

Guía para Ejecutar la Aplicación De Prueba Técnica en Java con NetBeans

Creación de API

Introducción

Esta guía proporciona instrucciones paso a paso para ejecutar la aplicación desarrollada en Java utilizando NetBeans. Asegúrate de seguir estas instrucciones cuidadosamente para garantizar una ejecución exitosa.

Requisitos Previos

Antes de comenzar, asegúrate de que tu sistema cumpla con los siguientes requisitos:

Java JDK: Debe estar instalado en tu sistema. Puedes descargarlo desde <https://www.oracle.com/java/technologies/javase-downloads.html>

NetBeans IDE: Debe estar instalado en tu sistema. Puedes descargarlo desde <https://netbeans.apache.org/download/nb122/nb122.html>

Git: Debe estar instalado en tu sistema. Puedes descargarlo desde <https://git-scm.com/downloads>

Proyecto de la prueba técnica: enlace de descarga

Proyecto BackEnd

<https://github.com/Jomeva12/crearApiListas.git>

Proyecto frontEnd

<https://github.com/Jomeva12/front-listas-de-reproduccion.git>

Cliente Postman

https://api.postman.com/collections/16188989-1410acdc-d630-49b3-9eac-c41ef1b2f9c7?access_key=PMAT-01HC77VQ3DB2JA1A1M5D6SRAXR

Acceso a base de datos h2

<http://localhost:8080/h2-console/>

Problema 3

3. Implemente en su IDE Favorito y usando las bibliotecas Java de su elección un conjunto de API cuyo contrato se detalla a continuación:

HTTP	Plantilla URI	Descripción
POST	/lists	Añadir una nueva lista de reproducción. Si se añade satisfactoriamente, devuelve "201 Created" con la referencia a la URI y el contenido de la lista. Si el nombre de la lista no es válido (ej: null) debe devolver un error "400 Bad Request".
GET	/lists	Ver todas las listas de reproducción existentes
	/lists/{listName}	Ver descripción de una lista de reproducción seleccionada. Si la lista no existe, debe devolver un "404 Not Found".
DELETE	/lists/{listName}	Borrar una lista de reproducción. Si se realiza correctamente, debe devolver un "204 No Content". Si la lista no existe debe devolver un "404 Not Found".

Para la implementación de la API, tener en cuenta:

- La representación JSON de los recursos del servicio es:

Lista de Reproducción: Una lista de reproducción tiene nombre, descripción y un conjunto de canciones.

```
{
  "nombre": "Lista 1",

  "descripcion": "Lista de canciones de Spotify",
  "canciones":
  [
    {
      "titulo": "",
      "artista": "",
      "album": "",
      "anno": ""
    },
    {
      "titulo": "",
      "artista": "",
      "album": "",
      "anno": ""
    },
    ...
  ]
}
```

Canción: Cada canción tiene un título, nombre del artista, álbum y año.

```
{
  "titulo": "",
  "artista": "",
  "album": "",
  "anno": ""
}
```

- Implementarle autenticación y autorización
- Implemente una capa de persistencia en una base de datos runtime (Sugerencia h2).
- Probar la aplicación usando alguna de las extensiones para REST en el navegador (ej Advanced REST Client para Chrome o Postman) y entregar cliente configurado.

Instrucciones para Ejecutar

1. Crea carpeta en una ruta preferida con nombre “**ProyectoBackSpring**”
2. Dentro de la carpeta abre una terminal y ejecuta el comando
git clone <https://github.com/Jomeva12/crearApiListas.git>

En netbeans

1. Abrir proyecto seleccionando la ruta de descarga

El proyecto fue creado con las siguientes dependencias

The screenshot shows the NetBeans IDE project configuration window for a project named 'creacion-api'. The window is divided into several sections:

- Project:** Shows the project name 'creacion-api' and the package name 'com.jomeva.creacion-api'.
- Language:** Shows 'Java' selected as the language.
- Spring Boot:** Shows '3.1.4' selected as the version.
- Project Metadata:** Shows the group 'com.jomeva', artifact 'creacion-api', name 'creacion-api', description 'Prueba tecnica creacion de api sodaTime Spring Boot', and package name 'com.jomeva.creacion-api'.
- Dependencies:** Lists the following dependencies:
 - Spring Boot DevTools** (DEVELOPER TOOLS): Provides fast application restarts, LiveReload, and configurations for enhanced development experience.
 - Spring Web** (WEB): Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
 - H2 Database** (SQL): Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.
 - Spring Data JPA** (SQL): Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.
 - Lombok** (DEVELOPER TOOLS): Java annotation library which helps to reduce boilerplate code.

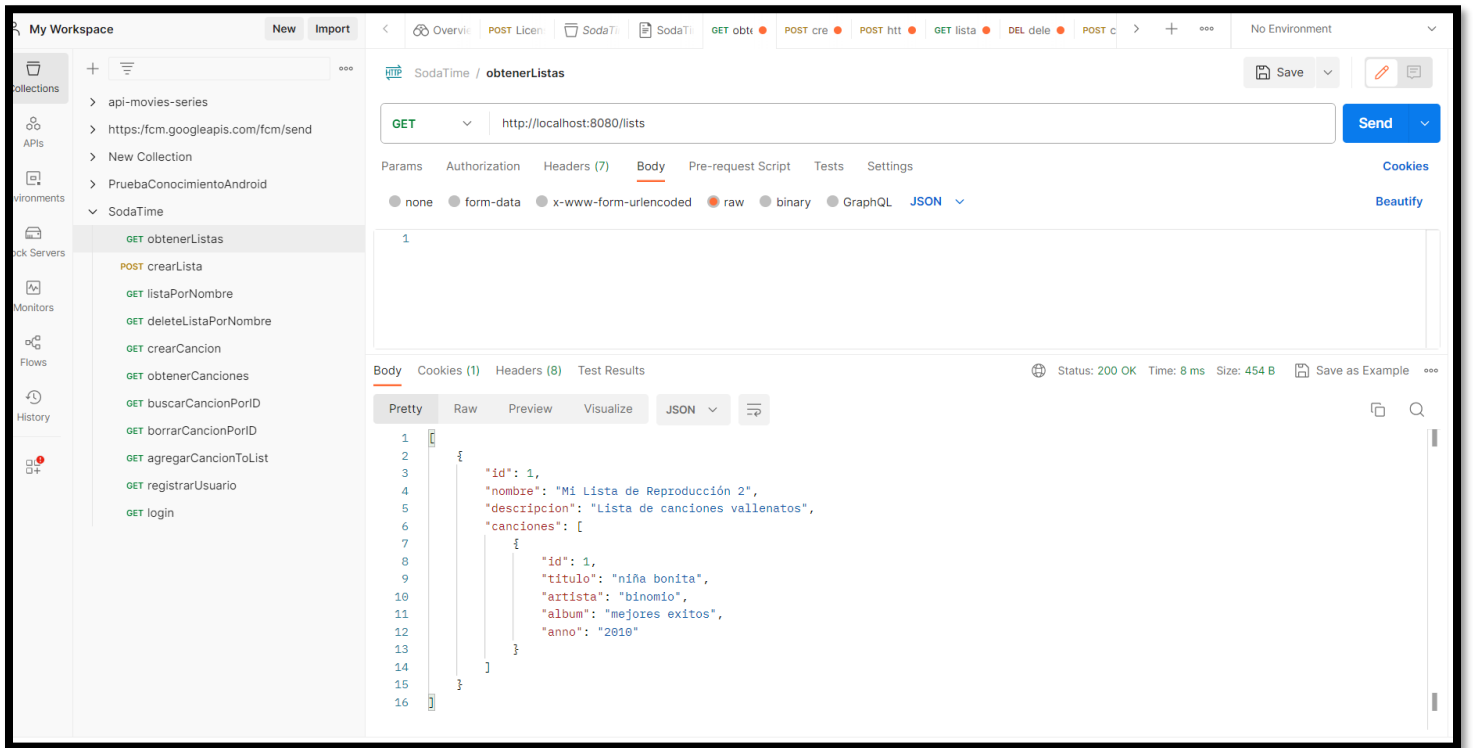
At the bottom of the window, there are buttons for 'GENERATE', 'EXPLORE', and 'SHARE...'. The 'GENERATE' button has a tooltip that says 'CTRL + G'.

Ejecutar prueba

Los endPoint creados son los siguientes:

Función	Method	EndPoint
Obtener todas las listas	GET	http://localhost:8080/lists
Crear listas	POST	http://localhost:8080/lists
Obtener lista por nombre	GET	http://localhost:8080/lists/get/nombre
Borrar lista por nombre	DELETE	http://localhost:8080/lists/delete/nombre
Crear canciones	POST	http://localhost:8080/canciones
Obtener canciones	GET	http://localhost:8080/canciones
Obtener canción por id	GET	http://localhost:8080/canciones/id
Borrara canción por id	DELETE	http://localhost:8080/canciones/id
Agregar canción a lista con id de lista	PUT	http://localhost:8080/lists/add/idLista

Pruebas realizadas en Postman



1. Levantar la aplicación “run” en la clase principal “CrearApiApplication”
2. Verificar la base de datos h2 en <http://localhost:8080/h2-console/>
3. Verificar los datos para acceder según la imagen
4. Test connection y conect para verificar las tablas creadas.
5. Realizar pruebas en postman con los endpoint entregados.

Login

Saved Settings: Generic H2 (Server) ▼

Setting Name: Generic H2 (Server) Save Remove

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:testdb

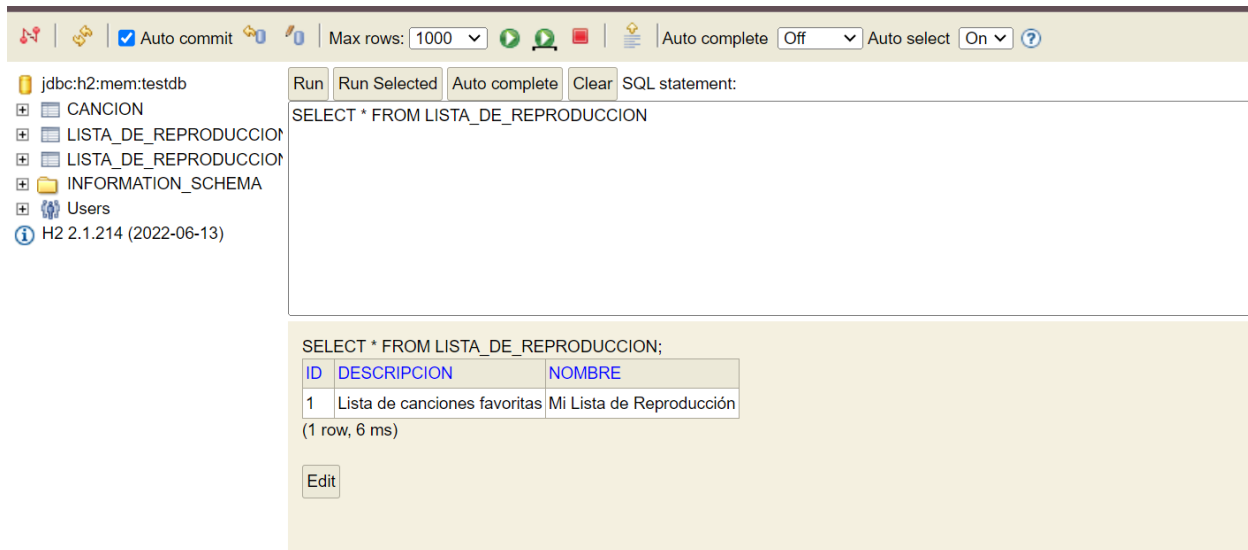
User Name: sa

Password:

Connect Test Connection

Test successful

Imagen de la Base de datos



Guía para Ejecutar la Aplicación De Prueba Técnica Consumir API

Requisitos Previos

Antes de comenzar, asegúrate de que tu sistema cumpla con los siguientes requisitos:

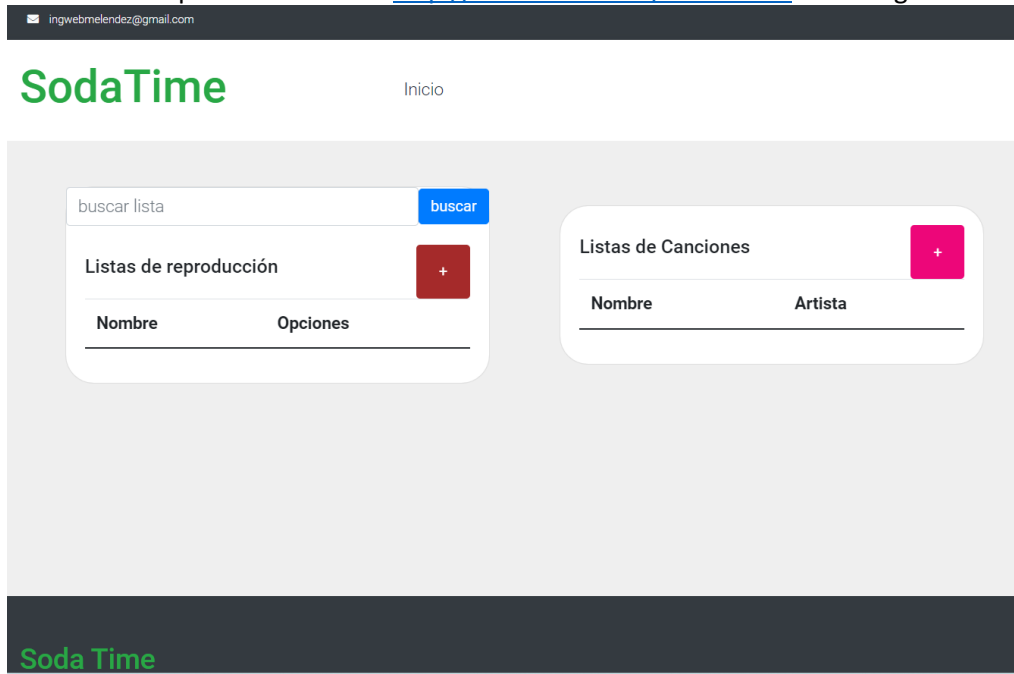
Instalado visual Code <https://code.visualstudio.com/download>

Instalar extensión live server desde el editor o desde el enlace
<https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>

Instrucciones para Ejecutar

1. Clic derecho dentro del contenido del index.html y seleccionar "open with live server"

2. Se levanta la aplicación en la url <http://127.0.0.1:5500/index.html> con la siguiente interfaz



Interactuar con la aplicación

1. En la card de listas de reproducción presionar el botón con el signo “+” para agregar una nueva lista
2. La lista se crea sin canciones, puedes seguir agregando listas
3. Cada lista creada muestra tres botones: agregar, ver y eliminar.
4. Para agregar canciones las canciones deben estar creadas en la otra card ubicada a la derecha.
5. Si ya existen canciones en la lista de canciones se pueden editar las listas de reproducciones para agregar
6. Posterior puede consultar el listado de canciones asociado a la lista de reproducción
7. Puede buscar listas de reproducción por nombre.

