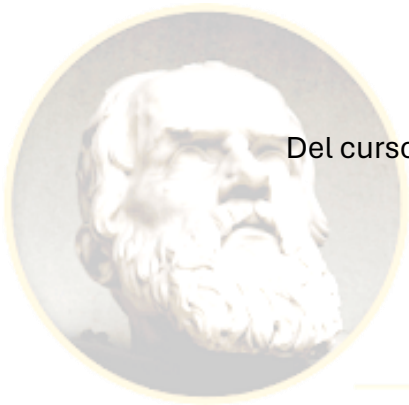


Universidad Galileo  
Facultad de ingeniería de Sistemas,  
informática y Ciencias de la computación  
Diplomado en Desarrollo Fullstack



## Tarea 7

Del curso de Introducción a la Programación

*Galileo*  
UNIVERSIDAD  
La Revolución en la Educación

Luis Diego Orozco López  
24011289  
Sección T

Guatemala, 11 de junio de 2024

## Tabla de Contenido

Resolución de problemas .....	3
Problema 1 .....	3
Código.....	3
Salida .....	3
Ejercicio 2 .....	4
Codigo.....	4
Ejercicio 3 .....	5
Codigo.....	5
Salida .....	5
Problema 4.....	6
Código.....	6
Salida .....	6

# Resolución de problemas

## Problema 1

### Código

```
15 ejercicio 1.js > ...
1  function callPunto(dx, dy){           //funcion que agregara propiedades y metodos al objeto llamado con call()
2      this.x = dx;                     //la propiedad x del objeto sera igual al parametro dx
3      this.y = dy;                     //la propiedad y del objeto sera igual al parametro dy
4      this.mover = function(dx,dy, dz){ //metodo mover que mueve las coordenadas del punto de acuerdo a los valores de los parametros
5          this.x += dx;                 //el valor x del objeto se movera dx unidades
6          this.y += dy;                 //el valor y del objeto se movera dy unidades
7          this.z +=dz;                  //el valor z del objeto se movera dz unidades
8      }
9  }
10
11 function moverPunto(punto, dx, dy){   //funcion que cambiara las propiedades del objeto punto mediante call()
12     callPunto.call(punto, dx, dy);     //se llama al objeto punto para asignarle las propiedades de la funcion callPunto
13 }
14
15 let miPunto = {z: 5};                  //creacion del objeto inicial miPunto
16
17 console.log("Objeto miPunto original");
18 console.log(miPunto);
19 moverPunto(miPunto, 1, 2);              //los valores 1 y 2 se le asignaran al objeto miPunto
20 console.log("Objeto miPunto despues de cambiar sus propiedades");
21 console.log(miPunto);
22 miPunto.mover(5,5,5);
23 console.log("Objeto miPunto despues de moverse 5 unidades en cada vertice");
24 console.log(miPunto);
```

### Salida

```
[Running] node "c:\Users\do30\OneDrive\Documentos\Galileo\Intro a la progra\Tarea 7\ejercicio 1.js"
Objeto miPunto original
{ z: 5 }
Objeto miPunto despues de cambiar sus propiedades
{ z: 5, x: 1, y: 2, mover: [Function (anonymous)] }
Objeto miPunto despues de moverse 5 unidades en cada vertice
{ z: 10, x: 6, y: 7, mover: [Function (anonymous)] }

[Done] exited with code=0 in 0.177 seconds
```

## Ejercicio 2

### Codigo

```
js ejercicio 2.js > ...
1  function callPunto(){           //funcion que agregara propiedades y metodos al objeto llamado con call()
2      this.dibujar = function(dx,dy, dz){ //funcion que dibujara el punto a partir de los valores dados
3          this.x = dx;                //el vertice x sera igual al valor dx asignado
4          this.y = dy;                //el vertice y sera igual al valor dy asignado
5          this.z = dz;                //el vertice z sera igual al valor dz asignado
6      }
7      this.mover = function(dx,dy, dz){ //funcion que aumentara o disminuira el vertice segun el valor asignado
8          this.x += dx;                //x aumentara dx unidades
9          this.y += dy;                //y aumentara dy unidades
10         this.z +=dz;                //z aumentara dz unidades
11     }
12     this.informar = function(dx,dy, dz){ //funcion que muestra las coordenadas donde se ubica el punto actualmente
13         console.log("\n\nCoordenada actual = (" + this.x + "," + this.y + "," + this.z + ")");
14     }
15 }
16
17 let punto = {x:1,y:2,z:3};         //creacion inicial del objeto punto
18
19 function extenderPunto(punto){     //funcion que llama al objeto punto para agregarle las propiedades de la funcion extenderPunto
20     callPunto.call(punto);         //usamos call para llamar a la funcion
21 }
22
23 console.log("\nMostrando punto inicialmente");
24 console.log(punto);
25 extenderPunto(punto);              //se ejecuta la funcion extenderPunto para asignar los metodos
26 console.log("\n\nMostrando punto despues de ser extendido");
27 console.log(punto);
28 punto.dibujar(5,10,15);             //se ejecuta la funcion dibujar para dibujar un punto con coordenadas x=5, y=10, z=15
29 punto.informar();                  //se ejecuta la funcion informar para mostrar las coordenadas
30 punto.mover(2,2,2);                //se ejecuta la funcion mover para mover el punto 2 unidades en cada vertice (x,y,z)
31 punto.informar();                  //se ejecuta la funcion informar nuevamente para mostrar como fue movido el punto
32
```

### Salida

```
[Running] node "c:\Users\do30\OneDrive\Documentos\Galileo\Intro a la progra\Tarea 7\ejercicio 2.js"

Mostrando punto inicialmente
{ x: 1, y: 2, z: 3 }

Mostrando punto despues de ser extendido
{
  x: 1,
  y: 2,
  z: 3,
  dibujar: [Function (anonymous)],
  mover: [Function (anonymous)],
  informar: [Function (anonymous)]
}

Coordenada actual = (5,10,15)

Coordenada actual = (7,12,17)

[Done] exited with code=0 in 0.164 seconds
```

## Ejercicio 3

### Codigo

```
js ejercicio 3.js > ...
1  function applyPunto(punto){           //funcion que agregara propiedades y metodos al objeto llamado con call()
2      this.x = punto.x;                 //el valor del vertice x del objeto punto se le asignara al objeto llamado con apply()
3      this.y = punto.y;                 //el valor del vertice y del objeto punto se le asignara al objeto llamado con apply()
4      this.sumar = function(punto){     //metodo sumar que se le asignara al objeto llamado con apply()
5          suma = punto.x + punto.y;     //se suman los valores x e y del objeto punto y se almacenan en la variable suma
6          console.log("La suma de las coordenadas es " + suma); //se despliega el resultado suma
7      }
8  }
9
10 //creacion de los objetos punto 1 y punto 2
11 punto1 = {x:1,y:1};
12 punto2 = {x:2,y:2};
13
14 function sumarPuntos(punto1, punto2){ //funcion que tomara los dos objetos puntos para despues ejecutar la funcion
15     applyPunto.apply(punto1, [punto2]); //se llama al punto 1 para asignarle los valores de punto 2
16 }
17
18 sumarPuntos(punto1,punto2);           //ejecucion de la funcion
19
20 //Ejecucion de las funciones y despliegue de los resultados en consola
21 console.log("\n\nPunto 1 despues de absorber las propiedades de punto 2 y de agregardo el metodo sumar");
22 console.log(punto1);
23 console.log("\n\nEjecutando punto1.sumar()...\n");
24 punto1.sumar(punto2);
```

### Salida

```
[Running] node "c:\Users\do30\OneDrive\Documentos\Galileo\Intro a la progra\Tarea 7\ejercicio 3.js"

Punto 1 despues de absorber las propiedades de punto 2 y de agregardo el metodo sumar
{ x: 2, y: 2, sumar: [Function (anonymous)] }

Ejecutando punto1.sumar()...

La suma de las coordenadas es 4

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

## Problema 4

### Código

```
JS ejercicio 4.js > ...
1  //funcion que invocara apply para almacenar el punto mas lejano del origen
2  function masLejano(punto){
3      if (Math.abs(this.x + this.y) <= Math.abs(punto.x + punto.y)){ //si el valor absoluto del objeto mas lejano es menor que
4          this.x = punto.x; //el valor absoluto del siguiente punto con el cual lo estamos
5          this.y = punto.y; //comparando, el nuevo punto mas lejano sera el segundo
6      }
7  }
8
9
10 function puntoMasLejano(puntos){
11     let lejano = {x:0, y:0}; //se crea un objeto lejano para comparar con los demas y almacenar el mas lejano
12     for (let i = 0; i < puntos.length; i++) { //ciclo for donde se recorre el array y se compara con el objeto pivote lejano
13         masLejano.apply(lejano, [puntos[i]]); //se utiliza apply para que lejano absorba las propiedades del punto mas lejano
14     } //si en dado casi aplica
15     return lejano; //se retorna el objeto mas lejano
16 }
17
18 //creacion de los puntos a comparar
19 punto1 = {x:1, y:1};
20 punto2 = {x:2, y:2};
21 punto3 = {x:3, y:3};
22
23 let puntos = [punto1,punto2,punto3]; //se insertan los puntos en el array
24
25 //resultado
26 console.log("El punto mas lejano del origen es: ");
27 console.log(puntoMasLejano(puntos));
```

### Salida

```
[Running] node "c:\Users\do30\OneDrive\Documentos\Galileo\Intro a la progra\Tarea 7\ejercicio 4.js"
El punto mas lejano del origen es:
{ x: 3, y: 3 }

[Done] exited with code=0 in 0.174 seconds
```