

Defensa Práctica ASI (I)

Arquitectura de los Sistemas de Información

Alejandro Cortiñas – alejandro.cortinas@udc.es

16 de octubre de 2025

1 Instrucciones

Para realizar el ejercicio se debe **clonar en local** el repositorio `https://gitlab.com/acortinas/2526-asi-defensa-p1`, que contiene tanto el código del lado cliente, *client*, como del lado servidor, *server*. Las instrucciones de ejecución de cada proyecto son las mismas que las de ejecución del esqueleto de la práctica. Recordar configurar los ficheros *application.yml* y *constants.js* de ser necesario¹.

Todas las funcionalidades se deben implementar modificando únicamente el lado cliente.

¹Esto será necesario si los puertos 8080 y 1234 están ocupados por otros procesos, o si se quiere configurar la base de datos de PostgreSQL en vez de usar H2 en memoria.

Entrega.

- Se debe entregar la carpeta del repositorio comprimida en formato zip.
- **No se debe hacer ningún commit con los cambios realizados durante la defensa.**
- Todo el código modificado que no aporte a la implementación de las HU solicitadas se valorará negativamente.
- Antes de comprimir el repositorio, eliminad la carpeta *client/node_modules* y cualquier carpeta de clases compiladas en *server (bin, target, build)*.
- Para estar seguros de subir correctamente el código podéis comprobar que al ejecutar el comando `git diff` se ven los cambios realizados durante la defensa.

Captura de pantalla.

- Se debe capturar la pantalla durante toda la duración de la prueba, desde el inicio hasta que hayáis subido el código a la tarea del campus virtual.
- El fichero con el vídeo capturado se compartirá al finalizar la prueba utilizando OneDrive. Se ha creado una tarea específica para subir el enlace compartido de OneDrive.
- **Justo antes de empezar a grabar debéis mostrar el nombre del alumno/a en grande, ocupando toda la pantalla (con el bloc de notas o un word con tamaño de letra enorme)². A continuación, se debe mostrar la lista de extensiones o plugins de cada IDE que vayáis a utilizar.**

La tarea de la entrega del código estará abierta en el Campus Virtual hasta la hora de entrega establecida durante la defensa. La tarea de la entrega del vídeo estará abierta hasta 2 horas después del límite de la defensa.

²Es decir, el primer fotograma del vídeo debería ser vuestro nombre en grande.

2 Descripción

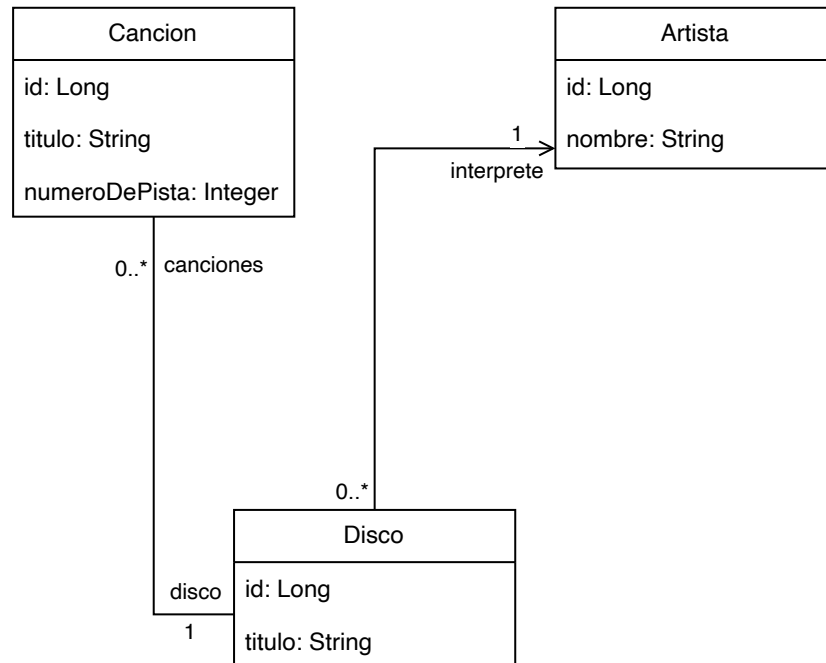


Figura 1

El modelo de datos del proyecto se puede ver en la figura 1. Estamos desarrollando una aplicación para gestionar nuestra mediateca musical personal, guardando datos como los artistas, discos, canciones y géneros.

El API REST proporciona los siguientes endpoints:

- GET /discos - Devuelve la lista de discos disponibles.
- GET /discos/{idDisco} - Devuelve la información de un disco concreto en función del identificador indicado.
- PUT /discos/{idDisco}/canciones - Añade una nueva canción al disco indicado. El formato del cuerpo de la petición debe ser:

```
{  
  "cancion": "El titulo de la cancion"  
}
```

- GET /artistas - Devuelve la lista de artistas disponibles.
- GET /artistas/{idArtista} - Devuelve la información de un artista en concreto en función del identificador indicado.
- GET /artistas/{idArtista}/discos - Devuelve los discos disponibles del artista indicado (mediante el identificador).

3 Funcionalidades a implementar

Todas las funcionalidades son públicas, es decir, accesibles por un usuario sin autenticarse.

HU1 Listado de discos (0.5). Accediendo a la ruta `http://localhost:1234/discos` se mostrará la lista de discos. Para cada disco se mostrará el título, que será a su vez un enlace para acceder a **HU4**, y el nombre del intérprete.

HU2 Enlace de menú (0.1). Se creará un enlace en el menú de la aplicación para acceder al listado de discos (**HU1**).

HU3 Número de canciones en el listado de discos (0.4). Se modificará el listado de discos (**HU1**), mostrando para cada disco el número de canciones que tiene.

HU4 Detalle de un disco (0.4). Accediendo a una ruta de la forma `http://localhost:1234/discos/_IdDelDisco_` (por ejemplo, `http://localhost:1234/discos/8472`) se mostrará la información del disco en cuestión. Se mostrará el título del disco, el nombre del intérprete del mismo, y la lista de canciones.

HU5 Lista de canciones ordenadas (0.4). La lista de canciones de un disco se mostrará ordenada por el número de pista de cada canción en la vista en detalle del mismo (**HU4**).

HU6 Detalle de un artista (0.3). Accediendo a una ruta de la forma `http://localhost:1234/artistas/_IdDelArtista_` (por ejemplo, `http://localhost:1234/artistas/550`) se mostrará la información del artista en cuestión. Se mostrará únicamente el nombre del artista. Además, se modificará el listado de discos (**HU1**), haciendo que el nombre del intérprete sea un enlace a la vista detalle del mismo.

HU7 Discos en detalle de artista (0.4). Modificamos la vista detalle de un artista (**HU6**) mostrando también la lista de discos de dicho artista. Únicamente mostramos el título de los discos.

HU8 Añadir canción a disco (0.5). En la vista detalle de un disco (**HU4**) se mostrará un pequeño formulario con un único campo y un botón para añadir una nueva canción al disco en cuestión. Una vez añadida la canción, los datos de la vista se deben recargar para mostrar también la nueva canción.

4 Imágenes de referencia



Figura 2: Ejemplo donde se listan los discos

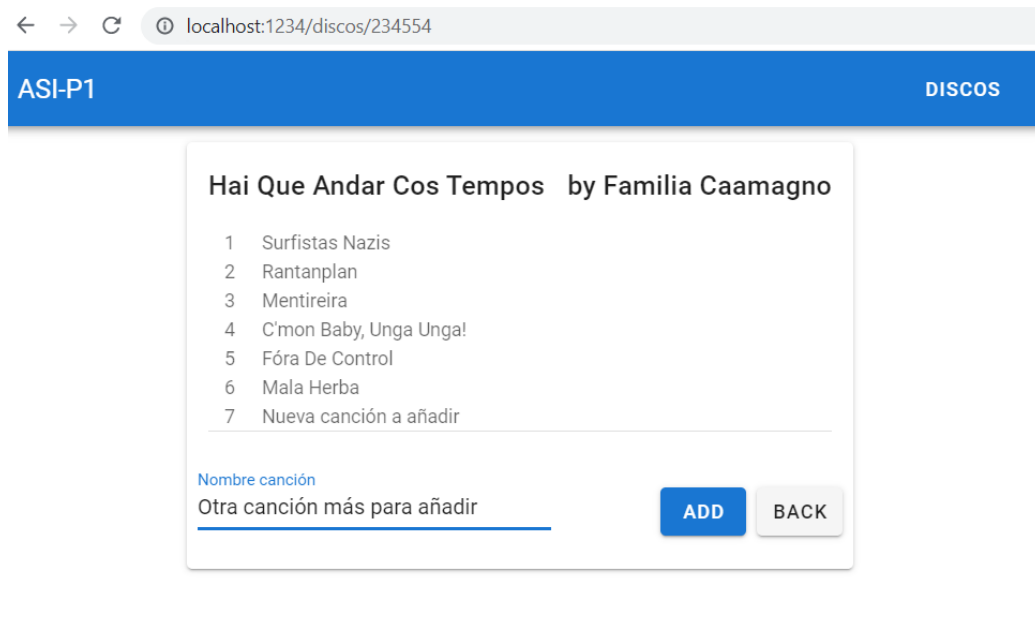


Figura 3: Ejemplo donde se muestran la vista en detalle de un disco

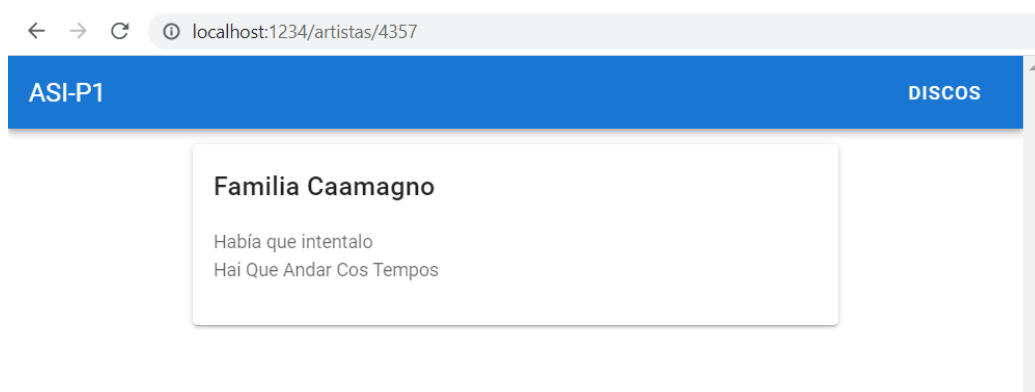


Figura 4: Ejemplo donde se muestra la vista en detalle de un artista