

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**

**LENGUAJE DE PROGRAMACIÓN 2**

**2da. práctica (tipo b)**  
**(Primer Semestre 2023)**

**Indicaciones Generales:**

- Tiempo estimado: 1h 50 minutos
- Se les recuerda que, de acuerdo al reglamento disciplinario de nuestra institución, constituye una falta grave copiar del trabajo realizado por otro estudiante o cometer plagio para el desarrollo de esta práctica.
- Está permitido el uso de apuntes de clase, diapositivas, ejercicios de clase y código fuente. (Debe descargarlos antes de iniciar con la solución del enunciado)
- No está permitido el uso de entornos de desarrollo integrado (IDEs). Debe utilizar un editor de texto: Notepad++, Sublime, etc.
- Está permitido el uso de Internet (únicamente para consultar páginas oficiales de Microsoft, Oracle y PAIDEIA). No obstante, está prohibida toda forma de comunicación con otros estudiantes o terceros.

**PARTE PRÁCTICA (20 puntos)**

PUEDE UTILIZAR MATERIAL DE CONSULTA.

Se considerará en la calificación el uso de buenas prácticas de programación (aquellas vistas en clase).

**PREGUNTA 1 (20 puntos)**

La institución conocida como **APDAYC** (*Asociación Peruana de Autores y Compositores*) está encargada de velar por los derechos de los autores y cantautores peruanos y extranjeros que se encuentran afiliados. Es por ello, que desea desarrollar un sistema informático que les permita tener un registro de todas las **productoras**, así como los eventos que estas realizan en el país.

Una productora organiza uno o más eventos en el país. Se requiere que el sistema permita registrar el nombre de las empresas productoras, así como un **identificador numérico que permita diferenciar a cada una de ellas en el sistema**. Con respecto a los eventos, estos pueden ser **de dos tipos: Obras Teatrales y Conciertos**. Para ambos tipos de eventos se requiere registrar el **nombre, el costo de realización del evento, un indicador para saber si el evento ha sido benéfico y que el sistema genere un identificador numérico correlativo**. En relación a las **obras teatrales se requiere además que el sistema maneje la cantidad total de actos teatrales que presentará y un indicador para saber si se trata o no de una obra teatral inmersiva**. Para el caso de los conciertos, se requiere registrar si es que son o no **acústicos y si ofrecerán tecnología asistiva para personas con discapacidad** (como los chalecos especiales que fueron utilizados en el concierto de Coldplay para personas con discapacidad auditiva).

Un mismo evento puede realizarse en distintas **funciones, y cada función llevarse a cabo en un local específico**. Por ejemplo, el evento "VIVO X EL ROCK" podría llevarse a cabo el 01 de abril del 2023 en el Estadio Nacional a las 7:00 p.m. y repetirse el día 02 de abril del 2023 en el Estadio San Marcos a las 8:00 p.m. Es decir, el evento puede presentar distintas funciones (en fechas específicas, a una hora de inicio, hora fin) y cada función realizarse en un local diferente. Con respecto a la **función del evento, se requiere registrar la fecha, la hora inicio y la hora fin. En relación con el local se requiere registrar el nombre del local, su dirección y la capacidad de personas que admite, es decir, el aforo**.

En un evento, se pueden presentar uno o varios artistas. Los artistas pueden ser agrupaciones o personas. De todos los artistas (sean agrupaciones o personas) se necesita registrar el nombre artístico y el origen del artista (el cual puede ser **NACIONAL o INTERNACIONAL**). Con respecto a las agrupaciones, se requiere que el sistema además permita registrar su año de conformación. En relación a los artistas que son personas se requieren datos adicionales como su nombre, apellido, fecha de nacimiento y género (que debe manejarse como carácter [M: en caso de que sea masculino y F: en caso de que sea femenino]). Las agrupaciones están conformadas por múltiples personas y se requiere que el sistema permita guardar esta relación para en un futuro sea posible consultar los miembros que conforman cada una de las agrupaciones que son registradas.

Por último, se exige que tanto los **eventos como los artistas sean consultables**, es decir, que permitan devolver a modo de cadena de texto algunos de sus datos. Si se trata de una agrupación, se devolverá la palabra "AGRUPACION:", seguido del nombre artístico, el origen y el año de conformación. Si se trata de una persona, se devolverá la palabra: "PERSONA:" seguido del nombre artístico y el origen. En el caso de eventos, si se trata de una obra teatral, se devolverá la frase: "OBRA TEATRAL:", seguido del nombre y los indicadores de si es benéfica e inmersiva. Si se trata de un concierto, se devolverá la frase: "CONCIERTO:", seguido del nombre y los indicadores de si es acústico y si proveerá tecnología asistiva. Finalmente, indistintamente de si se trata de una obra o concierto, se deberá devolver posteriormente la lista de artistas asociados al evento así como la lista de funciones indicando fecha, hora de inicio y nombre del local donde se llevará a cabo cada función.

Se ha realizado un análisis preliminar y se han detectado algunas clases de manera genérica:

- **IConsultable:** Clase de tipo interface que define la obligación de consultar los datos de un evento y un artista.
- **OrigenArtista:** Clase de tipo enumerate que define los dos tipos de origen (NACIONAL o INTERNACIONAL) de los artistas.
- **Productora:** Clase que define a las productoras que organizan eventos en nuestro país. Dentro de sus principales funciones está la de consultar los eventos que producirá en el país.

Se le solicita elaborar un programa en JAVA que permita dar soporte a la lógica de negocio mencionada y además permita imprimir el reporte que se muestra en la Figura 02.

Para validar el modelado de clases realizar y como parte de las pruebas del sistema, se cuenta con el código fuente de clase Principal programación definida en la Figura 01 y disponible en PAIDEIA. Asimismo, se solicita incorporar el código mostrado en la Figura 03 dentro de la clase "Evento" y reutilizarlo en sus clases derivadas empleando "super".

```
import java.util.ArrayList;
import java.text.SimpleDateFormat;
import java.time.LocalDateTime;
/* Colocar sus datos
    Nombre: ...
    Apellido: ...
*/
class Principal{
    public static void main(String[] args) throws Exception{
        //Creamos un objeto para manejo de fechas
        SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");

        //Creamos a la productora
        Productora prod01 = new Productora(1,"Artes PERU");

        //Inicializamos los eventos de la productora
        prod01.setEventos(new ArrayList<>());

        //Creamos un concierto organizado por la productora
        Concierto concert01 = new Concierto("ENANITOS VERDES EN TRACCION
ACUSTICA",10000.00,false,true,true);

        //Inicializamos las funciones del evento
        concert01.setFunciones(new ArrayList<>());

        //Creamos locales
        Local local01 = new Local("GRAN TEATRO NACIONAL","Av. Javier Prado Este
2225",1500);
        Local local02 = new Local("TEATRO PERUANO JAPONES","Av. Gregorio Escobedo
803",1025);

        //Creamos las funciones
        Funcion funcion01 = new Funcion(sdf.parse("01-04-
2023"),LocalTime.of(20,00,00),LocalTime.of(22,00,00),local01);
        Funcion funcion02 = new Funcion(sdf.parse("02-04-
2023"),LocalTime.of(20,00,00),LocalTime.of(22,00,00),local02);

        //Asociamos las funciones al evento
        concert01.getFunciones().add(funcion01);
        concert01.getFunciones().add(funcion02);

        //Creamos el artista
        Persona per01 = new Persona("MARCIANO CANTERO",OrigenArtista.INTERNACIONAL,
"HORACIO", "CANTERO", sdf.parse("25-08-1960"), 'M');
        Persona per02 = new Persona("FELIPE STAITI",OrigenArtista.INTERNACIONAL,
"DANIEL", "STAITI", sdf.parse("29-08-1961"), 'M');
        Persona per03 = new Persona("DANIEL PICCOLO",OrigenArtista.INTERNACIONAL,
"DANIEL", "PICCOLO", sdf.parse("15-04-1959"), 'M');

        Agrupacion agrupacion01 = new Agrupacion("LOS ENANITOS
VERDES",OrigenArtista.INTERNACIONAL, 1979);
```

```

    agrupacion01.setMiembros(new ArrayList<>());
    agrupacion01.getMiembros().add(per01);
    agrupacion01.getMiembros().add(per02);
    agrupacion01.getMiembros().add(per03);

    //Asociamos el artista al evento
    concert01.setArtistas(new ArrayList<>());
    concert01.getArtistas().add(agrupacion01);

    //Asociamos el evento a la productora
    prod01.getEventos().add(concert01);

    //Creamos una obra teatral organizada por la productora
    ObraTeatral obrat01 = new ObraTeatral("AMAR LO QUE
HACES",10000.00,false,2,false);

    //Inicializamos las funciones del evento
    obrat01.setFunciones(new ArrayList<>());

    //Creamos la función
    Funcion funcion03 = new Funcion(sdf.parse("08-04-
2023"),LocalTime.of(21,00,00),LocalTime.of(23,00,00),local01);

    //Asociamos la función al evento
    obrat01.getFunciones().add(funcion03);

    //Creamos la artista
    Persona per04 = new Persona("WENDY RAMOS",OrigenArtista.NACIONAL, "JANET",
"RAMOS", sdf.parse("01-12-1966"), 'F');

    //Asociamos el artista al evento
    obrat01.setArtistas(new ArrayList<>());
    obrat01.getArtistas().add(per04);

    //Asociamos el evento a la productora
    prod01.getEventos().add(obrat01);

    //Consultamos todos los eventos de la productora
    String reporte = prod01.consultarEventos();

    //Imprimimos el reporte
    System.out.print(reporte);
}
}

```

Figura 01. Código fuente de la clase Principal

```

CONCIERTO: ENANITOS VERDES EN TRACCION ACUSTICA - ACUSTICO:true - TEC ASISTIVA:true
ARTISTAS:
- AGRUPACION: LOS ENANITOS VERDES - INTERNACIONAL - 1979
FUNCIONES:
- Fecha: 01-04-2023 - Hora Inicio:20:00 en GRAN TEATRO NACIONAL
- Fecha: 02-04-2023 - Hora Inicio:20:00 en TEATRO PERUANO JAPONES
OBRA TEATRAL: AMAR LO QUE HACES - BENEFICA:false - INMERSIVA:false
ARTISTAS:
- PERSONA: WENDY RAMOS - NACIONAL
FUNCIONES:
- Fecha: 08-04-2023 - Hora Inicio:21:00 en GRAN TEATRO NACIONAL

```

Figura 02. Salida del sistema informático

```

public String consultarDatosArtistasyFuncionesEvento(){
    SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
    String cadena = "ARTISTAS: " + "\n";
    for(Artista artista : artistas){
        cadena += "- " + artista.consultarDatos();
    }
}

```

```

    }
    cadena += "FUNCIONES: " + "\n";
    for(Function funcion : funciones){
        cadena += "- Fecha: " + sdf.format(funcion.getFecha()) + " - Hora Inicio:" +
funcion.getHoraInicio() + " en " + funcion.getLocal().getNombre() + "\n";
    }
    return cadena;
}

```

Figura 03. Código a colocar en la clase "Evento" y a reutilizar en sus clases derivadas empleando "super".

Es indispensable que se encuentren programados todos los atributos de las clases y sus relaciones (como mínimo la programación de las relaciones que permiten la impresión del reporte). Con respecto a los constructores, getters y setters, puede programar solo aquellos que son requeridos para la salida del reporte.

#### **Aspectos a considerar para evitar descuento de puntos:**

- Nombrar correctamente a las clases, atributos y métodos.
- Utilice los principios de POO: abstracción (en clases y métodos donde sea requerido), polimorfismo y encapsulamiento.
- Utilice correctamente ámbitos, clases/métodos abstractas(os), clases de tipo interface, enumerados donde sea requerido.
- Colocar a modo de comentario su nombre completo y código PUCP en la parte superior de todas las clases.
- Respetar el orden en la estructura de una clase.
- Emplear un archivo por clase.
- El programa debe compilar correctamente, se descontarán puntos por errores de compilación.

#### **Rúbrica de calificación:**

(0.5 punto) Correcta programación de la clase de tipo interface.

(0.5 punto) Correcta programación de la clase enumerate.

(9 puntos) Correcta programación de todas las clases con sus atributos, relaciones, constructos, getters y setters (respetando encapsulamiento, abstracción (métodos y clases), herencia, interfaces y enumerados).

(7 puntos) Correcta programación del método consultarDatos() en las clases donde es requerido el uso de la interface.

(1 puntos) Correcta programación del método consultarEventos() en Productora.

(2 puntos) Correcta programación de la clase Principal y visualización del reporte.

**Profesor del Curso:**

**Dr. Freddy Paz**

**30 de marzo del 2023**