Aquí hay algunas recomendaciones para diseñar e implementar esta aplicación de "rockola virtual":

1. \*\*Plataforma y tecnologías\*\*: Puedes desarrollar una aplicación móvil nativa para iOS y Android, o una aplicación web progresiva (PWA) que funcione en cualquier dispositivo con un navegador moderno. Las PWAs son más fáciles de desarrollar y mantener.

2. \*\*Interfaz de usuario intuitiva\*\*: La aplicación debe tener una interfaz sencilla que permita a los usuarios buscar y seleccionar canciones de una manera fácil e intuitiva. Puedes integrar un buscador y una vista de lista de canciones/álbumes.

3. \*\*Integración con servicios de música\*\*: En lugar de almacenar toda la música localmente, lo más práctico es integrarte con servicios de streaming de música como Spotify, Apple Music o YouTube Music a través de sus APIs. Esto te evitará problemas de derechos de autor.

4. \*\*Sistema de cola de reproducción\*\*: Implementa un sistema de cola para que las canciones solicitadas por los usuarios se reproduzcan de manera ordenada. Puedes mostrar la lista de canciones en cola para que los usuarios vean lo que se reproducirá próximamente.

5. \*\*Control de volumen y reproducción\*\*: Permite que los usuarios puedan pausar, reanudar, avanzar o retroceder en la canción que se está reproduciendo en el local.

6. \*\*Restricción por ubicación\*\*: Para asegurarte de que solo se pueda usar la aplicación dentro de tu local, puedes implementar un sistema de geolocalización que verifique si el usuario se encuentra dentro de un radio específico de tu establecimiento.

7. \*\*Monetización (opcional)\*\*: Puedes agregar un sistema de créditos o monedas virtuales para que los usuarios "alimenten" la rockola antes de poder solicitar una canción. Esto puede generar ingresos adicionales.

8. \*\*Pantalla pública\*\*: Además de la aplicación móvil, puedes tener una pantalla pública en tu local que muestre la canción que se está reproduciendo actualmente, la lista de canciones en cola y quizás algunos anuncios o publicidad.

Para facilitar el desarrollo, te recomiendo utilizar un framework moderno como React Native (para aplicaciones móviles) o React (para PWA), junto con una base de datos en tiempo real como Firebase Realtime Database para manejar la cola de reproducción y la sincronización en tiempo real entre dispositivos.

Sí, hay algunos tutoriales y proyectos de ejemplo en YouTube que pueden servirte como guía para crear tu aplicación de "rockola virtual". Aquí te comparto algunos:

1. \*\*Spotify Queue Application with React & Firebase\*\*

- Link: https://www.youtube.com/watch?v=LXfV7hy2SkQ

- Este tutorial te muestra cómo crear una aplicación web para hacer cola y reproducir música de Spotify utilizando React y Firebase.

2. \*\*React Native Spotify App\*\*

- Link: https://www.youtube.com/watch?v=uQx5jKbyI-Q

- Este proyecto te enseña a crear una aplicación móvil con React Native que se conecta a la API de Spotify para buscar y reproducir música.

3. \*\*Jukebox App with React.js & Spotify API\*\*

- Link: https://www.youtube.com/watch?v=pTlMdMJHfBo

- Un tutorial para construir una aplicación web de "rockola" con React.js y la API de Spotify.

4. \*\*Build a Music Streaming App with React Native & Expo\*\*

- Link: https://www.youtube.com/watch?v=R3j6Z2BxTAo

- Este video te guía en el desarrollo de una aplicación móvil de streaming de música con React Native y Expo, aunque no utiliza una API de música específica.

Aunque estos proyectos no sean idénticos a tu idea de "rockola virtual", te pueden dar una buena base y ejemplos sobre cómo integrar servicios de música, manejar colas de reproducción, interfaces de usuario y otras funcionalidades que necesitarás en tu aplicación.

Link to projects

[Full Stack Music Player | For Beginners | React, MongoDb, NodeJs, FIrebase (youtube.com)](https://www.youtube.com/watch?v=2HvISWe9ArM&list=PL5JH4SeiwEwxQ6FwmZLrutvdznN6xrem-)

[Build a Music Player app with React Native, Expo, Typescript and Zustand (youtube.com)](https://www.youtube.com/watch?v=9CElrkFwiBU)

[🟦 Build A Advance Music Player Application with React JS | Fully Customized (youtube.com)](https://www.youtube.com/watch?v=dALOX3sIfsk)