

(6015) - Paradigmas de Programación - C1 - 2024/2

[Aula](#) [Participantes](#) [Calificaciones](#)[General](#)[Clase 1](#)[Clase 2](#)

. Hola a todos 🖐️

En esta sección vas a encontrar todo lo que vimos durante la clase II, resúmenes, presentaciones y los ejemplos que realizamos.

Todo este material les será útil no solo para esta cursada, sino para todas las materias de la carrera que tiene por delante.

Recuerden que todos los ejercicios nos pueden realizar en la plataforma <https://replit.com> tenés que sacarte una cuenta lo antes posible para que puedas comenzar a hacer los ejercicios.

Dentro de la plataforma deben seleccionar el lenguaje de programación **NodeJS** que básicamente es el lenguaje JavaScript para poder usarlo del lado del servidor, es decir por fuera de los navegadores web.



Objetos en Javascript

Objetos

Los objetos son los compañeros de ruta de todo JavaScriptero.

En JavaScript, los **objetos literales** son estructuras de datos que se definen utilizando llaves `{}`. Dentro de estas llaves, los datos se organizan en pares **clave: valor**. Cada par está separado por comas, y las claves permiten acceder a los valores asociados a ellas. Estos valores pueden ser de diferentes tipos, como números, cadenas de texto, otros objetos (creando estructuras anidadas) o incluso funciones.

```
const persona = {
  nombre: "Juan",
  edad: 30,
  direccion: {
    ciudad: "Buenos Aires",
    pais: "Argentina"
  },
  saludar: function() {
    console.log("Hola, soy " + this.nombre);
  }
};
```

En este ejemplo:

- `nombre` y `edad` son valores simples (cadena y número).
- `direccion` es un objeto anidado dentro del objeto principal.
- `saludar` es una función que usa la clave `nombre` del mismo objeto para crear un mensaje.

⚠ Entra a esta sección y lee toda la documentación complementaria asociada a esto.



Desafío

Una duda y varios desafíos

Lo que devuelve una función

Si defino

```
function theSpec() {  
  return { height: 200, width: 150 }  
}
```

e invoco varias veces esta función ¿obtengo siempre el mismo objeto, o cada vez un clon distinto?



Reduce

Explicación del método `reduce()`

- `reduce()` es un método que se utiliza en arrays para acumular o "reducir" todos sus elementos en un solo valor.
- Toma una **función de acumulación** como argumento, la cual recibe dos parámetros:
 1. **acumulador**: Es el valor que se va acumulando en cada iteración (el valor que eventualmente será el resultado final).
 2. **elementoActual**: Es el valor del elemento actual del array que se está procesando en ese momento.
- También puede recibir un segundo argumento que será el valor inicial del acumulador.

⚠ Entra y mira el ejemplo que usamos en la clase

⚠ Seguí el siguiente [link](#) si quieres conocer la documentación oficial de la función `reduce`



Ejemplo 1

Si partimos de la siguiente lista de empleados y queremos sumar la cantidad de hijos en total, podemos usar el `reduce` para hacerlo.

⚠ Entra y mira la explicación y la solución

```
1  
2  const empleados = [  
3    {  
4      nombre: 'Gerardo',  
5      edad: 50
```

```

5      edad: 50,
6      hijos:2
7    },
8    {
9      nombre: 'Florencia',
10     edad: 48,
11     hijos:1
12   },
13   {
14     nombre: 'Roman',
15     edad: 52,
16     hijos:0
17   },
18   {
19     nombre: 'Melina',
20     edad: 37,
21     hijos:3
22   }
23 ]
24

```



Ejemplo 2

Si partimos de la siguiente lista de empleados y queremos encontrar el empleado de mayor edad

▲ Entra y mira la explicación y la solución

```

1
2  const empleados = [
3    {
4      nombre: 'Gerardo',
5      edad: 50,
6      hijos:2
7    },
8    {
9      nombre: 'Florencia',
10     edad: 48,
11     hijos:1
12   },
13   {

```

```
13     {
14         nombre: 'Roman',
15         edad: 52,
16         hijos:0
17     },
18     {
19         nombre: 'Melina',
20         edad: 37,
21         hijos:3
22     }
23 ]
24
```



Archivo con la Solucion

Te dejo el archivo con la solución completa de los dos ejercicios para que puedan probarlo, simplemente copio y pega el contenido en Replit y veras los resultados.