

# EJERCICIOS CLASE 8 : Javascript

Basándonos en el modelo de prototipos de Javascript ...

## Ejercicio 1

Crear un objeto literal bajo la variable usuario que contenga un método que va a ser compartido por todos los objetos que se creen a partir de esa interfaz.

## Ejercicio 2

Usar el método create del objeto Object para crear dos objetos cuyo prototipo sea un usuario.

## Ejercicio 3

Crear dos métodos adicionales en el objeto usuario que se llamen crear y extender. Cada función debe retornar una instancia de un objeto nuevo vacío cuyo prototipo sea el contexto de la misma función.

## Ejercicio 4

Admitir un parámetro en el método extends llamado metodos que deberá ser un Object conteniendo todos los métodos de un nuevo prototipo.

## Ejercicio 5

Admitir un parámetro mas en el método extends llamado propiedades. Este deberá ser un Array y puede contener únicamente String que van a ser el nombre de las propiedades que podemos usar en las instancias que creemos usando este prototipo.

## Ejercicio 6

Admitir un parámetro en el método create que sea de tipo Object. El mismo únicamente puede contener los índices con nombres declarados en el punto anterior. Ej.: Si un prototipo lo creamos con `let subproto = Prototipo.extends({metodos:{...}},["prop1"])` no puedo hacer `subproto.create({"prop2":"test"})` pero si `subproto.create({"prop1":"test"})` porque prop1 está definida como propiedad en su prototipo.

## BONUS : Javascript

### Ejercicio 1

Crear prototipos para usuarios del tipo Admin , Regular y Guest que contengan propiedades como :

#### -USUARIO :

##### Métodos

- create : De los puntos anteriores pero con los nuevos métodos y propiedades. La propiedad status debe ser False por defecto.
- extends : De los puntos anteriores pero con los nuevos métodos y propiedades
- setInfo : Actualiza la información del usuario solo si es admin
- getInfo : Devuelve la información del usuario

##### Props

- username : String
- email : String
- password : String
- status : Boolean - Determina si está activo o no

#### -ADMIN :

##### Métodos

- registerUser : Igual a create pero ya le asigna un status True
- activateUser : Le puede activar el status a cualquier usuario.
- removesUser : Le cancela el status a cualquier usuario.

##### Props

- adminId : Number

#### -REGULAR :

##### Métodos

- saludar : Usa su nombre de usuario para mostrar un mensaje por consola
- updateInfo : Cambia la información válida para un usuario regular como nombre y

apellido

#### Props

- nombre : String
- apellido : String

-GUEST :

#### Métodos

- ninguno : Tiene todos los demás métodos restringidos

#### Props

- guestId : Number