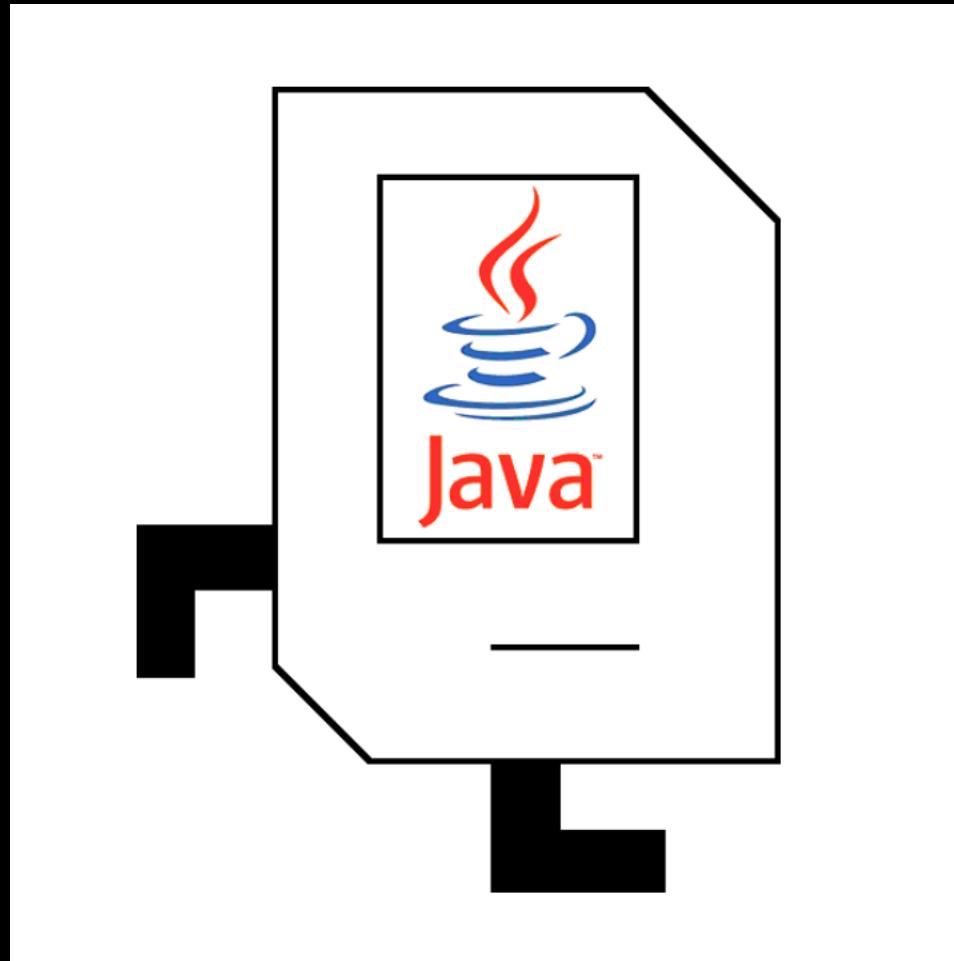


Universidad
de
los Andes

Variables

Repaso



Estructuras de control

```
private void solaDon() {  
    girarIzquierda();  
}  
  
for(int i = 0; i < N; i++) {  
    // repetir N veces  
}  
  
while(condición) {  
    // todo el código aquí se repite  
    // mientras la condición sea verdadera  
}  
  
if(condición) {  
    // ejecutar código aquí si condición es verdadera  
} else {  
    // ejecutar código aquí si condición es falsa  
}
```

Condicionales If-else

```
if(frenteDespejado( )) {      // verdadero  
    moverse();  
}  
ponerCono();  
  
                                moverse();  
                                // falso  
                                ponerCono();
```

```
if(frenteDespejado( )) {  
    moverse();  
} else {  
    girarIzquierda();  
}  
ponerCono();  
  
                                moverse();  
                                // verdadero  
                                ponerCono();  
                                girarIzquierda();  
                                // falso
```

Corchetes y sangría

```
public void run() {
    while (bolsaConHojas()) {
        encontrarArbol();
        agregarHojasAlArbol();
    }
    irHastaMuro();
}
private void agregarHojasAlArbol() {
    turnLeft();
    subirAlArbol();
        agregarHojas();
    descenderAlPiso();
    girarIzquierda();

private void encontrarArbol() {
    irHastaMuro();
}
```

Corchetes y sangría

```
public void run() {  
    while (bolsaConHojas()) {  
        encontrarArbol();  
        agregarHojasAlArbol();  
    }  
    irHastaMuro();  
  
}  
  
private void agregarHojasAlArbol() {  
    girarIzquierda();  
    subirAlArbol();  
    agregarHojas();  
    descenderAlPiso();  
    girarIzquierda();  
  
}  
  
private void encontrarArbol() {  
    irHastaMuro();  
}
```

Corchetes y sangría

```
public void run() {  
    while (bolsaConHojas()) {  
        encontrarArbol();  
        agregarHojasAlArbol();  
    }  
    irHastaMuro();  
}
```

```
private void agregarHojasAlArbol() {  
    girarIzquierda();  
    subirAlArbol();  
    agregarHojas();  
    descenderAlPiso();  
    girarIzquierda();
```

```
private void encontrarArbol() {  
    irHastaMuro();  
}
```

Corchetes y sangría

```
public void run() {  
    while (bolsaConHojas()) {  
        encontrarArbol();  
        agregarHojasAlArbol();  
    }  
    irHastaMuro();  
}
```

```
private void agregarHojasAlArbol() {  
    girarIzquierda();  
    subirAlArbol();  
    agregarHojas();  
    descenderAlPiso();  
    girarIzquierda();  
}
```

```
private void encontrarArbol() {  
    irHastaMuro();  
}
```

Corchetes y sangría

```
public void run() {  
    while (bolsaConHojas()) {  
        encontrarArbol();  
        agregarHojasAlArbol();  
    }  
    irHastaMuro();  
  
}  
  
private void agregarHojasAlArbol() {  
    girarIzquierda();  
    subirAlArbol();  
    agregarHojas();  
    descenderAlPiso();  
    girarIzquierda();  
}  
  
private void encontrarArbol() {  
    irHastaMuro();  
}
```

Corchetes y sangría

```
public void run() {  
    while (bolsaConHojas()) {  
        encontrarArbol();  
        agregarHojasAlArbol();  
    }  
    irHastaMuro();  
}
```

```
private void agregarHojasAlArbol() {  
    girarIzquierda();  
    subirAlArbol();  
    agregarHojas();  
    descenderAlPiso();  
    girarIzquierda();  
}
```

```
private void encontrarArbol() {  
    irHastaMuro();  
}
```

Descomposición – Recoger Periódico

Bien

```
public void run() {
    salirDeCasa();
    recogerPeriodico();
    regresar();
}

private void salirDeCasa()
{
    moverse();
    moverse();
    girarDerecha();
    moverse();
    girarIzquierda();
    moverse();
}

public void recogerPeriodico()
{
    ponerCono();
}

public void regresar() {
    mediaVuelta();
    moverse();
    moverse();
    moverse();
    girarDerecha();
    moverse();
}
```

Mal

```
public void run() {  
    moverse();  
    moverse();  
    girarIzquierda();  
    girarIzquierda();  
    girarIzquierda();  
    moverse();  
    girarIzquierda();  
    moverse();  
    recogerCono();  
    girarIzquierda();  
    girarIzquierda();  
    moverse();  
    moverse();  
    moverse();  
    girarIzquierda();  
    girarIzquierda();  
    girarIzquierda();  
    moverse();  
}
```

Use métodos para decomponer el código

Descomposición – Recoger Periódico

```
public void run() {
    salirDeCasa();
    recogerPeriodico();
    regresar();
}

private void salirDeCasa() {
    moverse();
    moverse();
    girarDerecha();
    moverse();
    girarIzquierda();
    moverse();
}

public void recogerPeriodico()
    ponerCono();
}

public void regresar() {
    mediaVuelta();
    moverse();
    moverse();
    moverse();
    girarDerecha();
    moverse();
}
```

```
public void run() {  
    moverse();  
    moverse();  
    girarIzquierda();  
    girarIzquierda();  
    girarIzquierda();  
    moverse();  
    girarIzquierda();  
    moverse();  
    recogerCono();  
    girarIzquierda();  
    girarIzquierda();  
    moverse();  
    moverse();  
    moverse();  
    girarIzquierda();  
    girarIzquierda();  
    girarIzquierda();  
    moverse();  
}
```

Descomposición – Recoger Periódico

```
public void run() {  
    salirDeCasa();  
    recogerPeriodico();  
    regresar();  
}  
  
private void salirDeCasa() {  
    moverse();  
    moverse();  
    girarDerecha();  
    moverse();  
    girarIzquierda();  
    moverse();  
}  
  
public void recogerPeriodico() {  
    ponerCono();  
}  
  
public void regresar() {  
    mediaVuelta();  
    moverse();  
    moverse();  
    moverse();  
    girarDerecha();  
    moverse();  
}
```

```
public void run() {  
    moverse();  
    moverse();  
    girarIzquierda();  
    girarIzquierda();  
    girarIzquierda();  
    moverse();  
    girarIzquierda();  
    moverse();  
    moverse();  
    recogerCono();  
    girarIzquierda();  
    girarIzquierda();  
    moverse();  
    moverse();  
    moverse();  
    girarIzquierda();  
    girarIzquierda();  
    girarIzquierda();  
    moverse();  
}
```

Bien o Mal?

Bien

```
// mover karel un paso
private void paso() {
    moverse();
}

// rotar karel
private void rotar() {
    girarIzquierda();
}
```

Mal

```
private void m1() {
    moverse();
}

private void m2() {
    girarIzquierda();
}
```

Use nombres de
método descriptivos

Empareja los corchetes !!

```
1 import stanford.karel.*  
2 */ **/  
3 * Programa : Poner100  
4 * -----  
5 * Este programa hace que karel coloque un montón de  
6 * 100 conos. Buenos tiempos  
7 */  
8 public class Poner100 extends EsKarel {  
9  
10 public void run() {  
11     moverse();  
12     // Este for-loop repetirá el código dentro 100 veces.  
13     for(int i = 0; i < 100; i++) {  
14         ponerCono();  
15     }  
16     moverse();  
17 }  
18 }
```

Empareja los corchetes !!

```
1 import stanford.karel.*  
2 /**  
3  * Programa : Poner100  
4  * -----  
5  * Este programa hace que karel coloque un montón de  
6  * 100 conos. Buenos tiempos  
7 */  
8 public class Poner100 extends EsKarel {  
9  
10    public void run() {  
11        moverse();  
12        // Este for-loop repetirá el código dentro 100 veces.  
13        for(int i = 0; i < 100; i++) {  
14            ponerCono();  
15        }  
16        moverse();  
17    }  
18 }
```

Empareja los corchetes !!

```
1 import stanford.karel.*  
2 /**  
3  * Programa : Poner100  
4  * -----  
5  * Este programa hace que karel coloque un montón de  
6  * 100 conos. Buenos tiempos  
7 */  
8 public class Poner100 extends EsKarel {  
9  
10    public void run() {  
11        moverse();  
12        // Este for-loop repetirá el código dentro 100 veces.  
13        for(int i = 0; i < 100; i++) {  
14            ponerCono();  
15        }  
16        moverse();  
17    }  
18 }
```

Empareja los corchetes !!

```
1 import stanford.karel.*  
2 /**  
3  * Programa : Poner100  
4  * -----  
5  * Este programa hace que karel coloque un montón de  
6  * 100 conos. Buenos tiempos  
7 */  
8 public class Poner100 extends EsKarel {  
9  
10    public void run() {  
11        moverse();  
12        // Este for-loop repetirá el código dentro 100 veces.  
13        for(int i = 0; i < 100; i++) {  
14            ponerCono();  
15        }  
16        moverse();  
17    }  
18}
```



Empareja los corchetes !!

```
1 import stanford.karel.*  
2 /**  
3  * Programa : Poner100  
4  * -----  
5  * Este programa hace que karel coloque un montón de  
6  * 100 conos. Buenos tiempos  
7 */  
8 public class Poner100 extends EsKarel {  
9  
10    public void run() {  
11        moverse();  
12        // Este for-loop repetirá el código dentro 100 veces.  
13        for(int i = 0; i < 100; i++) {  
14            ponerCono();  
15        }  
16        moverse();  
17    }  
18  
19}  
20
```

Empareja los corchetes !!

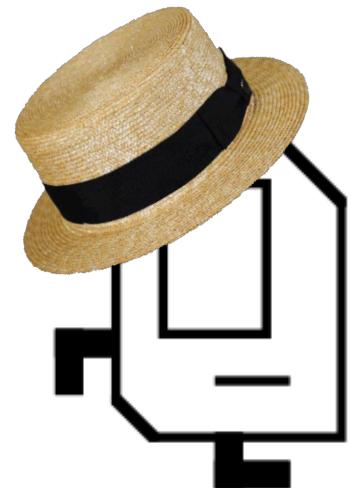
```
1 import stanford.karel.*  
2 /**  
3  * Programa : Poner100  
4  * -----  
5  * Este programa hace que karel coloque un montón de  
6  * 100 conos. Buenos tiempos  
7 */  
8 public class Poner100 extends EsKarel {  
9  
10    public void run() {  
11        moverse();  
12        // Este for-loop repetirá el código dentro 100 veces.  
13        for(int i = 0; i < 100; i++) {  
14            ponerCono();  
15        }  
16        moverse();  
17    }  
18}  
19}  
20
```

The diagram illustrates the matching of curly braces in the code. It uses colored circles to highlight specific braces: a large green circle surrounds the opening brace of the main class definition (line 8), a medium green circle surrounds the opening brace of the run() method (line 10), and a large blue circle surrounds the closing brace of the for-loop (line 15). A smaller blue circle highlights the opening brace of the inner for-loop (line 13). A yellow horizontal bar highlights the closing brace of the inner for-loop (line 17) and the closing brace of the outer for-loop (line 15). A small yellow circle highlights the closing brace of the run() method (line 16).

Tu Código lo van a leer otros

Uso descomposición usando
funciones descriptivas

Uso sangrías y empareja los
corchetes



Programar require práctica

Nos vemos!



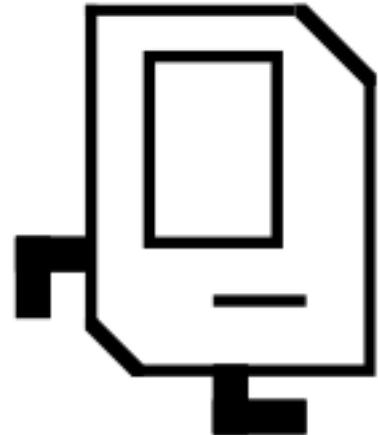
Te extrañaré.

Disfruta Java!

¿Me llamas
luego?

Nos vemos!

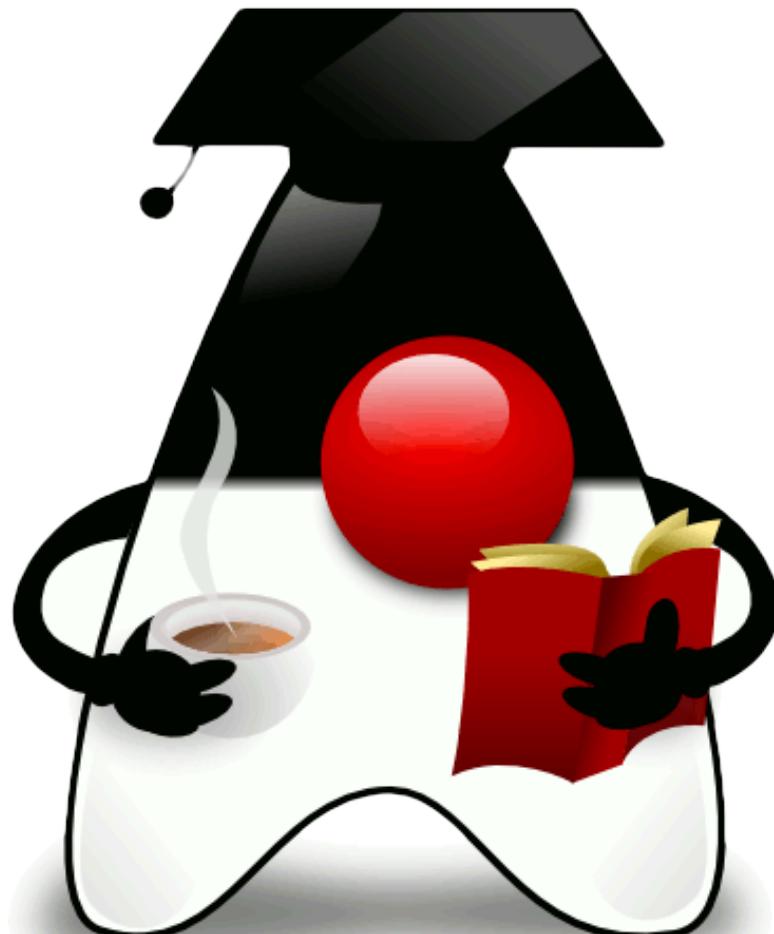
Karel es un buen inicio !! (Calentamiento)



No te preocupes si no terminaste.

Siquieres terminar tu "mountain Karel", nos vemos durante el almuerzo !!

Java



Objetivos de hoy

1. ¿Cómo escribir un programa de consola?
2. ¿Qué son variables y cómo usarlas?
3. ¿Cómo leer entradas del usuario en un programa de consola?



Programa de Consola



Lee entradas de usuario

Imprime
mensajes/datos

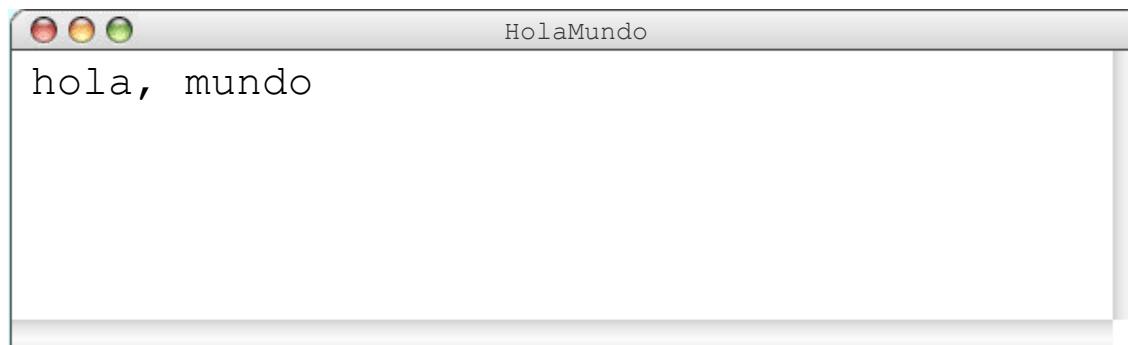
```
Last login: Mon Jul 3 17:20:41 on ttys007
/Users/Lisa $
```

Primer programa de consola: Hola Mundo

```
import acm.program.*;

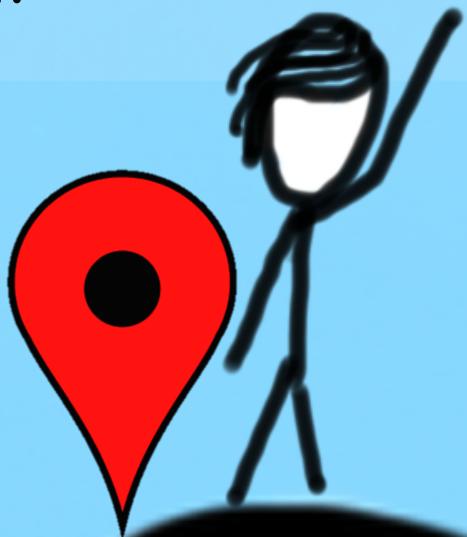
public class HolaMundo extends EsConsole {

    public void run() {
        imprimir("hola, mundo");
    }
}
```



Objetivos de hoy

- ✓ 1. ¿Cómo escribir un programa de consola?
- 8 2. ¿Qué son variables y cómo usarlas?
- 3. ¿Cómo leer entradas del usuario en un programa de consola?



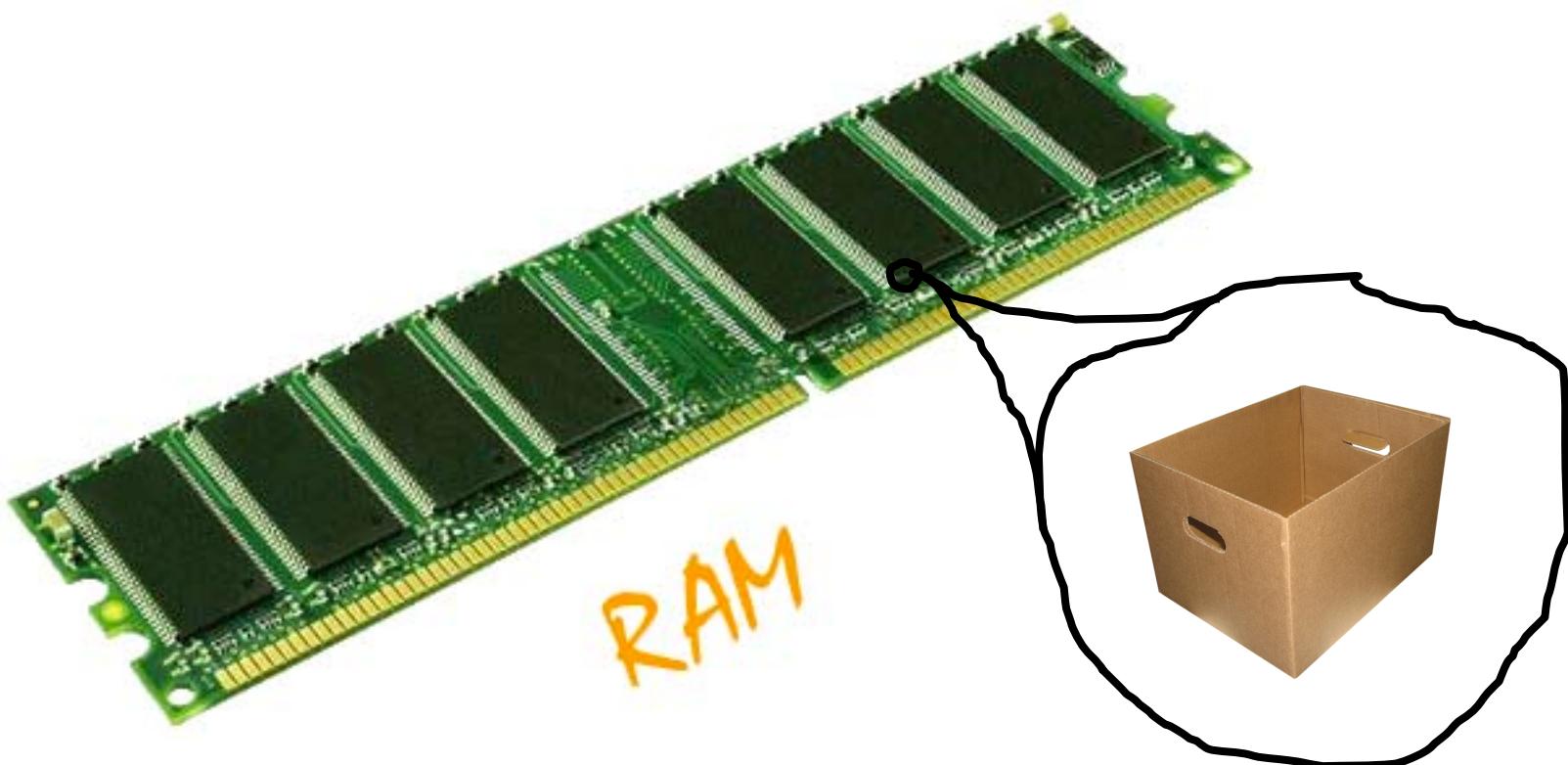
¿Qué es una variable?

[suspenso]

Las variables son como cajas



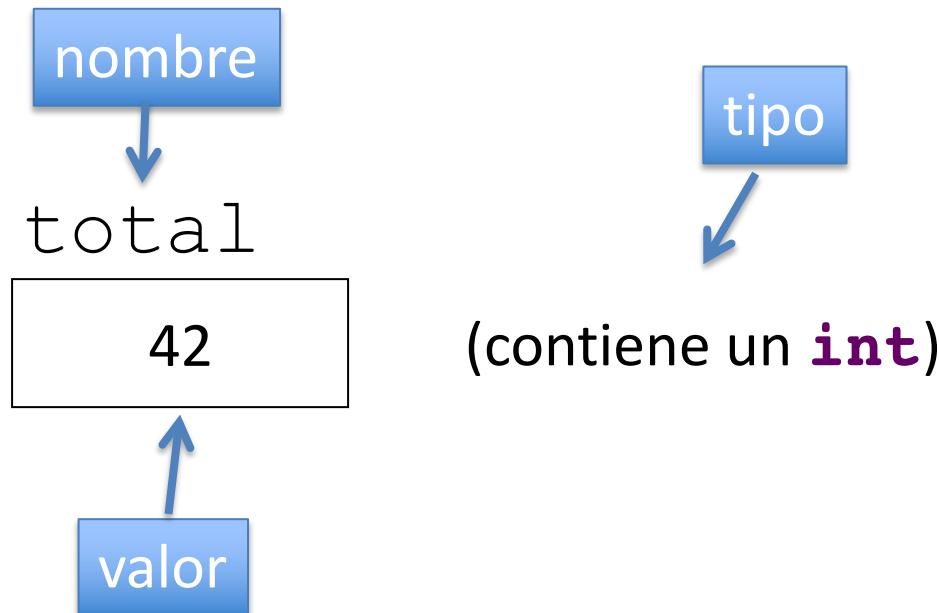
Cajas... muy pequeñas



Mi computador tiene espacio para 64 mil cajas

Las variables son como cajas

```
int total = 42;
```



Tipos

```
// valores enteros  
int num = 5;
```

```
// valores decimales  
double fraccion = 0.2;
```

```
// letras  
char letra = 'c';
```

```
// verdadero (true) o falso (false)  
boolean esAmor = true;
```

```
// cadena de texto  
String mensaje = "Hola amigo";
```

Double: ¿Cuál es mi peso?



* La respuesta puede ser un valor con decimales

Int: Cuántos hijos hay en una familia?



* Sería raro responder 1.7

Operadores binarios

- | | | |
|---------|---|----------------|
| + Suma | * | Multiplicación |
| - Resta | / | División |
| | % | Módulo/Residuo |

Operadores binarios

```
double ancho = 2.5; // metros  
double alto = 3.0;  
double area = ancho * alto;
```

nombre

ancho

alto

área

valor

2.5

3.0

7.5

tipo

double

double

double

Juguemos !!

```
int total = 5;
```

```
double promedio = 3.5;
```

Juguemos !!

```
int total = 5;
```

```
double promedio = 3.5;
```

```
total = total + 2;
```

Juguemos !!

```
int total = 5;
```

```
double promedio = 3.5;
```

```
total = total + 2;
```

```
promedio = 3.5 + 1.5;
```

Juguemos !!

```
int total = 5;
```

```
double promedio = 3.5;
```

```
total = total + 2;
```

```
promedio = 3.5 + 1.5;
```

```
total = 10;
```

Juguemos !!

```
int total = 5;
```

```
double promedio = 3.5;
```

```
total = total + 2;
```

```
promedio = 3.5 + 1.5;
```

```
total = 10;
```

```
promedio = promedio -1;
```

Juguemos !!

```
int total = 5;
```

```
double promedio = 3.5;
```

```
total = total + 2;
```

```
promedio = 3.5 + 1.5;
```

```
total = 10;
```

```
promedio = promedio -1;
```

Juguemos !!

```
int total = 5;
```

```
double promedio = 3.5;
```

```
total = total + 2;
```

```
promedio = 3.5 + 1.5;
```

```
total = 10;
```

```
promedio = promedio -1;
```

```
total = total - 5;
```

Juguemos !!

```
int total = 5;
```

```
double promedio = 3.5;
```

```
total = total + 2;
```

```
promedio = 3.5 + 1.5;
```

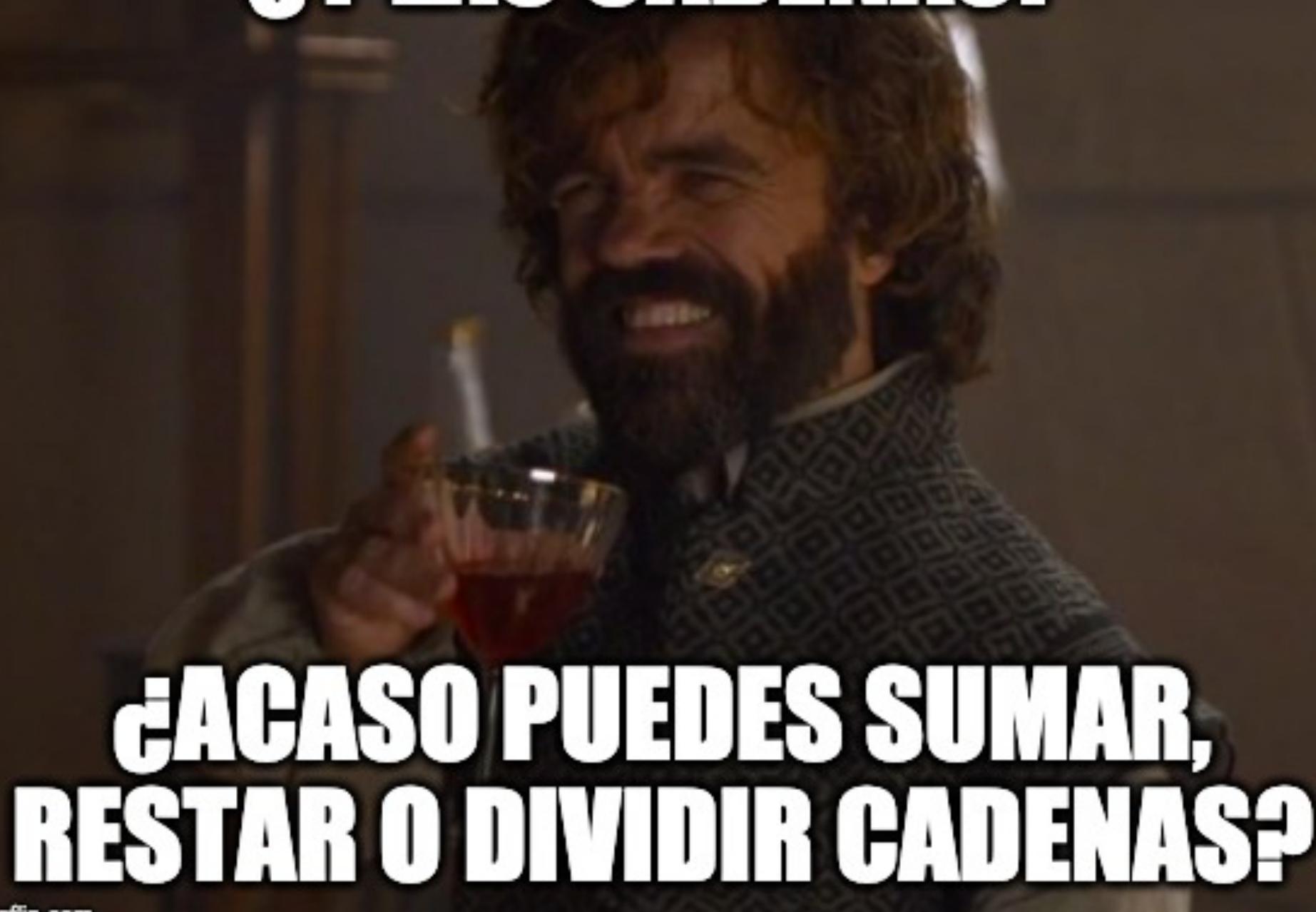
```
total = 10;
```

```
promedio = promedio -1;
```

```
total = total - 5;
```

```
promedio = (promedio+1)/2;
```

¿Y LAS CADENAS?



¿ACASO PUEDES SUMAR, RESTAR O DIVIDIR CADENAS?

Operadores binarios

```
String mensaje = "Hola";
String nombre = "Mario";
imprimir(mensaje+nombre);
```

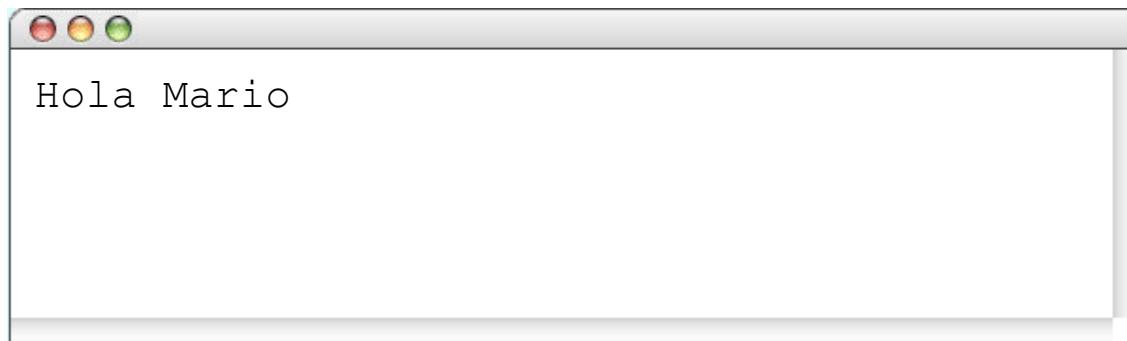
Operadores binarios

```
String mensaje = "Hola";
String nombre = "Mario";
imprimir(mensaje+nombre);
```



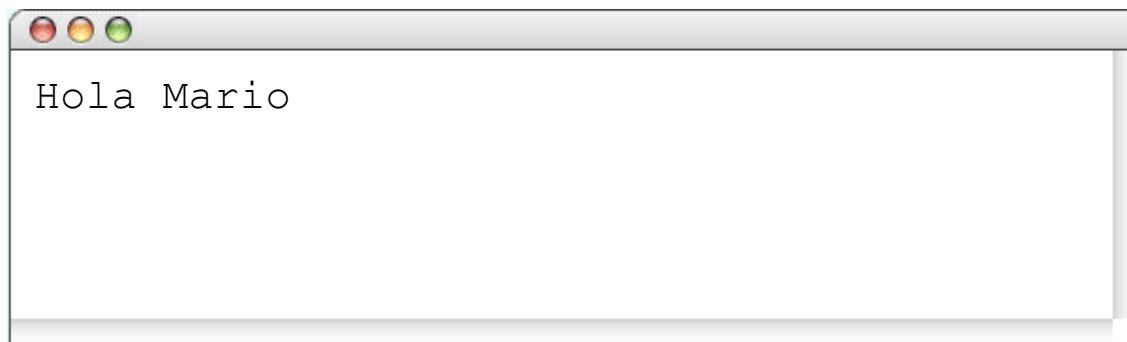
Operadores binarios

```
String mensaje = "Hola";
String nombre = "Mario";
imprimir(mensaje+" "+nombre);
```



Operadores binarios

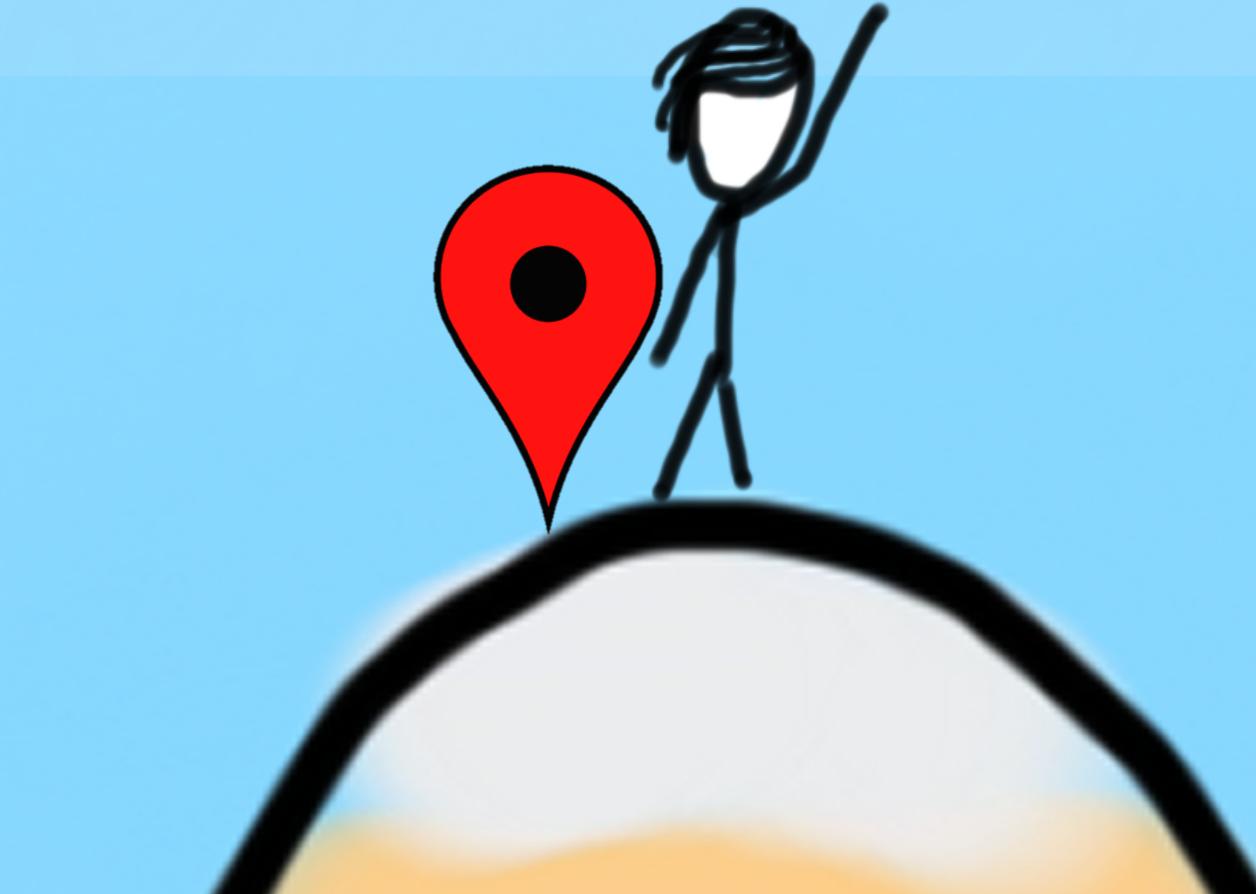
```
String mensaje = "Hola";
String nombre = "Mario";
String saludo = mensaje+ " "+nombre;
imprimir(saludo);
```



Objetivos de hoy



1. ¿Cómo escribir un programa de consola?
2. ¿Qué son variables y cómo usarlas?
3. ¿Cómo leer entradas del usuario en un programa de consola?



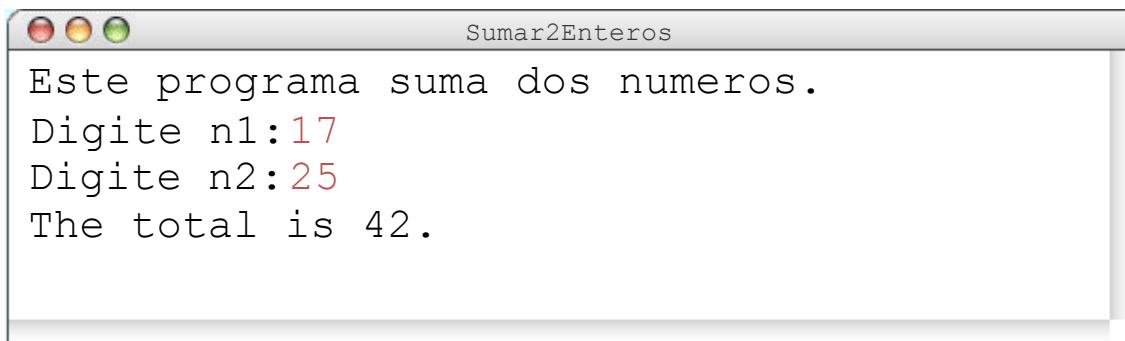
Entrada de usuario

```
int a = leerInt("Dame un entero!");  
  
double b = leerDouble("y un decimal");
```

Sumar2Enteros

```
public class Sumar2Enteros extends EsConsole {  
    public void run() {  
        imprimir("Este programa suma dos numeros.");  
        int n1 = leerInt("Digite n1: ");  
        int n2 = leerInt("Digite n2: ");  
        int total = n1 + n2;  
        imprimir("El total es " + total + ".");  
    }  
}
```

n1	n2	total
17	25	42

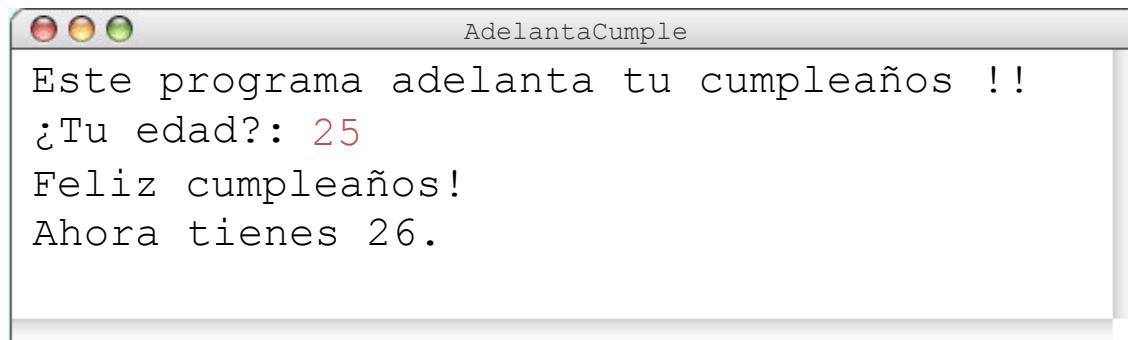


Cumpleaños

```
public class AdelantaCumple extends EsConsole {  
    public void run() {  
        imprimir("Este programa adelanta tu cumpleaños.");  
        int edad = leerInt("¿Tu edad?: ");  
        edad = edad + 1;  
        imprimir("Feliz cumpleaños!");  
        imprimir("Ahora tienes" + edad + ".");  
    }  
}
```

edad

25

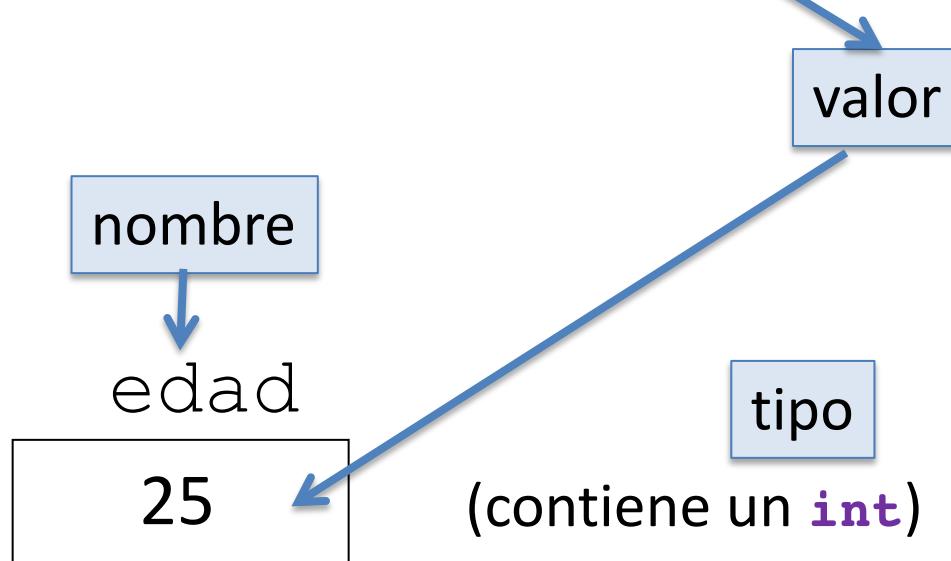


Cumpleaños

```
int edad = leerInt("¿Tu edad?: ");  
edad = edad + 1;
```

Birthday

```
int edad = leerInt("¿Tu edad? : ");
```



Cumpleaños

```
int edad = leerInt("¿Tu edad? : ");
```

```
edad = edad + 1;
```



valor

edad

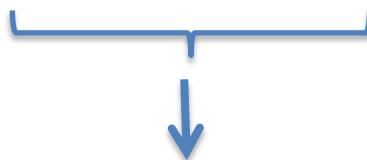
25

(contiene un int)

Cumpleaños

```
int edad = leerInt("¿Tu edad? : ");
```

```
edad = edad + 1;
```



$$25 + 1 = 26$$

edad

25

(contiene un int)

Cumpleaños

```
int edad = leerInt("¿Tu edad? : ");
```

```
edad = edad + 1;
```



26

dad



(contiene un **int**)

Cumpleaños

```
int edad = leerInt("¿Tu edad?: ");
```

```
edad = edad + 1;
```



26

edad



(contiene un **int**)

¿Cuál es el resultado?

```
imprimir(1 / 2);
```

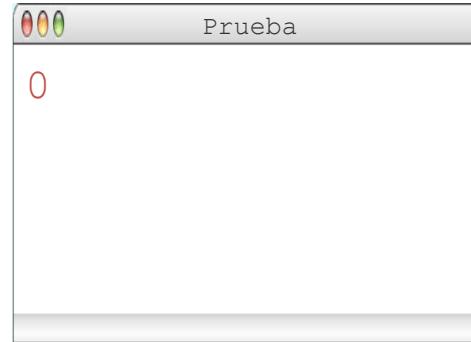
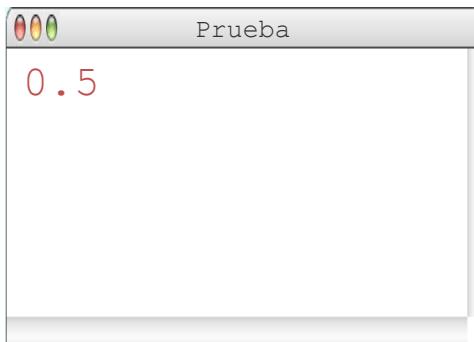
¿Cuál es el resultado?

```
imprimir(1 / 2);
```



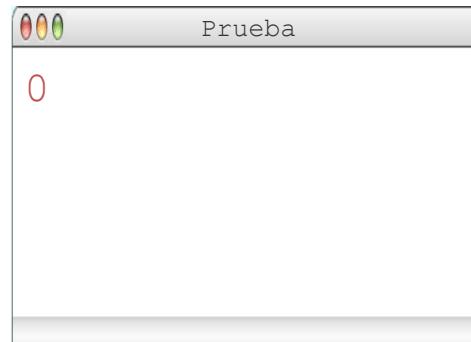
¿Cuál es el resultado?

```
imprimir(1 / 2);
```



AHHHHHHHH!!!!!!

```
imprimir(1 / 2);
```



Resultado

int + int es un int

double + double es un double

int + double es un double

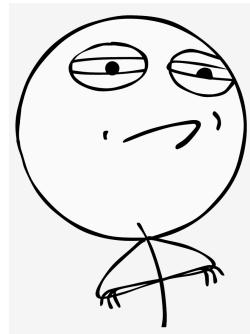
* The regla general es: las operaciones siempre retornan el tipo más expresivo

Celsius (C) a Farenheit (F)

$$F = \frac{9}{5}C + 32$$

BUSCA 2 O 3 COMPAÑEROS Y ESCRIBE EL CÓDIGO

5
MINUTOS



Celsius a Farenheit

Agrega una foto de la solución generada por tu equipo. No olvides escribir el nombre de los autores !!

<https://padlet.com/mlinarev/jl60d3llr2jy>



División entera

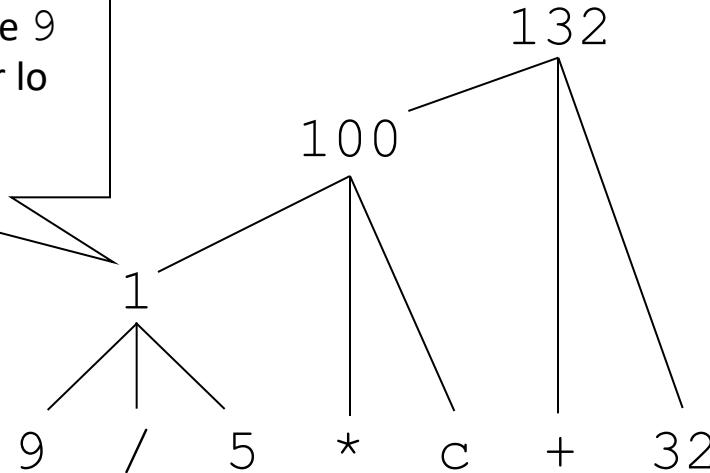
Convertir 100° Celsius a grados Fahrenheit:

```
double c = 100;  
double f = ((9 / 5) * c) + 32;
```



El cálculo consiste en evaluar la siguiente expresión

El problema radica en que 9
y 5 son de tipo int, por lo
tanto el resultado es
también un int.

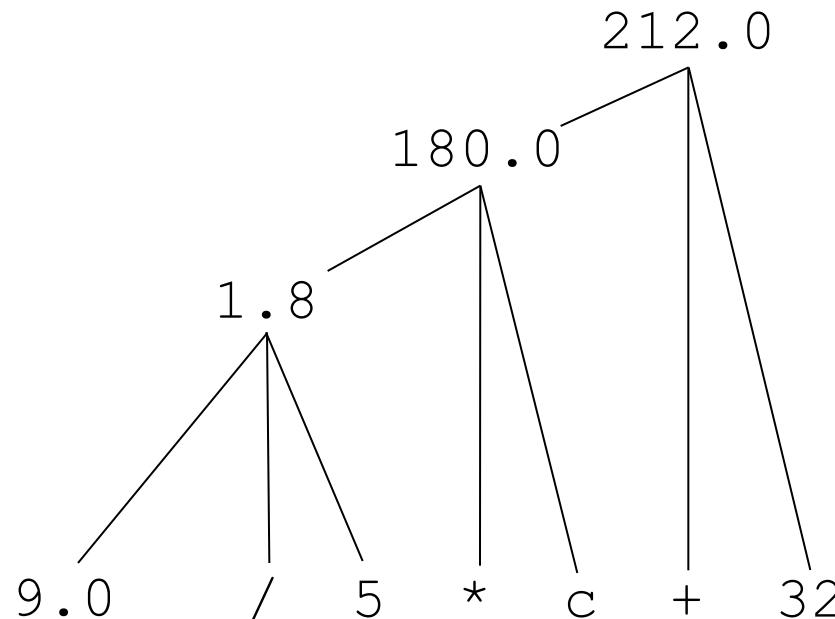


División entera

El problema se puede solucionar convirtiendo la división entera en división de tipo `double`, agregando decimales

```
double c = 100;  
double f = ((9.0 / 5) * c) + 32;
```

El cálculo se verá ahora así:



Operadores de comparación

Operadores de comparación

< Menor que

== Igual

> Mayor que

>= Mayor o igual

<= Menor o igual

!= Diferente

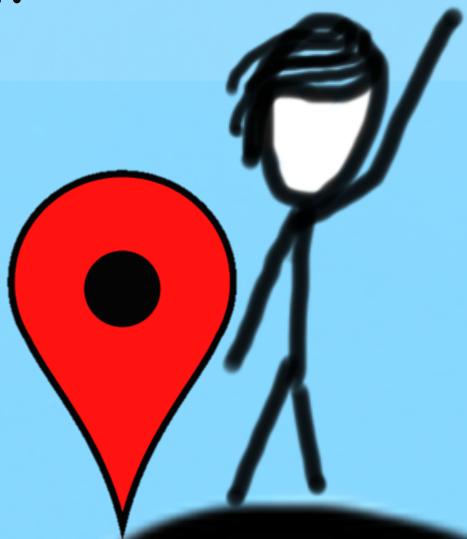
Condicionales

```
public void run() {  
    imprimir("Este programa identifica si eres cool o no.");  
    int numeroCanciones = leerInt("Cuántas canciones conoces de J. Balvin?:");  
    if(numeroCanciones == 1){  
        imprimir("Solo conoces mi gente !!");  
    }  
  
    if(numeroCanciones > 3){  
        imprimir("OK, eres cool !");  
    }  
    else {  
        imprimir("Lo siento, no eres cool.");  
    }  
}
```

Objetivos de hoy



1. ¿Cómo escribir un programa de consola?
2. ¿Qué son variables y cómo usarlas?
3. ¿Cómo leer entradas del usuario en un programa de consola?



A trabajar !!

Castillos de arena

