

Tarea 2

Bases de Datos

“Mr. Burns Needs your help”

Marco Espinoza marco.espinozaco@usm.cl	Juan Mira juan.mirag@usm.cl
Javier Peralta javier.peraltae@usm.cl	Rodrigo Munita rodrigo.rojasmu@sansano.usm.cl
Ricardo Lorca ricardo.lorca@sansano.usm.cl	Cecilia Reyes cecilia.reyes@usm.cl
Sebastián Gallardo sebastian.gallardod@usm.cl	

4 de Mayo, 2022

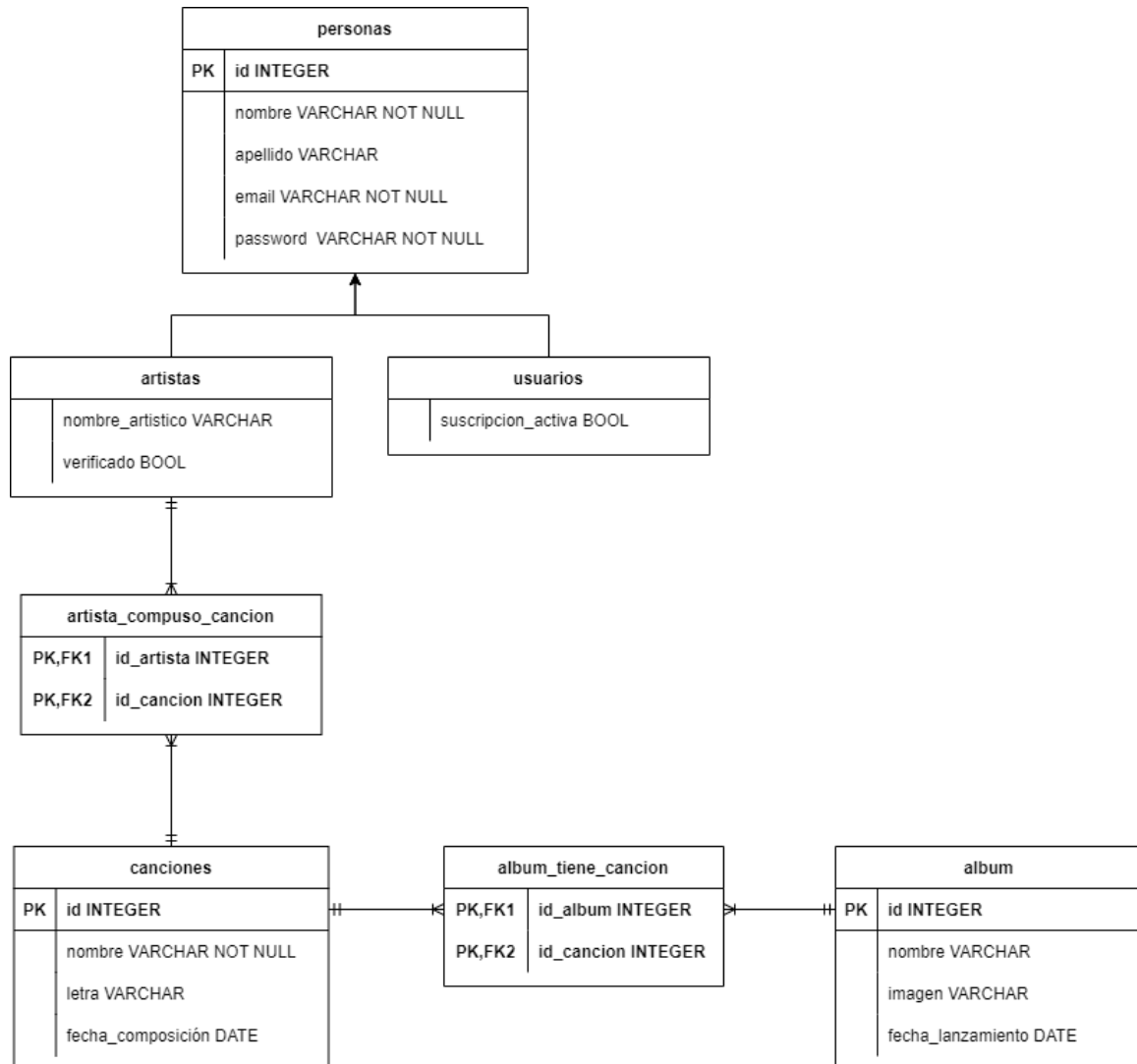


1. Introducción

Luego de su incursión en el mundo cinematográfico para el festival de cine celebrado en Springfield, el señor Burns decidió adentrarse en el mundo de la música, para esto, decide buscar a los y las mejores ingenier@s con experiencia en plataformas de streaming de música como Spopyfy, AwesomeMusic, YouSong y PearMusic, desafortunadamente no estaban disponibles para el trabajo, pero no todo está perdido, ya que el señor Burns se enteró de que en la Universidad Técnica Federico

Santa María hay ingenieros e ingenieras con el nivel necesario para el trabajo, por lo que les pide a ustedes que lo ayuden.

Para facilitarles el trabajo, el señor Burns les adjunta el siguiente modelo entidad relación:



2. Instrucciones de Entrega

Para esta entrega, su objetivo será diseñar y generar un sitio web que sirva como plataforma base para trabajar con los artistas musicales. La **plantilla** adjunta les será de utilidad como base para la tarea.

2.1. Modelo de Datos

Antes del desarrollo en sí, es necesario atender el siguiente requisito:

- Desde el modelo entregado, deben adaptar la entidad “persona” que es representada a través de herencia en el modelo a una tabla relacional, es decir, que de alguna forma su base de datos se pueda diferenciar entre una persona que es **Artista Musical** y uno que no lo es.
- Por lo tanto, deberán analizar el modelo e implementar los cambios que sean pertinentes para satisfacer el requisito.
- Existen múltiples formas válidas de implementar el requisito, por lo que deberán **justificar su elección, señalando ventajas y desventajas**.

2.2. Requisitos del Sistema

2.2.1. Sesiones

Una gran cualidad de la plataforma es que debe sostener el uso de **sesiones**, es decir, debe permitir que un Usuario ingrese sus datos, mantenga una sesión y pueda salir cuando así lo desee. Específicamente se necesita que:

- Un usuario pueda Iniciar Sesión, ingresando sus datos correo (email) y contraseña (password) en una interfaz común entregada por la plataforma.
- Cuando un Usuario ingrese sesión, el sitio debe determinar autónomamente si el Usuario es Artista Musical o **no**.
- Dentro de su sesión, el Usuario, independiente de su nivel de acceso (Artista Musical o no), debe tener una pestaña o interfaz que le permita ver su perfil (información de su cuenta).
- Además, debe ser capaz de cerrar sesión desde cualquier vista de la plataforma.

2.2.2. Interfaces

Como núcleo de funcionamiento, el sistema debe soportar interfaces y/o formularios que cumplan:

- Una **Página Principal**, comúnmente conocida como *index* o *frontpage*, la cual sirva como una bienvenida o “Primera impresión” de la plataforma, donde un Usuario que no la conoce pueda entender en qué consiste, sus desarrolladores, qué servicios entrega, etc.
 - Si no se especifica una URL particular de la plataforma (sólo la IP o dominio), esta debe ser la vista a la que llegue el Usuario.
 - Si se intenta entrar a una página de forma indebida (sin tener una sesión válida o teniendo los privilegios incorrectos), se debe redirigir al usuario a la página principal.

- Una interfaz de **Sign In/Sign Up** (Registro e inicio de sesión); donde un Usuario podrá iniciar sesión, entregando su correo y contraseña, o un visitante podrá registrarse en la página web, ingresando todos los datos necesarios para ello.
 - Se debe poder acceder a estas vistas sólomente si no se ha iniciado sesión. En caso contrario, debe redirigirse a la página principal.
 - Cuando un persona desea registrarse, se le asignará por defecto el rol **no artista**.
- Un **CRUD** para las canciones, donde se puedan crear nuevas canciones, ver todas las canciones registradas, modificar sus atributos o borrarlas.
 - Esta vista sólo puede ser accedida por un **Artista Musical**.
 - Al crear una canción, debe permitirse al usuario ingresar el id del álbum al que va a pertenece dicha canción.
- Un **CRUD** para los Álbumes, donde se puedan crear nuevos álbumes, ver todos los álbumes, modificar sus atributos o borrarlos.
 - Esta vista sólo puede ser accedida por un **Artista Musical**.
- Una página que contenga una lista de todas las canciones, cada fila de la lista debe contener: el nombre de la canción, los artistas musicales que la compusieron, el álbum al que pertenece y un botón que permita acceder al perfil de la canción.
 - Esta vista sólo puede ser accedida por un **Usuario no artista**.
- Un perfil de canción, en donde se debe listar el nombre, artistas, fecha de composición, álbumes a los que pertenece la canción y letra.
 - Esta vista sólo puede ser accedida por un **Usuario no artista**.
- Una interfaz de **Perfil**, la cual muestra los Datos personales al Usuario.
 - Cualquier Usuario (artista o no) puede acceder a esta vista, siempre y cuando ya este registrado.

3. Condiciones de Entrega

3.1. Informe

Dentro del mundo del desarrollo, nada está 100 % estandarizado, y las ideas e implementaciones varían de desarrollador en desarrollador. Es por esto que, independiente de cómo o qué implementen en su entrega, les solicitamos que adjunten a esta un **Informe**, el cual detallará todas sus consideraciones, suposiciones, detalles de uso y notas respecto a la plataforma. Como mínimo se solicita que el informe:

- Debe incluir los **datos personales de cada alumno** de su grupo, junto con su número de ROL USM.
- *(Opcional) Si lo desean, pueden adjudicarse un “Nombre de Grupo” y utilizarlo como marca o empresa desarrolladora de la página.*
- Debe incluir **todos los supuestos y consideraciones** que uds. tomaron en cuenta al desarrollar la entrega. Además, si se agregó algún archivo que no venía en la plantilla, debe indicarse qué se agregó, dónde y con qué propósito (Por ejemplo: *Se agregó foto.png a la carpeta img para mostrarse en la página principal*).
- Debe señalar **cómo se manejó el Modelo** para permitir sesiones de Artista Musical y Usuario, mostrando al menos una ventaja y desventaja de la forma en que lo plantearon.
- Debe indicar, si es que los hay, qué archivos de la plantilla fueron modificados. Esto **no considera** archivos tanto `.html` y `.php` con comentarios indicando que hagan cambios.
 - Expliciten también si las modificaciones fueron de funcionalidad o simplemente estéticas (ejemplo: uso de Bootstrap para hacerlas más estéticas).
- Debe incluir un enlace al repositorio GIT de la tarea. El repositorio debe ser privado, es por ello que deben dar permisos a los ayudantes.
- En una sección final, solicitamos que informen acerca de sus **dificultades** con esta implementación; en concreto, que puedan reportar cuáles tareas fueron las más difíciles (o largas) en su desarrollo. Puede variar desde instalación de tecnologías, ajuste de modelo, proceso creativo, desarrollo, etc.
- En esta misma última sección, solicitamos que reporten un estimado de cuánto **tiempo** les tomó el desarrollo de la tarea completa (valores por persona), buscando incluir todo momento que gastaron en ella.
 - *Estimadores Recomendados: Análisis del Enunciado, Modificaciones y ajustes al Modelo, Diseño de plataforma, Desarrollo de Plataforma, Pruebas Finales.*

El informe debe venir en formato **PDF**, e incluir las secciones detalladas anteriormente.

Nombre	Github
Marco	mavi3
Juan	juan-mira
Javier	Lord-Ureiva
Rodrigo	rgomunita
Ricardo	rilodeh

Cuadro 1: Usuarios de Github

3.2. Implementación

Respecto a la implementación de los requisitos, se solicita:

- Utilizar la **Plantilla** entregada como base de su tarea.
- Se subió un vídeo a AULA en donde están los **ajustes necesarios** para la correcta ejecución de la tarea.
- El código debe estar **ordenado**, es decir, indentado, estructurado y lo más comentado posible.
- **Importante:** Se debe hacer un buen uso de SQL, sin buscar que la lógica del sitio ejecute operaciones costosas que debieran ser resueltas por las consultas a la base de datos. Por ejemplo, se considerará un **mal uso** el que, en vez de utilizar un JOIN, iteren sobre una consulta con un *for* para hacer una consulta sobre cada elemento de la primera (estilo pythoniano).
 - Nota: Esto no implica que no deban utilizar ciclos (como *for*) en su código, nos referimos particularmente a reemplazar las consultas por procesamiento de la plataforma.
- Se debe mantener la estructura de los archivos lo más ordenado posible. En caso de agregar archivos nuevos a la plantilla, deben seguir la misma línea que los ya existentes.
- *Bonus 5pts: Utilizar Bootstrap u otra librería de desarrollo para que las vistas sean estéticas y consistentes.*
- **Chiste:** Pueden hacer una seccion de chiste en donde poner algun meme, chiste, video o algo que ponga de buen humor al ayudante a la hora de revisar.

4. Condiciones Generales

- La fecha de entrega es el **Lunes 6 de Junio del 2022 a las 23:59PM**.
 - Se aplicará un **descuento** lineal para tareas atrasadas hasta las **23:59 PM** del día siguiente.
- Las tecnologías a utilizar para el desarrollo de la plataforma son:
 - **Apache 2.4:** Montar plataforma con sus vistas (*Front-End*)
 - **PHP 8.1+:** Para manejar la lógica del negocio, es decir, el código detrás de la página
 - **PostgreSQL 14:** Para el manejo de la base de datos asociada al sistema (*Back-End*)

- Su entrega debe incluir el **informe** (nombre de archivo **InformeGrupoX**) y un **dump** de su base de datos modificada (nombre de dump **DumpGrupoX**). Todo esto en un comprimible de nombre **Tarea2-GrupoX.zip**, donde X es el número de su grupo.
- Es muy importante incluir el enlace al repositorio GIT en el informe, puesto que se evaluarán los commits realizados por cada integrante del grupo.
- Integrante que no es incluido en el informe y/o no participa en el desarrollo de la tarea tendrá **Nota 0**.
- La entrega de los archivos es por el medio Aula, sólo uno de los integrantes del grupo debe realizar la entrega.

5. Recomendaciones

- Aún en un gran número de tecnologías usadas en el oficio de desarrollo de software, es común que la instalación de tecnologías sea un paso difícil de realizar. Sin embargo, les recomendamos que, ante cualquier fallo o impedimento, **busquen en internet si alguien ha tenido el mismo problema**. Muchas veces horas de peleas se ahorran con un milagroso post en [Stackoverflow](#) del 2014.
- Esta tarea requiere en parte, el uso de creatividad y diseño por parte de ustedes. Sin embargo, esta es una excelente oportunidad para crear su propia página web y decorarla como a uds grupo les parezca mejor, implicando paletas de colores, organización de interfaces y/o contenido visual para el Usuario, temáticas, etc.
- A pesar de que no es necesario, Bootstrap es una excelente herramienta para quien quiere añadir estética a su página. Es sencillo de instalar y mucho más fácil de usar que otras tecnologías (y les servirá mucho para ramos posteriores).
- Parte del trabajo es programar en PHP, lo que implica nuevos entornos y comandos a utilizar. Ante cualquier duda o incógnita que tengan sobre el lenguaje, les recomendamos con mucha fuerza revisar W3Schools. Es la biblia del informático (e incluye un apartado de Bootstrap!).
- Finalmente, consejo general de desarrollador: Si quieres saber cómo hacer X con Y tecnología, googlea “How to do X with Y” (o en español si les complica el Inglés) y revisa los resultados.