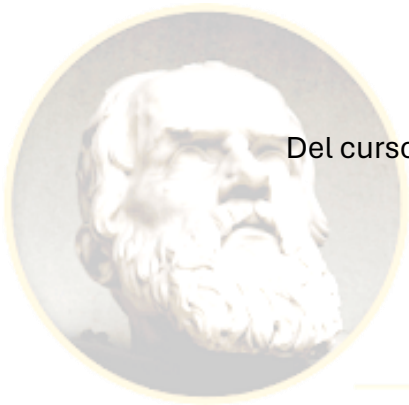


Universidad Galileo  
Facultad de ingeniería de Sistemas,  
informática y Ciencias de la computación  
Diplomado en Desarrollo Fullstack



## Tarea 5

Del curso de Introducción a la Programación

*Galileo*  
UNIVERSIDAD  
La Revolución en la Educación

Luis Diego Orozco López  
24011289  
Sección T

Guatemala, 27 de mayo de 2024

## Tabla de Contenido

Introducción .....	3
Resolución de problemas .....	4
Problema 1 .....	4
Código.....	4
Salida .....	5
Preguntas .....	5
Ejercicio 2 .....	6
Codigo.....	6
Salida .....	7
Preguntas .....	8
Ejercicio 3 .....	9
Codigo.....	9
Salida .....	10
Problema 4.....	11
Código.....	11
Salida .....	12
Conclusión .....	13

## Introducción

La Programación Orientada a Objetos (POO) se basa en el concepto de clases y objetos. Este tipo de programación se utiliza para estructurar un programa de software en piezas simples y reutilizables de planos de código para crear instancias individuales de objetos.

Los objetos en programación representan cosas del mundo real, así como conceptos abstractos con sus características y comportamientos específicos. Un objeto cuenta con su estructura interna que combina variables, funciones y estructuras de datos.

Los elementos de un objeto se dividen en dos categorías principales: propiedades y métodos.

Las propiedades, también conocidas como atributos, incluyen información sobre el objeto. Por ejemplo, si consideramos un objeto automóvil, algunas de sus propiedades serán: el color, la marca, el modelo o el año de fabricación.

Los métodos definen las operaciones que se pueden realizar con el objeto. Por ejemplo, para el objeto Automóvil, los métodos podrían ser acelerar, frenar o girar.

A continuación, se muestra mediante ejercicios prácticos la utilización de objetos y sus atributos y métodos a la hora de programar.

# Resolución de problemas

## Problema 1

### Código

```
1 //Creacion del usuario 0      31 //Creacion del usuario 3
2 let nombre = 'Paola';        32 let nombre3 = 'Paulina';
3 let appeliido = 'Ortiz';     33 let appeliido3 = 'Lopez';
4 let email = 'paola@company.ru'; 34 let email3 = 'paola@company.ru';
5 let direccion = {           35 let direccion3 = {
6   municipio: 'Jocotenango',  36   municipio: 'Jocotenango',
7   calle: 'Calle ancha',     37   calle: 'Calle ancha',
8   numero: 25,               38   numero: 15,
9 };                           39 };
10                               40
11 //Creacion del usuario 1     41 //Creacion del usuario 4
12 let nombre1 = 'Paolo';      42 let nombre4 = 'Alejandra';
13 let appeliido1 = 'Ortega';  43 let appeliido4 = 'Mendez';
14 let email1 = 'paolo@company.ru'; 44 let email4 = 'paola@company.ru';
15 let direccion1 = {         45 let direccion4 = {
16   municipio: 'Jocotenango',  46   municipio: 'Jocotenango',
17   calle: 'Calle ancha',     47   calle: 'Calle ancha',
18   numero: 25,               48   numero: 10,
19 };                           49 };
20                               50
21 //Creacion del usuario 2     51 //Creacion del usuario 5
22 let nombre2 = 'Daniela';    52 let nombre5 = 'Luis';
23 let appeliido2 = 'Perez';   53 let appeliido5 = 'Campos';
24 let email2 = 'paola@company.ru'; 54 let email5 = 'paola@company.ru';
25 let direccion2 = {         55 let direccion5 = {
26   municipio: 'Jocotenango',  56   municipio: 'Jocotenango',
27   calle: 'Calle ancha',     57   calle: 'Calle ancha',
28   numero: 20,               58   numero: 5,
29 };                           59 };
30                               60

61 /*Al mostrar los usuarios de esta manera, se muestran en línea recta y no se especifica a que variable corresponde cada valor, lo cual
62 puede hacer que la forma de leer y entender la informacion sea mas dificil*/
63
64 console.log(nombre, appeliido, email, direccion);
65 console.log(nombre1, appeliido1, email1, direccion1);
66 console.log(nombre2, appeliido2, email2, direccion2);
67 console.log(nombre3, appeliido3, email3, direccion3);
68 console.log(nombre4, appeliido4, email4, direccion4);
69 console.log(nombre5, appeliido5, email5, direccion5);
```

## Salida

```
[Running] node "c:\Users\do30\OneDrive\Documentos\Galileo\Intro a la progra\Tarea 5\Problema 1.js"
Paola Ortiz paola@company.ru { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 25 }
Paolo Ortega paolo@company.ru { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 25 }
Daniela Perez paola@company.ru { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 20 }
Paulina Lopez paola@company.ru { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 15 }
Alejandra Mendez paola@company.ru { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 10 }
Luis Campos paola@company.ru { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 5 }

[Done] exited with code=0 in 0.196 seconds
```

## Preguntas

- **¿Puedes mostrar estas características de los usuarios de manera ordenada y coherente en consola?** Podríamos hacerlo, pero eso conllevaría escribir cada atributo uno por uno dentro del console log para dejar claro a que objeto y atributo pertenece cada valor, lo cual conlleva tiempo y no es muy viable.
- **¿Tienen los usuarios alguna funcionalidad, pueden hacer algo?** Los usuarios no tienen ninguna función.



*Galileo*  
UNIVERSIDAD

La Revolución en la Educación

## Ejercicio 2

### Codigo

Aquí se crean ahora los usuarios como objetos y se les agrega un método RecuperarClave

```
3 //Creacion del usuario 0 como objeto
4 let user = {
5     nombre: 'Paola',
6     apellido: 'Ortiz',
7     email: 'paola@company.ru',
8     direccion: {
9         municipio: 'Jocotenango',
10        calle: 'Calle ancha',
11        numero: 25,
12    },
13    recuperarClave : function () {
14        console.log("Recuperar clave")
15    }
16 }

18 //Creacion del usuario 1 como objeto
19 let user1 = {
20     nombre: 'Paolo',
21     apellido: 'Ortega',
22     email: 'paolo@company.ru',
23     direccion: {
24         municipio: 'Jocotenango',
25         calle: 'Calle ancha',
26         numero: 25,
27     },
28     recuperarClave : function () {
29         console.log("Recuperar clave")
30     }
31 }

33 //Creacion del usuario 2 como objeto
34 let user2 = {
35     nombre: 'Daniela',
36     apellido: 'Perez',
37     email: 'daniela@company.ru',
38     direccion: {
39         municipio: 'Jocotenango',
40         calle: 'Calle ancha',
41         numero: 20,
42     },
43     recuperarClave : function () {
44         console.log("Recuperar clave")
45     }
46 }

48 //Creacion del usuario 3 como objeto
49 let user3 = {
50     nombre: 'Paulina',
51     apellido: 'Lopez',
52     email: 'paulina@company.ru',
53     direccion: {
54         municipio: 'Jocotenango',
55         calle: 'Calle ancha',
56         numero: 15,
57     },
58     recuperarClave : function () {
59         console.log("Recuperar clave")
60     }
61 }

63 //Creacion del usuario 4 como objeto
64 let user4 = {
65     nombre: 'Alejandra',
66     apellido: 'Mendez',
67     email: 'alejandra@company.ru',
68     direccion: {
69         municipio: 'Jocotenango',
70         calle: 'Calle ancha',
71         numero: 10,
72     },
73     recuperarClave : function () {
74         console.log("Recuperar clave")
75     }
76 }

//Creacion del usuario 5 como objeto
let user5 = {
    nombre: 'Luis',
    apellido: 'Campos',
    email: 'luis@company.ru',
    direccion: {
        municipio: 'Jocotenango',
        calle: 'Calle ancha',
        numero: 5,
    },
    recuperarClave : function () {
        console.log("Recuperar clave")
    }
}
```

```
93 //Ahora los usuarios son mostrados en consola como objetos, especificando las propiedades, atributos y metodos
94 console.log(user);
95 console.log(user1);
96 console.log(user2);
97 console.log(user3);
98 console.log(user4);
99 console.log(user5);
```

## Salida

```
[Running] node "c:\Users\do30\OneDrive\Documentos\Galileo\Intro a la progra\Tarea 5\Problema 2.js"
{
  nombre: 'Paola',
  apellido: 'Ortiz',
  email: 'paola@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 25 },
  recuperarClave: [Function: recuperarClave]
}
{
  nombre: 'Paolo',
  apellido: 'Ortega',
  email: 'paolo@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 25 },
  recuperarClave: [Function: recuperarClave]
}
{
  nombre: 'Daniela',
  apellido: 'Perez',
  email: 'daniela@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 20 },
  recuperarClave: [Function: recuperarClave]
}
{
  nombre: 'Paulina',
  apellido: 'Lopez',
  email: 'paulina@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 15 },
  recuperarClave: [Function: recuperarClave]
}
{
  nombre: 'Alejandra',
  apellido: 'Mendez',
  email: 'alejandra@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 10 },
  recuperarClave: [Function: recuperarClave]
}
{
  nombre: 'Luis',
  apellido: 'Campos',
  email: 'luis@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 5 },
  recuperarClave: [Function: recuperarClave]
}

[Done] exited with code=0 in 0.275 seconds
```

## Preguntas

- **¿Qué diferencias conceptuales observas entre el Problema I y el Problema II?** Los usuarios del problema 2 son objetos, los del primer problema, no.
- **¿Para crear usuarios es más fácil y coherente la manera del Problema I o la manera en que se crean en el Problema II?** La segunda manera es mejor, ya que se puede apreciar que es un usuario con sus atributos



*Galileo*  
UNIVERSIDAD

La Revolución en la Educación



## Ejercicio 3

### Codigo

```
196 //se crea un nuevo atributo dpi fuera de la declaracion inicial del objeto
197 user.dpi = "0";
198 user1.dpi = "1";
199 user2.dpi = "2";
200 user3.dpi = "3";
201 user4.dpi = "4";
202 user5.dpi = "5";
```

```
213 /*A cada objeto usuario se le agrega un metodo de cambiar direccion con el uso de Object.seal, el cual se utiliza para cambiar el valor
214 de un atributo ya existente */
215 user.cambiarDireccion = function (){
216     user.direccion = Object.seal({municipio: "Chinautla", calle: "Diagonal 13", numero: "25"});
217 };
218
219 user1.cambiarDireccion = function (){
220     user.direccion = Object.seal({municipio: "Chinautla", calle: "Diagonal 13", numero: "25"});
221 };
222
223 user2.cambiarDireccion = function (){
224     user.direccion = Object.seal({municipio: "Chinautla", calle: "Diagonal 13", numero: "20"});
225 };
226
227 user3.cambiarDireccion = function (){
228     user.direccion = Object.seal({municipio: "Chinautla", calle: "Diagonal 13", numero: "15"});
229 };
230
231 user4.cambiarDireccion = function (){
232     user.direccion = Object.seal({municipio: "Chinautla", calle: "Diagonal 13", numero: "10"});
233 };
234
235 user5.cambiarDireccion = function (){
236     user.direccion = Object.seal({municipio: "Chinautla", calle: "Diagonal 13", numero: "5"});
237 };
```



UNIVERSIDAD  
La Revolución en la Educación

## Salida

Despliegue de los objetos con el nuevo atributo dpi y la dirección cambiada

```
{
  nombre: 'Paola',
  apellido: 'Ortiz',
  email: 'paola@company.ru',
  direccion: { municipio: 'Chinautla', calle: 'Diagonal 13', numero: '5' },
  recuperarClave: [Function: recuperarClave],
  dpi: '0',
  cambiarDireccion: [Function (anonymous)]
}
{
  nombre: 'Paolo',
  apellido: 'Ortega',
  email: 'paolo@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 25 },
  recuperarClave: [Function: recuperarClave],
  dpi: '1',
  cambiarDireccion: [Function (anonymous)]
}
{
  nombre: 'Daniela',
  apellido: 'Perez',
  email: 'paola@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 20 },
  recuperarClave: [Function: recuperarClave],
  dpi: '2',
  cambiarDireccion: [Function (anonymous)]
}
{
  nombre: 'Paulina',
  apellido: 'Lopez',
  email: 'paola@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 15 },
  recuperarClave: [Function: recuperarClave],
  dpi: '3',
  cambiarDireccion: [Function (anonymous)]
}
{
  nombre: 'Alejandra',
  apellido: 'Mendez',
  email: 'paola@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 10 },
  recuperarClave: [Function: recuperarClave],
  dpi: '4',
  cambiarDireccion: [Function (anonymous)]
}
{
  nombre: 'Luis',
  apellido: 'Campos',
  email: 'paola@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: 5 },
  recuperarClave: [Function: recuperarClave],
  dpi: '5',
  cambiarDireccion: [Function (anonymous)]
}
```

## Problema 4

### Código

```
1 //Creacion de una funcion que creara los objetos usuario previamente vistos a partir de parametros que nosotros pasemos a la funcion
2 function createUser (nombre, apellido, email, municipio, calle, numero, dpi){
3     return {
4         nombre: nombre,
5         apellido: apellido,
6         email: email,
7         direccion: {
8             municipio: municipio,
9             calle: calle,
10            numero: numero
11        },
12        recuperarClave : function() {
13            console.log("Recuperando clave");
14        },
15        dpi: dpi,
16        cambiarDireccion : function (municipio, calle, numero){
17            user.direccion = Object.seal({municipio: municipio, calle: calle, numero: numero});
18        }
19    };
20 }
21
22
23 //Creacion de los usuarios mediante la funcion
24 user = createUser("Paola", "Ortiz", "paola@company.ru", "Jocotenango", "Calle ancha", "25", "0" );
25 user1 = createUser("Paolo", "Ortega", "paolo@company.ru", "Jocotenango", "Calle ancha", "25", "1" );
26 user2 = createUser("Daniela", "Perez", "daniela@company.ru", "Jocotenango", "Calle ancha", "20", "2" );
27 user3 = createUser("Paulina", "Lopez", "paulina@company.ru", "Jocotenango", "Calle ancha", "15", "3" );
28 user4 = createUser("Alejandra", "Mendez", "alejandra@company.ru", "Jocotenango", "Calle ancha", "10", "4" );
29 user5 = createUser("Luis", "Campos", "luis@company.ru", "Jocotenango", "Calle ancha", "05", "5" );
30
31 //Despliegue de los objetos usuario creados a partir de una factory function
32 console.log(user);
33 console.log(user1);
34 console.log(user2);
35 console.log(user3);
36 console.log(user4);
37 console.log(user5);
```



## Salida

```
[Running] node "c:\Users\do30\OneDrive\Documentos\Galileo\Intro a la progra\Tarea 5\Problema 4.js"
{
  nombre: 'Paola',
  apellido: 'Ortiz',
  email: 'paola@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: '25' },
  recuperarClave: [Function: recuperarClave],
  dpi: '0',
  cambiarDireccion: [Function: cambiarDireccion]
}
{
  nombre: 'Paolo',
  apellido: 'Ortega',
  email: 'paolo@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: '25' },
  recuperarClave: [Function: recuperarClave],
  dpi: '1',
  cambiarDireccion: [Function: cambiarDireccion]
}
{
  nombre: 'Daniela',
  apellido: 'Perez',
  email: 'daniela@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: '20' },
  recuperarClave: [Function: recuperarClave],
  dpi: '2',
  cambiarDireccion: [Function: cambiarDireccion]
}
{
  nombre: 'Paulina',
  apellido: 'Lopez',
  email: 'paulina@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: '15' },
  recuperarClave: [Function: recuperarClave],
  dpi: '3',
  cambiarDireccion: [Function: cambiarDireccion]
}
{
  nombre: 'Alejandra',
  apellido: 'Mendez',
  email: 'alejandra@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: '10' },
  recuperarClave: [Function: recuperarClave],
  dpi: '4',
  cambiarDireccion: [Function: cambiarDireccion]
}
{
  nombre: 'Luis',
  apellido: 'Campos',
  email: 'luis@company.ru',
  direccion: { municipio: 'Jocotenango', calle: 'Calle ancha', numero: '05' },
  recuperarClave: [Function: recuperarClave],
  dpi: '5',
  cambiarDireccion: [Function: cambiarDireccion]
}

[Done] exited with code=0 in 0.222 seconds
```

## Conclusión

Como se pudo apreciar en estos ejercicios, la creación de objetos facilita la programación al hacerla más natural.

Ver objetos y describirlos es algo que hacemos en el día a día y al implementar esta característica tan humana en la programación, se facilita la manera de escribir código, interactuar con el lenguaje de programación y crear programas que otras personas puedan entender a la hora de colaborar en proyectos grupales.

Los procesos y las funciones que interactúan con el código también serán beneficiadas, ya que se facilita el acceso a la información cuando esta está construida de manera mas ordenada.



*Galileo*  
UNIVERSIDAD

La Revolución en la Educación