

Trabajo Práctico N° 2

El presente trabajo práctico tiene como objetivo aplicar los conceptos, principios y características del paradigma de la programación orientada a objetos (Clases, objetos, métodos, interfaz, herencia, polimorfismo, colecciones, etc).

Entrega intermedia: 28 de Octubre de 2025 (diagrama de clases tentativo)

Fecha de entrega final: 18 y 25 de Noviembre de 2025

Objetivo

Implementar los conceptos abordados del paradigma de la Programación Orientada a Objetos, como solución a un problema real, partiendo del análisis de la problemática, el diseño de la solución (Diagrama de Clases) y posteriormente la construcción del producto final (Software).

Objetivos específicos

- Aplicar los conceptos fundamentales del paradigma orientado a objetos en un contexto lúdico y concreto.
- Analizar una problemática real y modelar a través de clases, relaciones y responsabilidades.
- Diseñar un sistema extensible, con bajo acoplamiento y alto nivel de cohesión.
- Trabajar con colecciones, estructuras de datos dinámicas y archivos externos.
- Realizar pruebas automáticas que verifiquen el correcto comportamiento del sistema ante diversos escenarios.

Consigna

Temática

Una discográfica quiere montar una "banda temporal" para un recital especial. Ya cuenta con algunos Artistas **base** (contratados o dentro del sello) que formarán el núcleo del proyecto. Para completar el plantel y poder tocar todas las canciones, la discográfica debe contratar artistas externos.

El desafío es modelar este problema y desarrollar software que, partiendo de la información sobre canciones, artistas y su historia, seleccione qué artistas externos contratar para poder interpretar todas las canciones, con el menor costo posible y cumpliendo las reglas del dominio.

Glosario

- **Canción:** pieza musical a interpretar en el recital. Cada canción requiere uno o más roles (por ejemplo: voz principal, guitarra eléctrica, bajo, batería, coros, teclados, etc.).
- **Artista:** músico o técnico que puede desempeñar un rol por canción, de varios roles. Cada artista tiene: nombre, lista de roles que ha ocupado históricamente, histórico de bandas/colaboraciones (lista de nombres de

bandas con las que tocó), un costo por contratación por canción, y la cantidad máxima de canciones dispuesto a tocar en un mismo recital (si es artista contratado). Un artista solo puede ser asignado a roles que haya desempeñado históricamente.

- **Costo contratación:** Cada artista candidato tiene un costo de contratación. Si algún artista base comparte historial de haber pertenecido a la misma banda histórica con el artista candidato, el costo del candidato se reduce a la mitad (50%). Si se comparte con más de un base, sigue siendo 50% (no acumulativo).
- **Recital:** conjunto de canciones que se quieren interpretar.

Descripción General

Se espera que el sistema permita responder a las siguientes funcionalidades:

1. **¿Qué roles (con cantidad) me faltan para tocar una canción X del recital?**
2. **¿Qué roles (con cantidad) me faltan para tocar todas las canciones?**

En este caso, cada canción puede incluir a los artistas base, disminuyendo la cantidad total de roles

3. Contratar artistas para una canción X del recital

Se debe optimizar el costo de contratación de artistas para una canción en particular. Al finalizar este proceso, se considera la contratación hecha para esa canción

4. Contratar artistas para todas las canciones a la vez

Para esta operación, se debe optimizar el costo de todas las contrataciones a la vez, teniendo en cuenta descuentos por compartir bandas y cantidad máxima de canciones que puede interpretar cada artista por recital. Las canciones que ya tengan contratos definidos en el punto anterior quedarán exentas

5. Entrenar artista

Se puede entrenar a un artista para que adquiera un nuevo rol, incrementando su costo un 50% por cada rol adicional que se agregue. No se puede entrenar un artista ya contratado para alguna canción, ni a un artista base

6. **Listar artistas contratados, su información relevante y su costo**
7. **Listar canciones con su estado (si tiene cubierto todos los roles, su costo, etc)**
8. **[PROLOG]**
9. **Salir**

Con todo lo que eso pueda implicar

Para los casos de contratación efectiva, si no hubiese artistas disponibles, se debe generar un error, y ofrecer si hay artistas disponibles restantes suficientes, la opción de entrenar artista

Cada operación debe mostrar la información suficiente previamente para que el usuario del sistema pueda tomar la decisión con el conocimiento necesario. También debe ser claro con los mensajes de respuesta

Requisitos técnicos

- Utilizar clases, objetos y relaciones entre ellos.
- Aplicar principios de encapsulamiento, composición, y responsabilidad única.
- Utilizar pruebas automatizadas y métodos de verificación simples que permitan testear los casos anteriores.
- Se debe poder agregar nuevos artistas y canciones al recital sin modificar el código existente (principio de abierto/cerrado).

Integración con Prolog

Realizar la integración con Prolog, para poder responder la siguiente pregunta:

¿Cuántos entrenamientos mínimos debo realizar para cubrir todos los roles para el recital, utilizando solo los miembros base, y artistas contratados sin experiencia y con un coste base por parámetro, para todos iguales?

Algunas referencias:

- <https://jpl7.org/TutorialJavaCallsProlog>
- <https://github.com/SWI-Prolog/packages-jpl/blob/a6a3617d17cde587ee0946c108d92cae8b1627d0/src/test/java/org/jpl7/standalone/Family.java>

Datos de origen

Deben utilizarse, al menos, dos archivos para ingresar la información necesaria al sistema. La misma **no debe estar codificada, sino que debe usarse dependiendo del archivo ingresado.**

1. Un archivo artistas.json (o XML, etc), que contenga todo lo necesario para tener la información para operar.
2. Un archivo recital.json (o XML, etc), que describa todas las canciones necesarias con la información necesaria, y los roles requeridos para ser interpretada
3. Un archivo artistas-incluidos.json (o XML, etc) que aclare que artistas ya se consideran contratados. La información de los mismos se debe encontrar en el archivo de artistas
4. Bonus (*) (1 punto): al salir del programa, crear un archivo recital-out.json (o XML, etc), que tenga la información final del estado del recital, con la información de contratación, y totalizadores. No necesariamente tiene que estar completo el recital, puede ser un estado intermedio. Bonus2 (*) (1 punto): Guardar esa información en un archivo con un nombre elegido y poder utilizar esta información para cargar un estado previo (nueva opción en el menú)

Pruebas

Crear una batería de pruebas para cada pregunta funcional.

Utilizar JUnit para esto, y validar que las contrataciones se ejecuten correctamente. Se espera una cobertura de pruebas adecuada para todas las funcionalidades descritas, incluyendo tanto casos exitosos como escenarios con errores o restricciones.

Bonus

Los bonus consisten en objetivosopcionales que potenciarán la nota del grupo. La nota del TP con todo lo anterior cumplido es de 8 (si todo está perfecto). A ello pueden sumar uno o varios de estos bonus para aumentar la nota (6 como máximo). Los pueden utilizar para compensar alguna otra funcionalidad que no haya quedado terminada completamente, o para llevar su nota a un 10.

1. Artista estrella invitado [descuento adicional] (2 puntos)

Cada recital tendrá un tipo, y si el mismo coincide con tipos de recitales deseados por el artista, se podrá aplicar un descuento adicional por artista estrella invitado. Este descuento solo puede ser aplicado a un Artista para todo el recital, y deberá ser tenido en cuenta en el paso de contratar todos los artistas a la vez

2. Arrepentimiento [quitar Artista del Recital] (2 puntos)

Se deberá agregar una opción adicional en el menú que permite quitar un Artista ya seleccionado, teniendo en cuenta todas las consideraciones.

3. Historial de colaboraciones visualizado (1 punto)

Mostrar un grafo simple de relaciones entre artistas según bandas compartidas o colaboraciones (ej: Bowie ↔ Queen por Under Pressure).

4. Restricciones logísticas (2 puntos)

Incluir limitaciones de disponibilidad horaria para artistas (ej: Elton solo puede tocar en la primera mitad del show), lo que afecta el algoritmo de asignación.

5. Datos de origen (1 o 2 puntos)

x. Sugerencias de equipos

Interfaz

Se evaluará el correcto funcionamiento del sistema utilizando dos enfoques simultáneos:

1. Se deberá probar cada una de las funcionalidades desarrolladas, utilizando JUnit correctamente
2. Se deberá mostrar la ejecución con un main, en el cual puedan ser mostradas las opciones del menú principal

Sugerimos tener varios escenarios preparados, para mostrar todas las características desarrolladas.

Ejemplos de Archivos

Se proporcionan ejemplos no exhaustivos para dar una idea del armado de los archivos. Estos archivos **no abarcan** todos los casos planteados en el enunciado, pero son un punto de partida. Se sugiere crear los propios. No es necesario respetar el formato sugerido.

artistas.json

```
[  
 {  
   "nombre": "Brian May",  
   "roles": ["guitarra eléctrica", "voz secundaria"],  
 }
```

```
        "bandas": ["Queen"],
        "costo": 0,
        "maxCanciones": 100
    },
    {
        "nombre": "Roger Taylor",
        "roles": ["batería", "voz secundaria"],
        "bandas": ["Queen"],
        "costo": 0,
        "maxCanciones": 100
    },
    {
        "nombre": "John Deacon",
        "roles": ["bajo"],
        "bandas": ["Queen"],
        "costo": 0,
        "maxCanciones": 100
    },
    {
        "nombre": "George Michael",
        "roles": ["voz principal"],
        "bandas": ["Wham!", "George Michael"],
        "costo": 1000,
        "maxCanciones": 3
    },
    {
        "nombre": "Elton John",
        "roles": ["voz principal", "piano"],
        "bandas": ["Elton John Band"],
        "costo": 1200,
        "maxCanciones": 2
    },
    {
        "nombre": "David Bowie",
        "roles": ["voz principal"],
        "bandas": ["Tin Machine", "David Bowie"],
        "costo": 1500,
        "maxCanciones": 2
    },
    {
        "nombre": "Annie Lennox",
        "roles": ["voz principal"],
        "bandas": ["Eurythmics"],
        "costo": 900,
        "maxCanciones": 2
    },
    {
        "nombre": "Lisa Stansfield",
        "roles": ["voz principal"],
        "bandas": ["Lisa Stansfield"],
        "costo": 800,
        "maxCanciones": 2
    }
]
```

recital.json

```
[  
  {  
    "titulo": "Somebody to Love",  
    "rolesRequeridos": ["voz principal", "guitarra eléctrica", "bajo",  
"batería", "piano"]  
  },  
  {  
    "titulo": "We Will Rock You",  
    "rolesRequeridos": ["voz principal", "guitarra eléctrica", "bajo",  
"batería"]  
  },  
  {  
    "titulo": "These Are the Days of Our Lives",  
    "rolesRequeridos": ["voz principal", "guitarra eléctrica", "bajo",  
"batería"]  
  },  
  {  
    "titulo": "Under Pressure",  
    "rolesRequeridos": ["voz principal", "voz principal", "guitarra  
eléctrica", "bajo", "batería"]  
  }  
]
```

artistas-discografica.json

```
[  
  "Brian May",  
  "Roger Taylor",  
  "John Deacon"  
]
```

Entrega

El trabajo se realizará en forma grupal, con equipos de 4 a 6 personas.

Se espera un informe que contenga al menos: carátula, índice, introducción, desarrollo, conclusiones y referencias (**APA**).

Habrá una defensa y presentación oral en la fecha designada, y una serie de rúbricas se aplicarán oportunamente.