Desarrollo de Software 2024 (1er. Cuat)

Parcial 3K2 - Miércoles

✓ **Hecho:** Ver ✓ **Hecho:** Hacer un envío

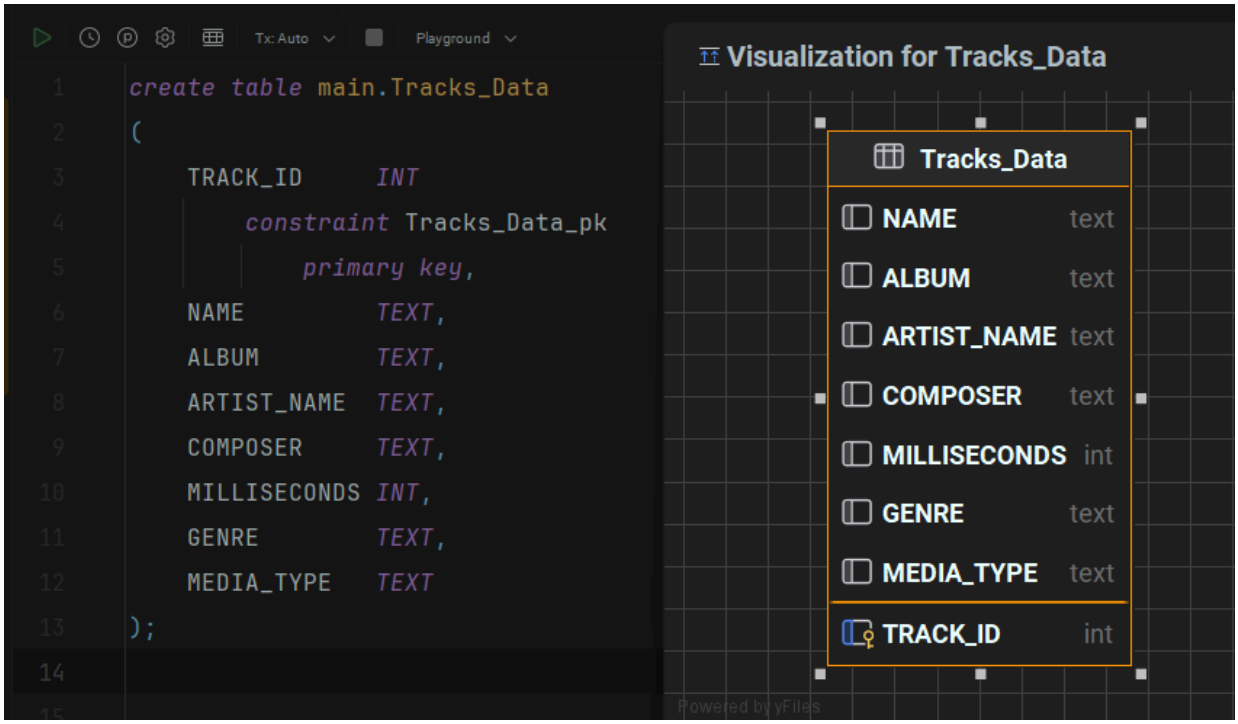
Apertura: miércoles, 19 de junio de 2024, 10:45
Cierre: miércoles, 19 de junio de 2024, 13:00

SISTEMAS
UTN - FRC

DESARROLLO DE SOFTWARE
Evaluación Sumativa

Introducción

Continuando con el dominio de problema del primer parcial, usted dispone del acceso a la siguiente tabla de la base de datos, en ella se encuentra la lista de canciones para las que la Radio tiene contratados los derechos y a partir de ella se solicita que se construya una aplicación de Frontend usando React para la visualización filtrado y creación de canciones en la mencionada tabla a través de una API Rest que ya está desarrollada y se adjunta a las presentes consignas, para que los musicalizadores puedan tomar canciones para musicalizar los programas de la emisora. La estructura de la tabla es la siguiente:



En este escenario, a usted le toca programar la pantalla del listado de canciones que deberá incluir la posibilidad de hacer filtros, y también el formulario para agregar una nueva canción a la base de datos. Para esto usted va a interactuar con la API Rest ya desarrollada, el código de dicho proyecto está adjuntado a esta tarea y lo puede descargar al final de las presentes consignas.

La mencionada API Rest es un proyecto node + express + sequelize como vimos en clase, al iniciar escucha de acuerdo con la configuración provista en el puerto 3001 y expone los siguientes endpoints:

1. GET: **/api/status**: simplemente realiza un health check del servidor.

2. GET: **/api/tracks**: retorna una lista de objetos JSON con las primeras 15 canciones de la tabla ordenadas por álbum y nombre.
3. GET: **/api/tracks?filterText=texto-para-filtrar&genre=genero-para-filtrar**: es el mismo endpoint anterior pero utiliza el o los parámetro/s suministrado/s para filtrar aquellas canciones que incluyan el valor del filtro de texto entre el nombre, álbum, artista o compositor y/o que sean estrictamente del género indicado. Cabe aclarar que se pueden enviar ambos parámetros para filtrar o solo uno de ellos pero si se envían deben enviarse con un valor asociado.
4. POST: **/api/tracks**: agrega una canción en base al JSON suministrado en el body, la estructura esperada en dicho objeto es:

```
{
   "name": "Bandidos Rurales",
   "album": "Bandidos Rurales",
   "artistName": "Leon Gieco",
   "composer": "Hugo Chumbita, Luis Gurevich, Leon Gieco",
   "milliseconds": 376648,
   "genre": "Rock",
   "mediaType": "Protected AAC audio file"
 }
```

   , la creación la realiza si y solo si el nombre de la canción y el álbum son únicos en la base de datos y la duración es mayor a 10 segundos. En caso de éxito responde indicando la creación con un objeto JSON que incluye además el id, en caso de repetición responde indicando que el nombre ya existe o el error que corresponda.
5. PATCH: **/api/tracks/{id}**: actualiza la canción con el id proporcionado, tener en cuenta que solo se pueden actualizar los atributos: artistName, composer, milliseconds y genre. En caso de no encontrar la canción retorna canción no encontrada y en caso de una actualización exitosa retorna el objeto actualizado.
6. GET: **/api/genres**: retorna una lista de strings con todos los géneros existentes en alguna canción de la tabla ordenados alfabéticamente.

Para tener funcionando este proyecto y por tanto el servidor de backend debe descargar el archivo, descomprimir el archivo en el directorio de su repositorio y instalar las dependencias y ejecutar con:

```
$: npm run dev
```

Para esta evaluación usted deberá actuar como un desarrollador frontend programando el proyecto del frontend utilizando React.

Consigna

Despliegue del Backend

Deberá descargar el archivo que contiene los fuentes de la mencionada API Rest con el backend de la aplicación y luego:

1. Descomprimir el archivo comprimido en el directorio clonado de su repositorio, esto creará un directorio llamado backend (borre el archivo comprimido una vez que lo haya descomprimido).
2. Instalar las dependencias, debe hacerlo dentro del directorio backend.
3. Iniciar el proyecto para que quede escuchando peticiones (puede validar los endpoints mencionados usando bruno u otro cliente rest o el mismo browser).

Desarrollo del Frontend

Debe desarrollar un nuevo proyecto dentro del directorio del repositorio clonado para el parcial llamado frontend:

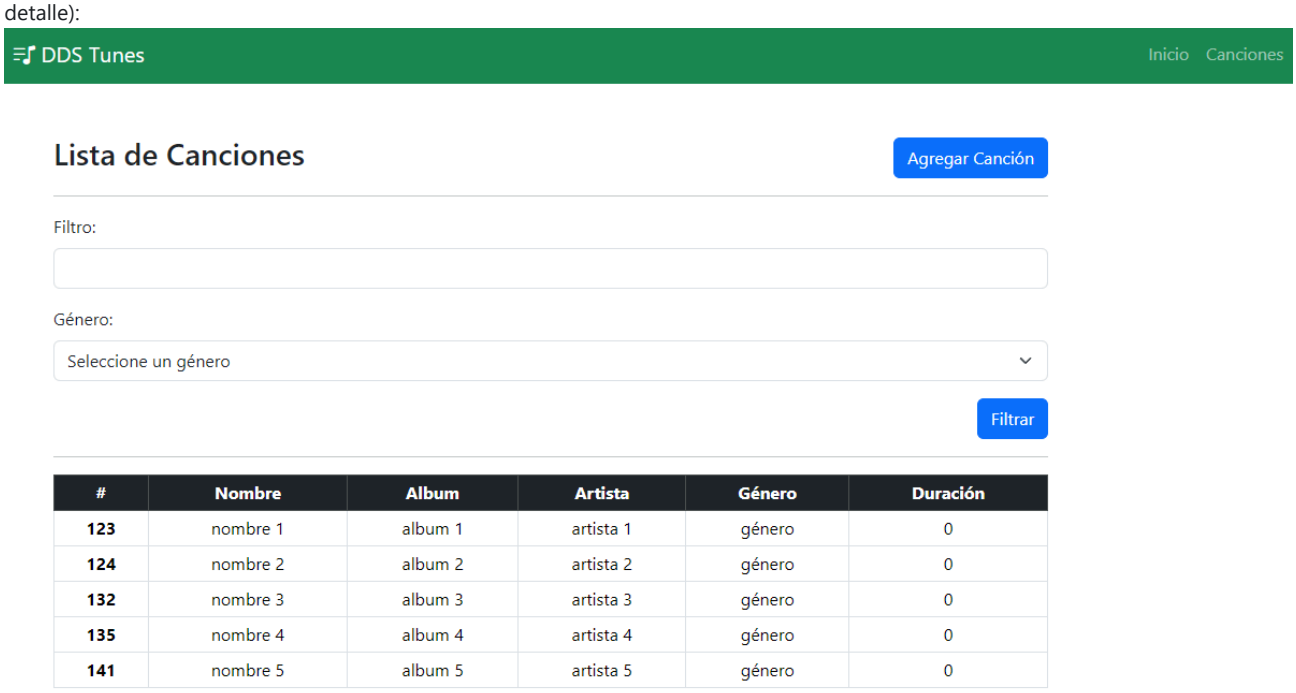
1. En el directorio frontend deberá crear un proyecto React agregando las dependencias que crea necesarias.
2. Construir los elementos principales de la aplicación (se recomienda ejecutar la aplicación y verificar que haya quedado funcionando).

A partir de aquí deberá cumplir con los siguiente requerimientos acerca de lo que la aplicación debe contener como mínimo:

1. Un componente principal llamado Menu.jsx que actúe como menú principal de la aplicación y gestione las rutas utilizando React Router. Este componente deberá definir una ruta llamada /nueva-cancion que haga referencia al componente encargado de agregar una nueva canción a la base de datos y otra ruta llamada /lista-canciones que haga referencia al componente encargado de mostrar la lista de canciones con sus filtros.
2. Deberá desarrollar el componente para Agregar una canción con el formulario para el ingreso de los datos de la canción de acuerdo con lo solicitado en el POST del backend de canciones.
    1. Los campos a solicitar para reiterar el tema son: nombre (texto), álbum (texto, nombre del artista (texto), compositor/es (texto), duración en millisegundos (número entero), género (lista de selección en base al endpoint GET: /api/genres) y tipo de codificación (texto).
        1. **Nota**: debe cargar la lista de géneros con los géneros que se obtienen al invocar el backend al iniciar el componente de creación de una nueva canción y el género de la nueva canción debe ser si o sí uno de los géneros existentes.
    2. Una vez recuperados los datos componer el objeto según el modelo propuesto en la definición del backend y enviarlo al backend para ser persistido en la tabla de la base de datos.
    3. Si la creación es exitosa debe navegar nuevamente a la Lista de canciones

4. Si hay un problema debe mantenerse en la pantalla de Agregar canción hasta navegar manualmente a la lista o resolver el problema y crear la canción.

3. Deberá crear un componente para mostrar el listado de canciones con la capacidad de filtrar el mismo, dicho componente deberá verse como sigue (siempre comprendiendo que la imagen es una propuesta y no estrictamente un requerimiento a respetar en detalle):



Nota: La duración de las canciones en la base de datos está expresada en Milisegundos, sin embargo, **se solicita que la pantalla muestre dicha duración en el formato [horas]:[Minutos]:[Segundos]**. Usted va a recibir los milisegundos desde el backend y debe hacer la conversión en el frontend para que la pantalla muestre el valor de acuerdo con lo solicitado.
Nota 2: La lista de géneros a mostrar en la lista desplegable debe ser renderizada al cargar el componente en base al listado obtenido desde el endpoint GET: /api/genres creado para tal fin.

1. Al hacer click en el botón "Agregar Canción" se debe navegar al componente encargado de Agregar una canción nueva.
2. Al iniciar la pantalla debe cargar las 15 canciones que retorna el endpoint /api/tracks sin parámetros
3. En cualquier momento el usuario puede ingresar un texto en el Filtro y/o seleccionar un género de la lista y al hacer click en Filtrar se debe mostrar el resultado del endpoint /api/tracks?filter="agregando aquí el texto ingresado"&genre="agregando aquí el value de la opción del select con el género seleccionado", el usuario puede optar por usar ambos filtros o uno de ellos pero al presionar filtrar se debe controlar que al menos haya cargado uno de los dos.

Instrucciones

1. **Clonar el repositorio** ubicado en: https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/parciales2024/parcial2/3k2/##### (donde los # representan su número de legajo).
2. Dentro de dicho repositorio descargar el backend suministrado
    1. Descargar backend de aquí.
    2. Iniciar el proyecto de Backend de acuerdo con lo indicado en el apartado específico.
3. Construir las consignas especificadas previamente acerca del frontend en un directorio dentro del mismo repositorio.
4. **Una vez realizado el parcial deberá**:
    1. **Hacer push** de todo lo realizado en su repositorio de git.
    2. **Subir el código** del frontend en la presente tarea para lo cual deberá:
        1. Borrar la carpeta node_modules
        2. Comprimir el directorio frontend.
        3. Subir el archivo comprimido a la presente taréa de aula virtual.
5. Las presentes consignas son requerimiento de la evaluación del parcial y el mismo no será corregido si no se cumplen estos requisitos.

Evaluación

La corrección se llevará a cabo de acuerdo con la siguiente tabla de criterios:.

Criterios de Evaluación

Consigna	Puntaje

Componente Menú	10
Componente Registro y Validación	40
Componente Consulta y Filtro	20
Navegación general de la aplicación	10
Diseño y comunicación entre componentes	10
Integración correcta con backend	10

Se adjunta la tabla de calificaciones que propone el departamento de sistemas y que adoptamos también en la cátedra de Desarrollo de Software.

Escala de notas

NOTA	PORCENTAJE	CALIFICACIÓN
1		No Aprobado
2		No Aprobado
3		No Aprobado
4	55% a 57%	Aprobado
5	58% a 59%	Aprobado
6	60% a 68%	Aprobado
7	69% a 77%	Aprobado
8	78% a 86%	Aprobado
9	87% a 95%	Aprobado
10	96% a 100%	Aprobado

backend.zip 19 de junio de 2024, 00:39

Estado de la entrega

Estado de la entrega	Enviado para calificar
Estado de la calificación	Publicada
Tiempo restante	La tarea fue enviada 14 minutos 31 segundos antes
Última modificación	miércoles, 19 de junio de 2024, 12:45
Archivos enviados	front.rar 19 de junio de 2024, 12:45 Exportar al portafolios
Comentarios de la entrega	Comentarios (0)

Comentario

Calificación	3,00 / 10,00
Calificado sobre	martes, 25 de junio de 2024, 10:58
Calificado por	 Ninfa Milagros Zea Cárdenas

Comentarios de retroalimentación

+

...

Consigna	Puntaje
Componente Menú	10
Componente Registro y Validación No funciona. Falta get generos.	

◄ Parcial 3K2 - Martes

Ir a...

Contactar con el soporte del sitio