



SEDE  
SANTO DOMINGO

**UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE**

**SEDE SANTO DOMINGO DE LOS TSÁCHILAS**

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN - DCCO-SS**

**CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**



<b>PERIODO</b>	:	202450 Mayo – Septiembre 2024
<b>ASIGNATURA</b>	:	Programación Orientada a Objetos
<b>TEMA</b>	:	Proyecto Grupal
<b>ESTUDIANTE</b>	:	Rodríguez Montalván Bruce Leroy
<b>NIVEL-PARALELO - NRC:</b>	:	Segundo A: 15279
<b>DOCENTE</b>	:	Ing. Verónica Martínez C., Mgs.
<b>FECHA DE ENTREGA</b>	:	07 de mayo de 2023.

**SANTO DOMINGO – ECUADOR**

## Contenido

1. Introducción.....	3
2. Objetivos .....	3
Objetivo General: .....	3
Objetivos Específicos:.....	3
3. Desarrollo / Marco Teórico/ Práctica .....	3
Marco Teórico: .....	3
Desarrollo: .....	4
CAPTURA DEL CODIGO .....	4
Clase principal Main.....	4
Clase padre (GrupoD_Event): .....	4
Clase hija (GrupoD_Concert).....	5
Clase hija (GrupoD_Party).....	6
Clase Event Management.....	8
4. Conclusiones.....	9
5. Recomendaciones .....	10
6. Bibliografía/ Referencias.....	10
7. Anexos: .....	10
8. Legalización de documento.....	10

# Fundamentos de la Programación

## 1. Introducción

El programa realizado como proyecto abarca una variedad de temas esenciales en Programación Orientada a Objetos (POO) implementada en Java, que incluye guardar datos y la lectura de datos en JSON y CSV, también aplicamos herencia, polimorfismo y encapsulamiento, se aplican buenas prácticas de programación, también aborda la seguridad del código mediante el uso de modificadores de acceso como 'private' y 'protected', finalmente se realiza el modelado UML en clases y usos.

## 2. Objetivos

### Objetivo General:

Crear un programa en Java que cubra aspectos clave de la Programación Orientada a Objetos, como el manejo de datos en JSON y CSV, el uso de herencia, polimorfismo y encapsulamiento, junto con la aplicación de buenas prácticas de programación y la seguridad del código mediante modificadores de acceso.

### Objetivos Específicos:

- Aplicar el concepto de herencia para compartir atributos comunes en otras clases
- Implementar polimorfismo para permitir que la funciones realicen diferentes acciones según la situación
- Incorporar buenas prácticas de programación para mejorar la legibilidad y eficiencia del código

## 3. Desarrollo / Marco Teórico/ Práctica

### Marco Teórico:

**NO APLICA**

Desarrollo:

## CAPTURA DEL CODIGO

### Clase principal Main

```
1 package GrupoD_Proyecto;
2
3 public class GrupoD_Main {
4     public static void main(String[] args) {
5         GrupoD_EventManagement gestionEventos = new GrupoD_EventManagement(0);
6         gestionEventos.ejecutarGestionEventos();
7     }
8 }
9
```

### Clase padre (GrupoD\_Event):

```
1 package GrupoD_Proyecto;
2
3 import java.util.ArrayList; //LIBRERIA ARRAYLIST
4 import java.util.Scanner;
5
6 public abstract class GrupoD_Event {
7     // ATRIBUTOS
8
9     protected String place, eventName, band, personName, personID, client;
10    protected int day, month, year;
11    protected Scanner scanner;
12    private int nPeople;
13    // DATOS DEL ARRAY <STRING> DEPENDE DEL TIPO DE DATO QUE SE VAYA A GUARDAR
14    protected ArrayList<String> listPeople = new ArrayList();
15
16    // CONSTRUCTOR
17    public GrupoD_Event(int day, int month, int year, String place, String eventName) {
18        this.day = day;
19        this.month = month;
20        this.year = year;
21        this.place = place;
22        this.eventName = eventName;
23        band = "";
24        nPeople = 0;
25        personName = "";
26        personID = "";
27        client = "";
28        scanner = new Scanner(System.in);
29    }
30
31    public void requestEventData() {
32        do {
33            System.out.print("Ingrese el día del evento (1-31): ");
34            day = scanner.nextInt();
35        } while (day < 1 || day > 31);
36        do {
37            System.out.print("Ingrese el mes del evento (1-12): ");
38            month = scanner.nextInt();
39        } while (month < 1 || month > 12);
40        do {
41            System.out.print("Ingrese el año del evento (2024): ");
42            year = scanner.nextInt();
43        } while (year < 2024 || year >= 2025);
44        scanner = new Scanner(System.in); // Limpiar el buffer
45        System.out.print("Ingrese el lugar del evento: ");
46        place = scanner.nextLine();
47        do {
48            System.out.println("Ingrese cuantas personas desea comprar el boleto [1-4]:");
49            nPeople = scanner.nextInt();
50        } while (nPeople < 1 || nPeople > 4);
51        // FOR DE PERSONAS
52        for (int i = 0; i < nPeople; i++) {
53            scanner = new Scanner(System.in); // LIMPIA BUFFER
54            System.out.print("Nombre de la persona: ");
55            personName = scanner.nextLine();
56            do {
57                System.out.print("Ingrese cédula (10 dígitos) ");
58                personID = scanner.nextLine();
59            } while (personID.length() != 10);
60            client = personName + " - " + personID;
61            listPeople.add(client); // NOMBRE.ADD(DATO A GUARDAR)
62        }
63    }
64
65    // METODO POLIMORFISMO
66    public abstract void saveData();
67
68    public abstract void readData();
69
70    public abstract void selectpackageName();
71 }
```

## Clase hija (GrupoD\_Concert)

```

1 package GrupoD_Proyecto;
2
3 import java.io.BufferedReader;
4 import java.io.BufferedWriter;
5 import java.io.File;
6 import java.io.FileReader;
7 import java.io.FileWriter;
8 import java.io.IOException;
9
10 public class GrupoD_Concert extends GrupoD_Event {
11     protected String bandName, packageName;
12     private int opMenu;
13     private final String file_CSV = "DatosEvento.csv"; // VA A ESTAR CONSTANTE HASTA QUE TERMINE EL PROGRAMA
14     private File file;
15     private String lineReader;
16
17     public GrupoD_Concert(int day, int month, int year, String place, String bandName, String eventName,
18         String packageName, int opMenu) {
19         super(day, month, year, place, eventName);
20         this.bandName = bandName;
21         this.opMenu = opMenu;
22         this.packageName = packageName;
23         this.file = new File(file_CSV);
24         lineReader = "";
25     }
26
27     public void addConcert() {
28         eventName = "Concierto";
29         requestEventData();
30         System.out.print("Ingrese el nombre de la banda/artista: ");
31         bandName = scanner.nextLine();
32         do {
33             System.out.println("¿Desea comprar un paquete para el concierto? (Si/No)");
34             packageName = scanner.nextLine().trim().toLowerCase(); // ELIMINAR ESPACIOS EN BLANCO Y CONVERTIR EN MINUSCULA
35         } while (!packageName.equals("si") && !packageName.equals("no"));
36         if (packageName.equalsIgnoreCase("si")) {
37             selectpackageName();
38         } else {
39             packageName = "NINGUNO";
40         }
41         client = "";
42         for (int i = 0; i < listPeople.size(); i++) {
43             client += listPeople.get(i);
44             // EN EL ÚLTIMO ÍNDICE YA NO AÑADE ","
45             if (i != listPeople.size() - 1) {
46                 client += ", ";
47             }
48         }
49         System.out.println("*****");
50         System.out.println("LOS DATOS DE concierto SON:");
51         System.out.println("Cliente/s: " + client);
52         System.out.println("Fecha del concierto: " + day + "/" + month + "/" + year);
53         System.out.println("Banda/Artista: " + bandName);
54         System.out.println("Lugar del concierto: " + place);
55         saveData();
56         System.out.println("*****");
57         readData();
58         System.out.println("*****");
59     }
60
61     public void selectpackageName() {
62         do {
63             System.out.println("ELIJA ALGUNA OPCIÓN");
64             System.out.println("[1] PAQUETE ECONÓMICO ($30)");
65             System.out.println("[2] PAQUETE FAMILIAR ($80)");
66             System.out.println("[3] PAQUETE VIP ($150)");
67             System.out.println("[4] SALIR");
68             opMenu = scanner.nextInt();
69             switch (opMenu) {
70                 case 1:
71                     System.out.println("Has elegido paquete económico. Incluye:");
72                     System.out.println("Entrada general al concierto");
73                     System.out.println("Asiento reservado");
74                     packageName = "económico";
75                     opMenu = 4;
76                     break;
77                 case 2:
78                     System.out.println("Has elegido paquete familiar. Incluye:");
79                     System.out.println("4 entradas generales al concierto");
80                     System.out.println("Asientos juntos");
81                     System.out.println("4 bebidas gratis");
82                     System.out.println("Descuento en alimentos");
83                     packageName = "familiar";
84                     opMenu = 4;
85                     break;
86                 case 3:
87                     System.out.println("Has elegido paquete vip. Incluye:");
88                     System.out.println("Entrada VIP al concierto");

```

## Fundamentos de la Programación

```
89         System.out.println("Asiento en primera fila");
90         System.out.println("Estacionamiento VIP");
91         packageName = "VIP";
92         opMenu = 4;
93         break;
94     case 4:
95         System.out.println("Ha salido, muchas gracias");
96         break;
97     default:
98         System.out.println("Ingrese una opcion correcta 1-4");
99     }
100 } while (opMenu != 4);
101 }
102
103 // SOBRESERIBIR METODO POLIMORFISMO
104 @Override
105 public void saveData() {
106     // METODO GUARDAR DATOS CSV
107     // MANEJO ERRORES files
108     try (BufferedWriter writer = new BufferedWriter(new FileWriter(file_CSV, true))) {
109         //SI EL ARCHIVO NO EXISTE, SE ESCRIBE LA CABECERA
110         if (file.length() == 0) {
111             writer.write("type evento, band/artista, day, month, año, place,packageName");
112             writer.newLine(); // SIMULA ENTER
113         }
114         //ESCRIBIR DATOS
115         writer.write(eventName + "," + bandName + "," + day + "/" + month + "/" + year + "," + place + ","
116             + packageName);
117         writer.newLine(); // SIMULA ENTER
118         System.out.println("Los datos han sido guardados en " + file_CSV);
119     } catch (IOException e) {
120         System.out.println("Error al guardar los datos en el file CSV");
121     }
122 }
123
124 // POLIMORFISMO SE UTILIZA PARA MISMO METODO, DIFERENTE LOGICA
125 @Override // SOBRESERIBIR METODO
126 public void readData() {
127     try (BufferedReader leer = new BufferedReader(new FileReader(file_CSV))) {
128         // LINEREADER ES PARA LEER LINEA A LINEA
129         while ((lineReader = leer.readLine()) != null) {
130             System.out.println(lineReader);
131         }
132     } catch (IOException e) {
133         System.out.println("No se pudo leer el file");
134     }
135 }
136 }
```

### Clase hija (GrupoD\_Party)

```
1 package GrupoD_Proyecto;
2
3 import java.io.BufferedReader;
4 import java.io.FileReader;
5 import java.io.FileWriter;
6 import java.io.IOException;
7
8 import org.json.simple.JSONObject; //MANEJO DE OBJETOS
9
10 public class GrupoD_Party extends GrupoD_Event {
11     protected JSONObject fiestaJSON;
12     protected String type, packageName;
13     private String lineReader;
14     private int opMenu;
15
16 public GrupoD_Party(int day, int month, int year, String place, String type, String eventName, int opMenu,
17     String packageName) {
18     super(day, month, year, place, eventName);
19     this.type = type;
20     this.packageName = packageName;
21     this.opMenu = opMenu;
22     fiestaJSON = new JSONObject();
23     lineReader = "";
24 }
25
26 public void agregarFiesta() {
27     eventName = "fiesta";
28     requestEventData();
29
30     System.out.print("Ingrese el tipo de la fiesta: ");
31     type = scanner.nextLine();
32     do {
33         System.out.println("¿Desea comprar un paquete para la fiesta? (Si/No)");
34         packageName = scanner.nextLine().trim().toLowerCase(); // ELIMINAR ESPACIOS EN BLANCO Y CONVERTIR EN MINUSCULA
35     } while (!packageName.equals("si") && !packageName.equals("no"));
36     if (packageName.equalsIgnoreCase("si")) {
37         selectpackageName();
38     } else {
39         packageName = "NINGUNO";
40     }
41     client = "";
42     for (int i = 0; i < listPeople.size(); i++) {
43         client += listPeople.get(i);
44         // EN EL ÚLTIMO ÍNDICE YA NO AÑADE ","
45     }
```

## Fundamentos de la Programación

```
45         if (i != listPeople.size() - 1) {
46             client += ", ";
47         }
48     }
49     System.out.println("*****");
50     System.out.println("LOS DATOS DE LA FIESTA SON:");
51     System.out.println("Cliente/s: " + client);
52     System.out.println("Tipo de fiesta: " + type);
53     System.out.println("Lugar de la fiesta: " + place);
54     System.out.println("Fecha de la fiesta: " + day + "/" + month + "/" + year);
55     saveData();
56     System.out.println("*****");
57     System.out.println("LECTURA Fiesta.JSON");
58     readData();
59     System.out.println("*****");
60 }
61
62 @Override
63 public void selectpackageName() {
64     do {
65         System.out.println("ELIJA ALGUNA OPCIÓN");
66         System.out.println("[1] PAQUETE ECONÓMICO ($10)");
67         System.out.println("[2] PAQUETE FAMILIAR ($40)");
68         System.out.println("[3] PAQUETE VIP ($70)");
69         System.out.println("[4] SALIR");
70         opMenu = scanner.nextInt();
71         switch (opMenu) {
72             case 1:
73                 System.out.println("Has elegido paquete económico. Incluye:");
74                 System.out.println("Acceso a la pista de baile");
75                 System.out.println("1 bebida gratis");
76                 System.out.println("Acceso a una montha compartida");
77                 packageName = "económico";
78                 opMenu = 4;
79                 break;
80             case 2:
81                 System.out.println("Has elegido paquete familiar. Incluye:");
82                 System.out.println("Acceso prioritario a actividades familiares");
83                 System.out.println("Asientos juntos");
84                 System.out.println("4 bebidas sin alcohol gratis");
85                 packageName = "familiar";
86                 opMenu = 4;
87                 break;
88             case 3:
89                 System.out.println("Has elegido paquete vip. Incluye:");
90                 System.out.println("Fotos con el anfitrión o celebridad de la fiesta");
91                 System.out.println("Servicio de monthero exclusivo");
92                 System.out.println("Acceso a aperitivo exclusivos");
93                 packageName = "VIP";
94                 opMenu = 4;
95                 break;
96             case 4:
97                 System.out.println("Ha salido, muchas gracias");
98                 break;
99             default:
100                 System.out.println("Ingrese una opcion correcta 1-4");
101         }
102     } while (opMenu != 4);
103 }
104
105 @Override
106 public void saveData() {
107     // INICIALIZA JSON
108     fiestaJSON = new JSONObject();
109     // GUARDAR EN JSON
110     fiestaJSON.put("Dia:", day);
111     fiestaJSON.put("Mes:", month);
112     fiestaJSON.put("Año:", year);
113     fiestaJSON.put("Lugar:", place);
114     fiestaJSON.put("Tipo:", type);
115     fiestaJSON.put("Paquete:", packageName);
116     // INICIAR CLIENTE VACIO
117     client = "";
118     // FOR
119     for (int i = 0; i < listPeople.size(); i++) {
120         client += listPeople.get(i);
121         // EN EL ÚLTIMO ÍNDICE YA NO AÑADE ","
122         if (i != listPeople.size() - 1) {
123             client += ", ";
124         }
125     }
126     fiestaJSON.put("clients", client);
127     // GUARDAR FIESTA EN ARCHIVO .JSON
128     try (FileWriter file = new FileWriter("Fiesta.json")) {
129         // ESCRITURA JSON
130         file.write(fiestaJSON.toString());
131         // LIMPIAR BUFFER ARCHIVO
132         file.flush();
133     }
```

## Fundamentos de la Programación

```
133         System.out.println("Datos de la fiesta han sido guardados");
134     } catch (Exception e) {
135         // IMPRIME ERRORES SI NO GUARDA EL ARCHIVO
136         System.out.println("El archivo no existe");
137     }
138 }
139
140 @Override
141 public void readData() {
142     try (BufferedReader leer = new BufferedReader(new FileReader("Fiesta.json"))) {
143         while ((lineReader = leer.readLine()) != null) {
144             System.out.println(lineReader);
145         }
146     } catch (IOException e) {
147         System.out.println("No se pudo leer el archivo");
148     }
149 }
150 }
```

## Clase Event Management

```
1 package GrupoD_Proyecto;
2 import java.util.Scanner;
3 public class GrupoD_EventManagement {
4     // ATRIBUTOS
5     private Scanner scanner;
6     private int opMenu;
7     private GrupoD_Concert concierto;
8     private GrupoD_Party fiesta;
9     // CONSTRUCTOR
10    public GrupoD_EventManagement(int opMenu) {
11        this.opMenu = opMenu;
12        scanner = new Scanner(System.in);
13        concierto = new GrupoD_Concert(0, 0, 0, "", "", "", "", 0);
14        fiesta = new GrupoD_Party(0, 0, 0, "", "", "", 0, "");
15    }
16    public void ejecutarGestionEventos() {
17        do {
18            System.out.println("SALON DE EVENTOS");
19            System.out.println("ELIJA ALGUNA OPCION");
20            System.out.println("[1] CONCIERTO");
21            System.out.println("[2] FIESTA");
22            System.out.println("[3] SALIR");
23            opMenu = scanner.nextInt();
24            scanner = new Scanner(System.in); // LIMPIAR BUFFER
25            switch (opMenu) {
26                case 1:
27                    concierto.addConcert();
28                    break;
29                case 2:
30                    fiesta.agregarFiesta();
31                    break;
32                case 3:
33                    System.out.println("Ha salido del programa");
34                    break;
35                default:
36                    System.out.println("Ingrese alguna opción disponible 1-3");
37                    break;
38            }
39        } while (opMenu != 3);
40        scanner.close();
41    }
42 }
```



## CAPTURAS DE EJECUCIÓN

```
SALON DE EVENTOS
ELIJA ALGUNA OPCION
[1] CONCIERTO
[2] FIESTA
[3] SALIR
-9
Ingrese alguna opción disponible 1-3
SALON DE EVENTOS
ELIJA ALGUNA OPCION
[1] CONCIERTO
[2] FIESTA
[3] SALIR
1
Ingrese el día del evento (1-31): -9
Ingrese el día del evento (1-31): 2
Ingrese el mes del evento (1-12): 15
Ingrese el mes del evento (1-12): 12
Ingrese el año del evento (2024): 2025
Ingrese el año del evento (2024): 2024
Ingrese el lugar del evento: Santo Domingo
Ingrese cuantas personas desea comprar el boleto [1-4]
-9
Ingrese cuantas personas desea comprar el boleto [1-4]
5
Ingrese cuantas personas desea comprar el boleto [1-4]
2
Nombre de la persona: Bruce
Ingrese cédula (10 dígitos) 2300446305
Nombre de la persona: Paulina
Ingrese cédula (10 dígitos) 4589671230
Ingrese el nombre de la banda/artista: Aventura
¿Desea comprar un paquete para el concierto? (Si/No)
Quitas
¿Desea comprar un paquete para el concierto? (Si/No)
Si
ELIJA ALGUNA OPCIÓN
[1] PAQUETE ECONÓMICO ($30)
[2] PAQUETE FAMILIAR ($80)
[3] PAQUETE VIP ($150)
[4] SALIR
-9
Ingrese una opción correcta 1-4
ELIJA ALGUNA OPCIÓN
[1] PAQUETE ECONÓMICO ($30)
[2] PAQUETE FAMILIAR ($80)
[3] PAQUETE VIP ($150)
[4] SALIR
2
Has elegido paquete familiar. Incluye:
4 entradas generales al concierto
Asientos juntos
4 bebidas gratis
Descuento en alimentos
*****
LOS DATOS DE concierto SON:
Cliente/s: Bruce - 2300446305, Paulina - 4589671230
Fecha del concierto: 2/12/2024
Banda/Artista: Aventura

*****
LECTURA Fiesta.JSON
{"clients":"Manuel - 7845963210, Belen - 7485963210","Año":":2024","Paquete":":VIP","Tipo":":Infantil","Lugar":":Manta","Dia":":2","Mes":":2}
*****
SALON DE EVENTOS
ELIJA ALGUNA OPCION
[1] CONCIERTO
[2] FIESTA
[3] SALIR
3
Ha salido del programa

Lugar del concierto: Santo Domingo
Los datos han sido guardados en DatosEvento.csv
*****
type evento, band/artista, day, month, año, place, packageName
concierto,aventura,1/1/2024,gfhgf,familiar
Concierto,Aventura,2/12/2024,Santo Domingo,familiar
*****
SALON DE EVENTOS
ELIJA ALGUNA OPCION
[1] CONCIERTO
[2] FIESTA
[3] SALIR
2
Ingrese el día del evento (1-31): -9
Ingrese el día del evento (1-31): 2
Ingrese el mes del evento (1-12): 14
Ingrese el mes del evento (1-12): 2
Ingrese el año del evento (2024): 2023
Ingrese el año del evento (2024): 2024
Ingrese el lugar del evento: Manta
Ingrese cuantas personas desea comprar el boleto [1-4]
5
Ingrese cuantas personas desea comprar el boleto [1-4]
-9
Ingrese cuantas personas desea comprar el boleto [1-4]
2
Nombre de la persona: Manuel
Ingrese cédula (10 dígitos) 7845963210
Nombre de la persona: Belen
Ingrese cédula (10 dígitos) 7485963210
Ingrese el tipo de la fiesta: Infantil
¿Desea comprar un paquete para la fiesta? (Si/No)
nop
¿Desea comprar un paquete para la fiesta? (Si/No)
Si
ELIJA ALGUNA OPCIÓN
[1] PAQUETE ECONÓMICO ($10)
[2] PAQUETE FAMILIAR ($40)
[3] PAQUETE VIP ($70)
[4] SALIR
-8
Ingrese una opción correcta 1-4
ELIJA ALGUNA OPCIÓN
[1] PAQUETE ECONÓMICO ($10)
[2] PAQUETE FAMILIAR ($40)
[3] PAQUETE VIP ($70)
[4] SALIR
3
Has elegido paquete vip. Incluye:
Fotos con el anfitrión o celebridad de la fiesta
Servicio de monthero exclusivo
Acceso a aperitivo exclusivos
*****
LOS DATOS DE LA FIESTA SON:
Cliente/s: Manuel - 7845963210, Belen - 7485963210
Tipo de fiesta: Infantil
Lugar de la fiesta: Manta
Fecha de la fiesta: 2/2/2024
Datos de la fiesta han sido guardados
*****
```

## 4. Conclusiones

- El programa logró abarcar los ítems del silabo, cumpliendo con el manejo de datos, seguridad del código y buenas prácticas de programación
- La aplicación de polimorfismo, herencia y encapsulamiento permitió mejorar el diseño y estructura del código
- Las pruebas realizadas en el código ha demostrado su importancia al detectar y corregir errores en etapas del desarrollo del código

### 5. Recomendaciones

- Aconsejo realizar una planificación del proyecto antes de iniciar la programación. Esto con el fin de identificar los requisitos y diseñar una estructura para el código
- Mantener una documentación clara y completa del código es fundamental para facilitar su comprensión y mantenimiento
- Personalizar nuestro entorno de desarrollo según nuestras necesidades y preferencias puede aumentar considerablemente nuestra productividad y comodidad al programar

### 6. Bibliografía/ Referencias

### 7. Anexos:

### 8. Legalización de documento

**Nombres y Apellidos:** Bruce Leroy Rodríguez Montalván

**CI:** 2300446305

**Firma:**

A handwritten signature in black ink, appearing to read 'Bruce Leroy Rodríguez Montalván', written over a horizontal line.