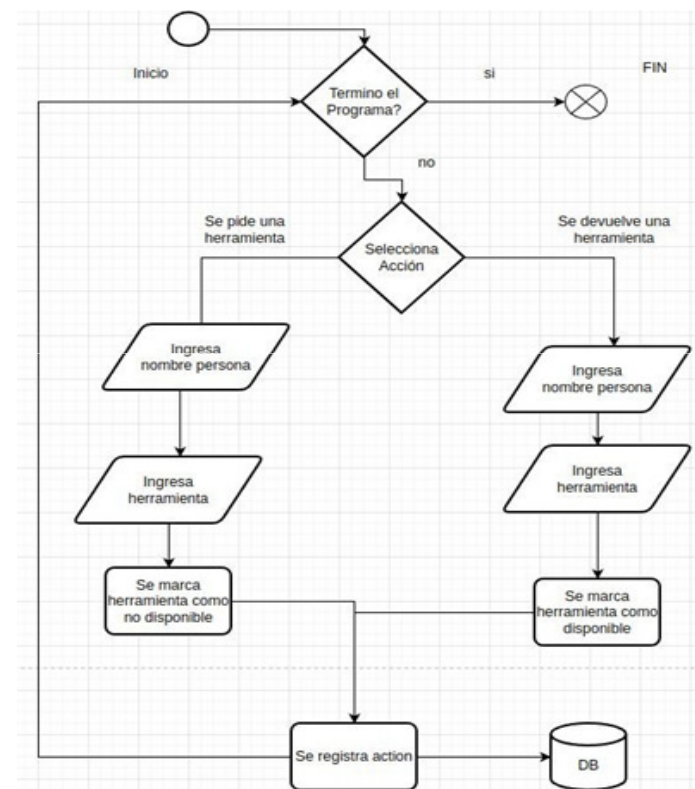


Ejemplo	Ingresos	Sup.	Energía	Cat?
Docente X	\$500.000,00	0	330 Kw	?
Carpintero X	\$1000000	30 m2	10000 Kw	?
Vendedor X	\$1.112.460	0	0	?

# Otro tipo, más interactivo

Problema: Llevar contabilidad de quien usa las herramientas de un taller

- Préstamo de herramienta
  - Entrada: Una persona se lleva una herramienta
  - Salida: eso queda registrado y la herramienta no está disponible para el siguiente
- Devolución:
  - Entrada: Una devuelve una herramienta que pidió prestada
  - Salida: La acción queda registrada y la herramienta está disponible para el siguiente que venga



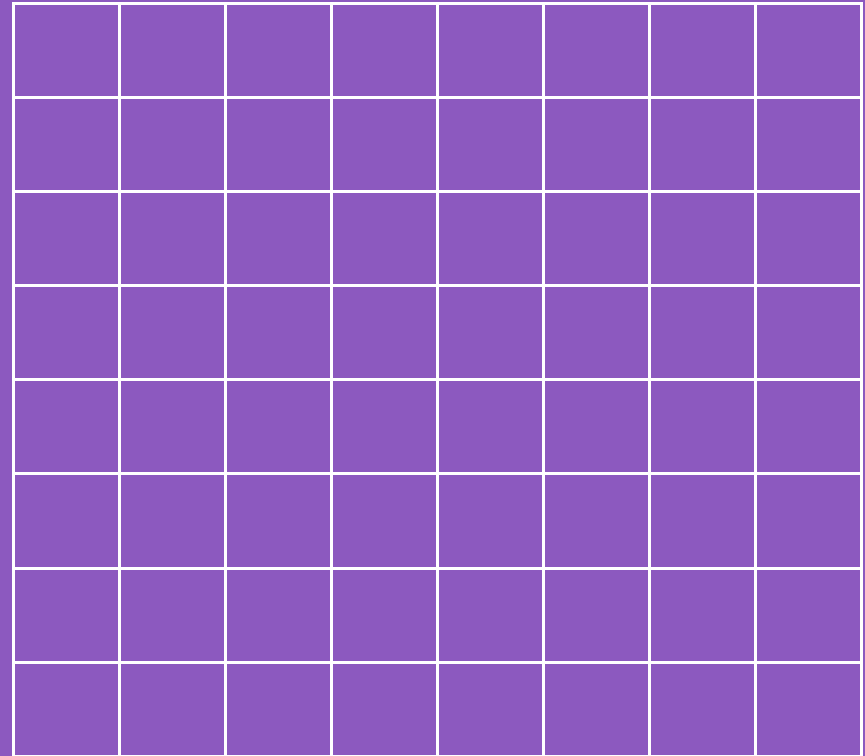
# ¿Cómo expresar el problema?



- Existen diversas formas de explicar y resolver algoritmos.
- Nosotros elegiremos implementarlos en un lenguaje de programación
- Para resolver el problema nos vamos a valer de instrucciones y variables
- **Variables:** lugares donde se guardan datos. Los mismos pueden ser desde algo “simple” como un número hasta una relación compleja de registros.
  - por ejemplo  $A = 1$
  - o nombre = “José”
- **Instrucciones:**
  - Asignaciones, operaciones y modificaciones de una variable:  $A = 1 + 1$
  - Evaluaciones y condicionales: elegir seguir por un camino o por otro, dependiendo de una condición. Por ejemplo si  $A > 1$  hacer una cosa y si no se cumple hacer otra
  - Ciclos o loops: mientras no se cumpla una determinada condición, continuar una determinada actividad.



# Java



# Java

- Lenguaje de desarrollo de propósito general
- Portable: Funciona en todos los sistemas operativos
- Gran Comunidad y base de un gran número de proyectos
- Orientado a Objetos
- Sencillo, seguro
- De alto nivel
- Compilado
- etc.

# Java - Sintaxis Básica - Variables y Tipos de datos primitivos



```
char unaLetra = 'a';  
  
boolean unValorBooleano = true;int  
    miPrimerContador = 66; double  
    unValor = 1.68;  
float otroNum = 2.344f;
```

**Variables:** es un espacio en memoria donde se puede almacenar un valor.

Una variable tiene nombre, un tipo , un valor. Se define, se declara, se inicializa, se utiliza en operaciones, etc.

**Operadores:**

**Aritmeticos:** + , -, \*, / , %

**Relacionales:** ==, !=, < >, >=, <=

**Asignacion:** =

**Logicos:** && (si logico - and) || (no logico - or) ! (no logico -not)

# Java - Sintaxis Básica - Operadores y Expresiones



## Op Binarias básicas

```
10 + 20
15 - 12
10 * 3
8 / 3
8 % 3
2 * 3
```

```
int miPrimerContador = 66;
double unValor = 1.68;

miPrimerContador + 20
15 - 12
10 * 3
unValor / 3
8 % 3
```

## Precedencia|

```
3 * 2 + 3
```

```
(3 * 2) + 3
```

# Java - Sintaxis Básica - Booleanos



## Operaciones y predicados

```
10 > 20  
15 >= 12  
10 == 3  
8 != 3
```

```
boolean unBooleano = true;  
boolean otroBooleano = false;
```

```
! unBooleano // false  
unBooleano && otroBooleano // false  
unBooleano || otroBooleano // true  
  
unBooleano && (otroBooleano || True) // t
```

```
int miPrimerContador = 66;  
double unValor = 1.68;  
double otroValor = 1.67;
```

```
unValor == (otroValor + 0.01)
```

Relacionar con condiciones  
"Monotributo"

# Java - Sintaxis Básica - Condicionales

```
if(unValor < otroNum) {  
    //una accion  
}  
if(unValor < otroNum) {  
    //una accion  
} else {  
    //otra accion  
}
```

```
char unaLetra = 'a';  
switch (unaLetra) {  
    case 'b':  
        //Hacer A  
        break;  
    case 'a':  
        //Hacer B  
        break;  
    default:  
        //Hacer Z  
}
```

```
unBooleano && (otroBooleano || True)  
if(unValor < otroNum) {  
    //una accion  
}  
  
int unValor = 1;  
int otroValor = 2;  
  
boolean unaCond = unValor == (otroValor + 1);  
  
if(unaCond) {  
    // hacer algo  
}
```

Relacionar con resolución  
de "Monotributo"

# Java - Salida Básica - System Out



Todos los lenguajes de programación tienen una forma de enviar información al usuario, ya sea mediante ventanas o lo que llamamos la "consola". En este curso, utilizaremos mucho mostrar contenidos de variables e información usando el comando:

```
System.out.println(...)  
//Ejemplos  
System.out.println(1) ;  
System.out.println('a') ;  
System.out.println(true) ;  
int x = 14;  
System.out.println(x) ;
```

Todavía falta para que comprendamos por qué hay puntos o que es "System", pero rápidamente, hay algo denominado "Clase", que representa conceptos de distinto tipo y ofrece funcionalidades varias.

Por ahora simplemente lo usaremos y en una pocas clases entenderemos la estructura de lo que estamos usando.

# Java - Sintaxis Básica - Bucles

```
while(condicionX) {  
    //Una Accion  
    //En algun momento se tiene  
    // modificar la condicionX  
}
```

```
for(inicia;condicionX;cambiaElemento) {  
    //Una Accion  
}
```

```
int unNum = 10;  
while(unNum > 0) {  
    System.out.println(unNum);  
    unNum = unNum -1;  
}
```

```
for(int otroNum=0;otroNum<10;otroNum++) {  
    System.out.println(otroNum);  
}
```

Relacionar con Programa de  
préstamo de herramientas





**Argentina  
programa  
4.0**

# Gracias!

---