

**PUERTO** *e-learning*

CURSO VIRTUAL

# Introducción al desarrollo de aplicaciones web con JavaScript.

JS

Universidad Autónoma de Entre  
Ríos



# Introducción al desarrollo de aplicaciones web con JavaScript (ECMAScript 6)

Módulo 2: Las bases de JavaScript.

---

**DOCENTE**

**López Gastón**

Técnico Universitario en Tecnologías de la Información



# Algunos métodos de los arrays

```
const primos = [ 3, 2, 7, 5]; // Array ejemplo
```

Método	Acción	Ejemplo	Resultado
push	Insertar un elemento al final.	primos.push(11);	[ 3, 2, 7, 5, 11]
splice	Insertar un elemento en cualquier parte.	primos.splice(2, 0, 23);	[ 3, 2, 23, 7, 5]
concat	Concatena dos arrays.	primos.concat([1,2,3]);	[3, 2, 7, 5, 1, 2, 3];
pop	Elimina el último elemento, y lo retorna.	primos.pop();	[3, 2, 7]
shift	Elimina el primer elemento, y lo retorna.	primos.shift();	[2, 7, 5]
splice	Elimina cualquier cantidad de elementos en cualquier posición.	primos.splice(1,2);	[3, 5]
find	Devuelve el primer valor que cumpla la condición de la función.	primos.find((e) => e > 5);	7
indexOf	Devuelve el index del primer elemento encontrado.	primos.indexOf(7);	2
lastIndexOf	Devuelve el index del último elemento encontrado.	primos.indexOf(7);	2
sort	Ordena un array de menor a mayor, o alfabéticamente.	primos.sort();	[2, 3, 5, 7]
reverse	Revierde el orden de un array	primos.reverse();	[5, 7, 2, 3]

# Arrays multidimensionales

## 1-D array

`arr = [1, 2, 3, 4, 5]`

(0)

1	2	3	4	5
---	---	---	---	---

`arr[0], arr[1], arr[2]`

## 2-D array

`arr = [  
 [1, 2, 3],  
 [4, 5, 6],  
 [7, 8, 9]  
]`

(0, 0)

1	2	3
4	5	6
7	8	9

`arr[0][0] => 1  
arr[1][2] => 6  
arr[2][0] => 7`

## 3-D array

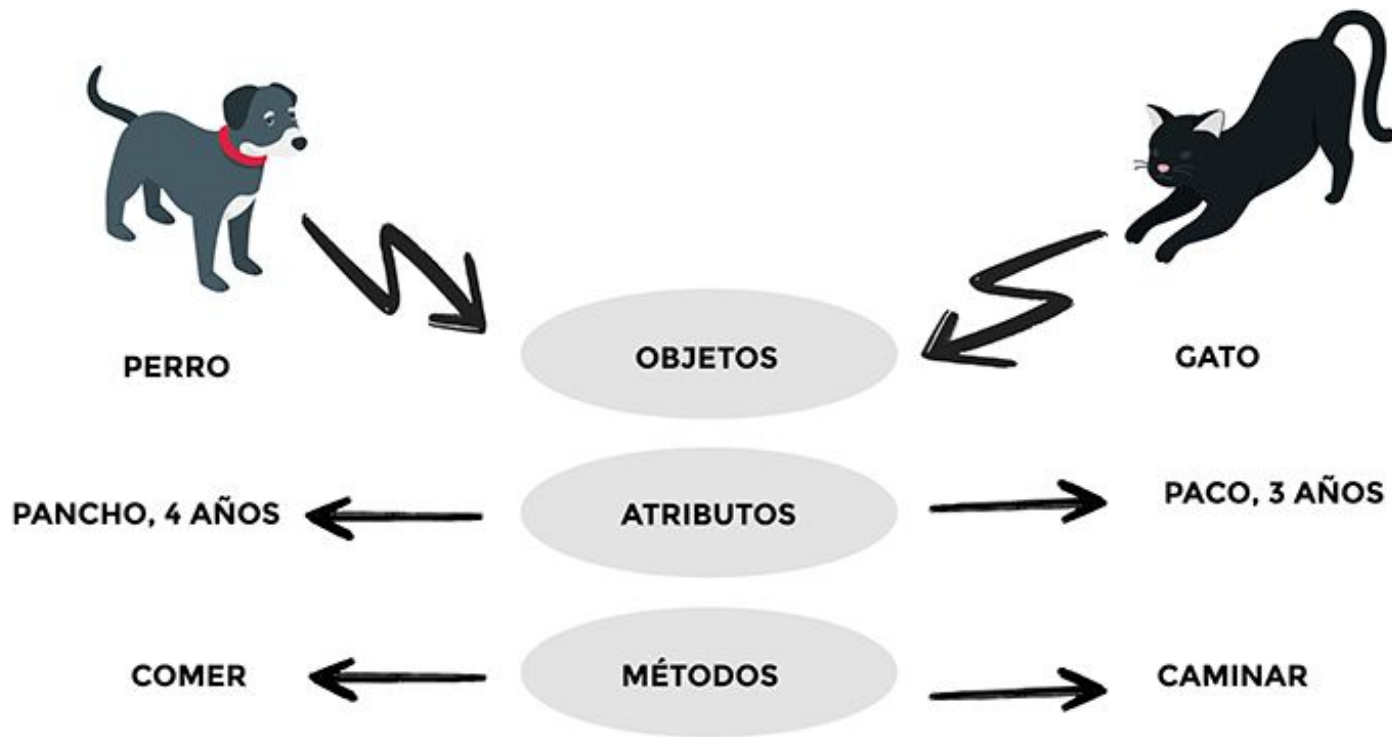
`arr = [  
 [ [1, 2, 3], [4, 5, 6], [7, 8, 9] ],  
 [ [10, 11, 12], [13, 14, 15], [16, 17, 18] ],  
 [ [19, 20, 21], [22, 23, 24], [25, 26, 27] ],  
]`

(0, 0, 0)

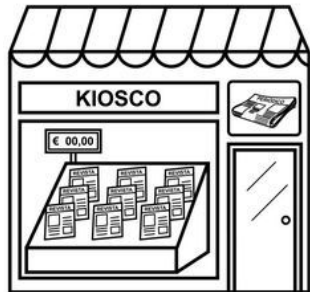
1	2	3
4	5	6
7	8	9

`arr[0][0][0] => 1  
arr[1][0][0] => 2  
arr[0][1][0] => 4  
arr[0][0][1] => 10`

# Objetos



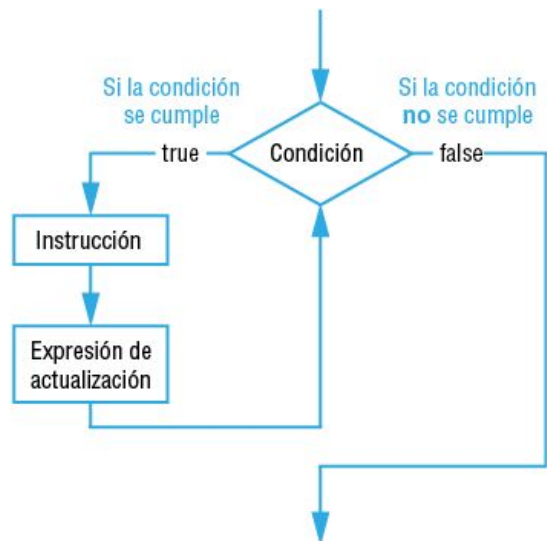
# Trabajando con objetos y arrays



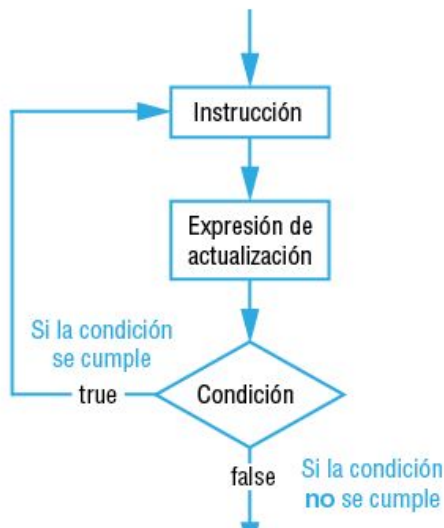
```
let empresa1 = {
  nombre: 'Carlitos',
  rubro: 'Kiosco',
  direccion: 'Siempre Viva 123',
  inicioActividades: '01/01/2000'
};
empresa1.direccion = {
  calle: 'Siempre Viva',
  altura: 123,
  ciudad: 'Springfield'
};
console.log(empresa1.direccion.ciudad);
empresa1.rubro = ['Kiosco', 'Recargas', 'Ventana pagos', 'Encomiendas'];
console.log(empresa1.rubro[2]);
let empresa2 = {
  nombre: 'Chango menos',
  rubro: ['Comida', 'Bebidas', 'Cafeteria'],
  direccion: 'Siempre Viva 456',
  inicioActividades: '12/12/2000'
};
let empresa3 = {
  nombre: 'Tornillazos',
  rubro: ['Ferreteria', 'Carpinteria'],
  direccion: 'Siempre Viva 789',
  inicioActividades: '01/01/2010'
};
```

# Estructuras iterativas - Bucles o Loops

## While y Do...While



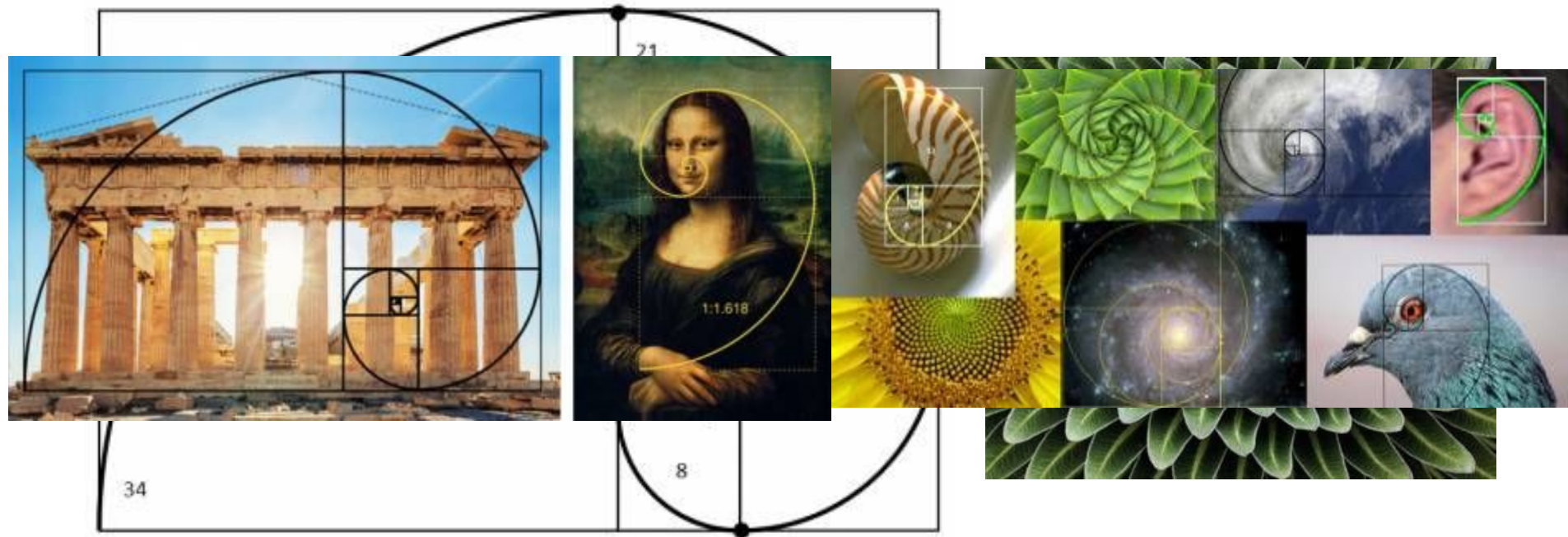
```
while (expresión a evaluar){  
    instrucción a ejecutar si la condición se cumple;  
    actualización de la variable;  
}
```



```
do {  
    instrucción a ejecutar;  
    actualización de la variable;  
} while (expresión a evaluar);
```

# Caso de estudio - La sucesión Fibonacci

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597 ...





**PUERTO** *e-learning*

[www.puertoelearning.com.ar](http://www.puertoelearning.com.ar)



Universidad Autónoma  
de Entre Ríos