



TECNICATURA UNIVERSITARIA EN DISEÑO  
INTEGRAL DE VIDEOJUEGOS

FACULTAD DE INGENIERÍA  
Universidad Nacional de Jujuy



# FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS

Trabajo Práctico/Actividad

N°1

Santiago Larraburu – TUV000618 /

Grupo:

Integrantes

AyN /LU

*Profesores:*

*Mg. Ing. Ariel Alejandro Vega*

*Ing. Carolina Cecilia Apaza*

*Año*



FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS  
TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS

FACULTAD DE INGENIERÍA

Universidad Nacional de Jujuy

PENSAMIENTO COMPUTACIONAL y PROGRAMACIÓN:

Problema y Solución – PC y P – Algoritmos – Principio de la P



Punto 1: Evaluar (obtener resultado) la siguiente expresión para A = 2 y B = 5

$$3 * A - 4 * B / A ^ 2$$

Desarrollo del punto

Resolución en Word:

$$(3 * A) - (4 * B / (A^2))$$

$$6 - (4 * B / 4)$$

$$6 - 5 = 1$$

Captura de Processing

```

1  int A=2, B=5;
2
3  float resultado = 3 * A - 4 * B / pow(2,2);
4
5  println(resultado);
6

```

1.0

**Ejercicio 2:** Evaluar la siguiente expresión  $4 / 2 * 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 * 2$

Resolución en Word:

$$(((4/2) * 3) / 6) + (((((6/2)/1)/(5^2))/4)*2)$$

$$((2*3)/6) + (((3/1)/(25))/4)*2)$$

$$(6/6) + ((3/25)/4)*2)$$

$$1 + 0.06 = 1.06$$

Captura Processing:

```

1  float resultado = (((4/2) * 3) / 6) + (((((6/2)/1)/pow(5,2))/4)*2);
2  println(resultado);
3

```

1.06

**Ejercicio 4:** : Evaluar las siguientes expresiones aritméticas, para lo cual indicar en el caso de las variables, el valor indicado. Luego escribirlas como expresiones algebraicas.

Variables: A=3, B=5, C=1, D=2

a)  $b^2 - 4 * a * c$

$(5^2) - (4 * 3 * 1)$

$25 - 12$

**13**

Expresión algebraica:

$(5^2 - 4(3)(1))$

```
int a=3, b=5, c=1, d=2;
```

```
float resultado = pow(b,2)-(4*a*c);
```

```
println(resultado);
```

13.0

b)  $3 * X^4 - 5 * X^3 + X^2 - 17$

c)  $(b + d) / (c + 4)$

$(5+2) / (1+4)$

$7/5$

**1,4**

Expresión algebraica:

$(5+2) : (1+4)$

d)  $(x^2 + y^2)^{1/2}$

**Ejercicio 5:** Si el valor de A es 4, el valor de B es 5 y el valor de C es 1, evaluar las siguientes expresiones:

a)  $B * A - B^2 / 4 * C$

$(5*4) - (5^2/4*1)$

$20 - 25/4$

$20 - 6,25$

**13,75**



```
int a=4, b=5, c=1;  
  
float resultado = b*a-pow(b,2)/4*c;  
  
println(resultado);
```

13.75

b)  $(A * B) / 3^2$

$(4*5) / 9$

$20/9$

**2,22**

```
float resultado2 = (a*b)/pow(3,2);
```

```
println(resultado2);
```

2.2222223

c)  $((B + C) / 2 * A + 10) * 3 * B - 6$

$((5+1) / 2 * 4+10) * 3 * 5 - 6$

$((6/2*4+10)*3*5)-6$

$(22*3*5)-6$

**324**

```
float resultado3 = (((b+c)/2*a+10)*3*b)-6;
```

```
println(resultado3);
```

324.0



**Ejercicio 6:** Para  $x=3$ ,  $y=4$ ;  $z=1$ , evaluar el resultado de:

$R1 = y+z$

$R2 = x \geq R1$

$R1 = 4+1 = 5$

$R2 = 3 \geq R1$

$R2 = \text{false}$

```
int x=3, y=4, z=1;

int r1 = y+z;

boolean r2 = x >= r1;

println(r1,r2);
```

5 false

**Ejercicio 7:** Para  $\text{contador1}=3$ ,  $\text{contador2}=4$ , evaluar el resultado de

$R1 = ++\text{contador1}$

$R2 = \text{contador1} < \text{contador2}$

$R1 = 4$

$R2 = \text{False}$

```
int contador1=3, contador2=4;

int R1= ++contador1;

boolean R2= contador1<contador2;

println(R1,R2);
```

4 false



**Ejercicio 8:** Para  $a=31$ ,  $b=-1$ ;  $x=3$ ,  $y=2$ , evaluar el resultado de

$$a+b-1 < x*y$$

$$31+(-1)-1 < 3*2$$

$$29 < 6$$

**False**

```
int a=31,b=-1,x=3,y=2;  
  
boolean resultado= a+b-1 < x*y;  
  
println(resultado);
```

false

**Ejercicio 9:** Para  $x=6$ ,  $y=8$ , evaluar el resultado de

$$!(x<5) \text{CC} !(y \geq 7)$$

**False**

```
int x=6,y=8;  
  
boolean resultado= !(x<5) && !(y>=7);  
  
println(resultado);
```

false

**Ejercicio 10:** Para  $i=22$ ,  $j=3$ , evaluar el resultado de

$$!((i>4) \text{ || } !(j \leq 6))$$

**False**



```
int i=22, j=3;  
  
boolean resultado = !((i>4)|| !(j<=6));  
  
println(resultado);
```

false

Ejercicio 11: Para a=34, b=12, c=8, evaluar el resultado de

!(a+b==c) || (c!=0)&&(b-c>=19)

True

```
int a=34, b=12, c=8;  
  
boolean resultado= !(a+b==c) || (c!=0)&&(b-c>=19);  
  
println(resultado);
```

true

### Sección Análisis – Diseño y Codificación de algoritmos – Aplicación de estructuras de control

Para cada ejercicio, en el archivo Word agregar las secciones de análisis y diseño, mientras que, para la codificación, crear el archivo de Processing.

**Ejercicio 13:** Será común resolver problemas utilizando variables. Calcule el perímetro y área de un rectángulo dada su base y su altura.

**Especificación del problema:** Se debe determinar simultáneamente el perímetro y el área de un rectángulo según su base y su altura.

Análisis:

Datos de entrada: baseDelRectangulo:int;

alturaDelRectangulo:int;

Datos de salida: perimetroRectangulo:int;

areaRectangulo:int;



Algoritmo:

Entidad que resuelve el problema: Usuario

Variables:

baseDelRectangulo:int; //almacena el valor de la base del rectangulo

alturaDelRectangulo:int; // almacena el valor de la altura del rectángulo

Nombre algoritmo: calcular\_perimetro\_y\_area\_rectangulo

Proceso del Algoritmo:

1. leer baseDelRectangulo;
2. leer alturaDelRectangulo;
3.  $\text{perimetroRectangulo} \leftarrow 2 * (\text{baseDelRectangulo} + \text{alturaDelRectangulo});$
4.  $\text{areaRectangulo} \leftarrow \text{baseDelRectangulo} * \text{alturaDelRectangulo};$
5. mostrar perimetroRectangulo y areaRectangulo;

```
Ejercicio13
int baseDelRectangulo = 10;
int alturaDelRectangulo = 30;

int perimetroRectangulo = 2 * (baseDelRectangulo + alturaDelRectangulo);
int areaRectangulo = baseDelRectangulo * alturaDelRectangulo;

println("El perimetro es: " + perimetroRectangulo);
println("El area es: " + areaRectangulo);
```

```
El perimetro es: 80
El area es: 300
```

**Ejercicio 14:** Ejercicio 14: Una ayuda importante al momento de resolver problemas con algoritmos es asumir que su gran amigo son las matemáticas. Obtenga la hipotenusa de un triángulo rectángulo conociendo sus catetos

Especificación del problema: Se debe conseguir la hipotenusa de un triángulo rectángulo haciendo uso de: sus catetos y el teorema de Pitágoras.

Análisis del problema:

Datos de entrada: cateto1:int; cateto2:int;

Datos de salida: hipotenusa:float;

Algoritmo: Entidad que resuelve el problema: Usuario

Variables: cateto1



cateto2

hipotenusa

Nombre del Algoritmo: calcular\_Hipotenusa

Proceso Algoritmo: 1. leer cateto1

2. leer cateto2

3.  $hipotenusa \leftarrow \sqrt{(cateto1^2) + (cateto2^2)}$

4. mostrar hipotenusa.

sketch 240415o

```
int cateto1 = 3;
int cateto2 = 4;

float hipotenusa = sqrt(pow(cateto1, 2) + pow(cateto2, 2));

println("La hipotenusa es: " + hipotenusa);
```

La hipotenusa es: 5.0

**Ejercicio 16:** : Necesitamos convertir una temperatura Fahrenheit en grados Celsius.

Especificación del problema: Se debe convertir una temperatura dada en Fahrenheit a la medida Celsius.

Análisis del problema:

Datos de entrada: temperaturaFahrenheit:int;

Datos de salida: temperaturaCelsius:float;

Algoritmo: Entidad que resuelve el problema: usuario

Variables: temperaturaFahrenheit: entero

temperaturaCelsius: real

Nombre del Algoritmo: convertir\_a\_Celsius

Proceso algoritmo:

- 1) Leer temperaturaFahrenheit
- 2)  $\text{temperaturaCelsius} = (\text{temperaturaFahrenheit} - 32) / 1.8$
- 3) Mostrar temperaturaCelsius.

Ejercicio 19: Declare las variables necesarias para dibujar una línea que se dibuja desde las coordenadas iniciales del lienzo y se extiende por todo el ancho. Sobre el punto medio de la línea y a una distancia de 40px (en sentido vertical desde la línea) dibuje una elipse que tenga como ancho 80px y de alto 80px. Dentro de la función draw(), actualice las variables necesarias para que la línea desde su inicio se mueva en dirección hacia abajo arrastrando la elipse. Mantenga en cero el valor para background(). Cuando la línea supere la posición de la altura del lienzo, debe invertir su sentido, es decir dirigirse hacia arriba arrastrando la elipse. Cuando la línea alcance nuevamente el valor 0 para su posición en y, el desplazamiento debe ser hacia abajo y así sucesivamente. El lienzo debería verse como en las siguientes figuras.

Especificación del problema: Se debe crear una línea que tenga una elipse de 80px por 80px encima. Luego la línea debe moverse hacia el fondo con la elipse mirando hacia el fondo. Cuando la línea llegue al limite del lienzo, debe subir y la elipse colocarse por encima de la línea. Este algoritmo debe repetirse.

Analisis: Datos de entrada: float AnchoElipse , float AltoElipse, float yPos, float velocidad ;

Algoritmo: Entidad que resuelve el problema: Usuario

Variables: AnchoElipse, AltoElipse, yPos, velocidad, moviendoAbajo.

Nombre del algoritmo : mover\_Linea\_Y\_elipse\_de\_forma\_continua

Proceso del Algoritmo:

- 1) Establecer background 0
- 2) Dibujar una línea equivalente al ancho del lienzo.
- 3) Establecer elipse en el medio de la línea.
- 4) Leer AnchoElipse y AltoElipse, yPos y velocidad.
- 5) Si moviendoAbajo es true mueve la línea y la elipse hacia abajo hasta que de false.
- 6) Mostrar dibujo en el lienzo.



FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS  
TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS  
FACULTAD DE INGENIERÍA  
Universidad Nacional de Jujuy  
Trabajo Practico N° / Actividad

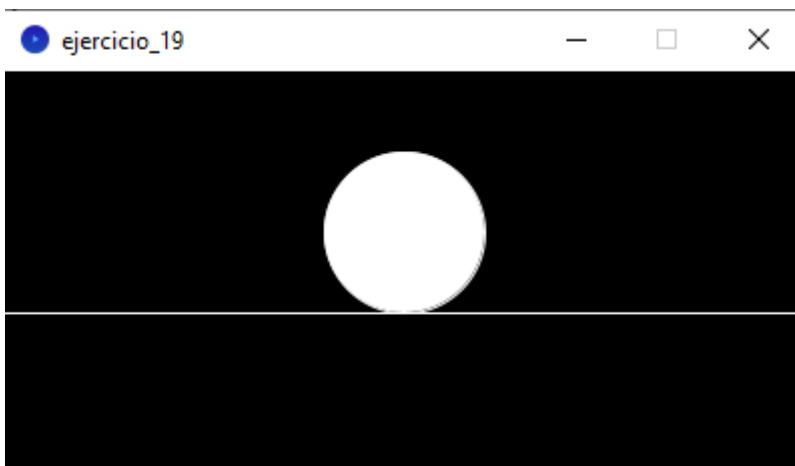




```
ejercicio 19
1 float yPos;
2 float velocidad = 2;
3 boolean moviendoAbajo = true;

4
5 float AnchoElipse = 80;
6 float AltoElipse = 80;
7
8 void setup() {
9     size(400, 200);
10
11     yPos = 0;
12 }
13
14 void draw() {
15     background(0);
16
17     stroke(255);
18
19     if (moviendoAbajo) {
20         line(0, yPos, width, yPos); |
21         fill(255);
22         ellipse(width / 2, yPos + 40, AnchoElipse, AltoElipse);
23     } else {
24         fill(255);
25     }
26 }
```

```

27     if (moviendoAbajo) {
28         line(0, yPos, width, yPos); |
29         fill(255);
30         ellipse(width / 2, yPos + 40, AnchoElipse, AltoElipse);
31     } else {
32         fill(255);
33         ellipse(width / 2, yPos - 40, AnchoElipse, AltoElipse);
34         line(0, yPos, width, yPos);
35     }
36
37     if (moviendoAbajo) {
38         yPos += velocidad;
39         if (yPos >= height) {
40             moviendoAbajo = false;
41         }
42     } else {
43         yPos -= velocidad;
44         if (yPos <= 0) {
45             moviendoAbajo = true;
46         }
47     }
48 }
```



	<p>FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS  TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS  <b>FACULTAD DE INGENIERÍA</b>  Universidad Nacional de Jujuy  Trabajo Practico N° / Actividad</p>	
---	--	---

Conclusión

Párrafos de las conclusiones

Fuentes bibliográficas

Se deben enunciar las fuentes (apuntes de la materia, páginas web, videos de youtube, libro (nombre, autores, año), etc)