

## CIERRE DEL PROYECTO

Jefferson Jair Figueroa Escobar  
Juan David Beltran Orjuela  
John Jairo Paez Albino  
Juan Sebastian Leguizamón Silva

27 de Noviembre del 2025

Universidad Nacional de Colombia  
Sistemas de Información

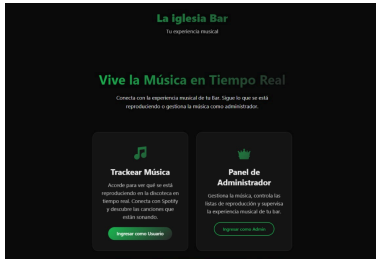
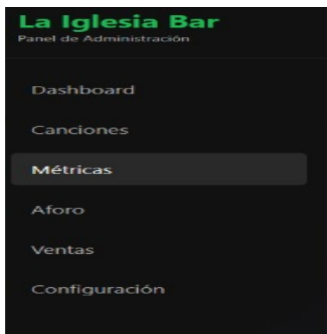
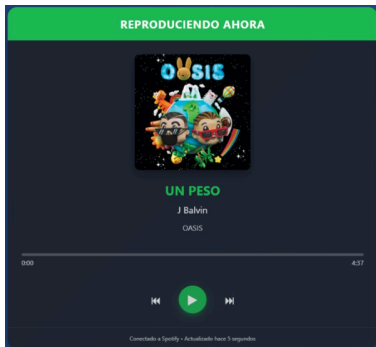
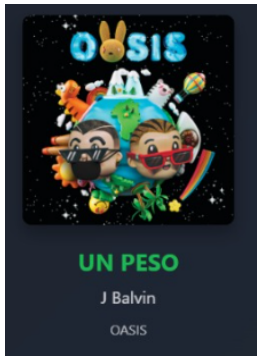
## Índice

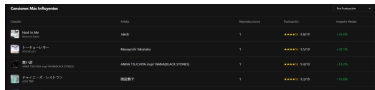
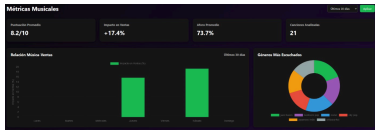
<b>Validación del Sistema</b> .....	<b>3</b>
Tabla de Requerimientos.....	3
<b>Dashboards y Análisis</b> .....	<b>6</b>
Herramienta a utilizar.....	6
Conectar la base de datos.....	6
Dashboards.....	7
Dashboard 1.....	7
Dashboard 2.....	8
Resumen del proyecto.....	10

## Validación del Sistema

### Tabla de Requerimientos

Presentamos actualmente la siguiente tabla con los requerimientos que funcionan y los que no, junto a algunas capturas

Requisito	¿Funciona?	Captura
RF 1: Como usuario quiero visualizar una página de inicio antes de iniciar sesión, para ver que si estoy entrando a el aplicativo correcto	<b>SI.</b> La aplicación cumple con un lobby donde el usuario puede visualizar la página a la que ingresa y a la que va ingresar posteriormente	
RF 2: Como usuario quiero ver un DashBoard, para ingresar a los apartados de canciones, listas, métricas u otros.	<b>SI.</b> En la app hay una barra lateral la cual permite acceder a las distintas opciones que hemos desarrollado para el usuario	
RF 3: Como usuario, quiero tener control sobre las canciones, para pausar o cambiar de canción.	<b>SI.</b> hemos implementando un reproductor en el cual se puede cambiar de canción o pausar la misma	
RF 4: Como usuario, quiero visualizar el nombre de la canción en la interfaz gráfica para identificar la canción que está sonando.	<b>SI.</b> El sistema permite visualizar la imagen del álbum de la canción, el nombre, y el artista.	
RF 5: Como usuario, quiero visualizar el nombre de la	<b>NO.</b> Esto no fue posible agregarlo debido a	

lista de reproducción, para identificar si es la lista de reproducción programada.	problemas, con ingresar los datos de las playlist a la BD, pero igual nos permite ver la tendencia por el género musical	
RF 6: Como usuario, quiero calificar las canciones, para plasmar el ambiente que se vivió mientras estas estaban sonando.	<b>SI.</b> Agregamos un sistema de estrellas como calificación las cuales se van jerarquizando de la mejor a la peor.	
RF 7: Como usuario, quiero llenar un formulario sobre las canciones presentadas en ese día para plasmar qué tan útiles pueden ser estas canciones en el bar	<b>NO.</b> Para esta entrega no fue viable usar el formulario, debido a que en su mayoría los datos son simulados con funciones matemáticas, pero una vez el programa entre a producción no habría problema crear el formulario pues los espacios en la BD ya están.	
RF 8: Como Administrador, quiero tener un panel de métricas, para analizarlas y tomar decisiones sobre las canciones que deberían estar posteriormente en las listas de reproducción.	<b>SI.</b> El sistema le permite al administrador mirar las estadísticas y métricas relacionadas con la música	
RF 9: Como usuario, quiero visualizar una BlackList, para mover las canciones que tienen un desempeño muy bajo o su reproducción afecta de forma negativa a el bar.	<b>NO.</b> No encontramos la forma de unir las playlist de Spotify con nuestro software, por lo que tampoco logramos hacer que se puedan modificar las mismas	
RNF 1: El sistema debe responder en menos de 2 segundos	<b>SI.</b> la conexión al api es muy sólida por lo que la velocidad de carga principalmente depende de la conexión de internet	
RNF 2: Las contraseñas deben encriptarse con SHA-256	<b>NO APLICA.</b> nuestro único logueo es a través del api de spotify la cual cuenta con toda la reglamentación adecuada para la seguridad	

	del usuario, explícitamente usamos el método de autenticación Oauth 2.0.	
RNF 3: La interfaz debe ser accesible desde dispositivos móviles	<b>NO.</b> por motivos de tiempo no invertimos tiempo en hacer la app web responsive por lo que funciona únicamente en ordenadores por otro lado a futuro no son cosas muy complicadas de agregar pero si llevan su tiempo.	
RNF 4: El sistema debe tener 99.9% de disponibilidad	Para este caso es complicado en versiones futuras implementaremos una solución a problemas de como quedarse sin conexión de ethernet, guardando un backup de las playlist de forma local para que el bar siga funcionando con normalidad aun cuando no haya acceso a internet	
RNF 5: Código modular, claro, documentado para fácil actualización	<b>SI.</b> El código fue escrito por un grupo de personas que conocen muy bien las herramientas utilizadas, y en el proceso se fue generando la documentación de todo el proyecto	
RNF 6: Implementar autenticación y autorización robusta para controlar accesos	<b>SI.</b> Tenemos esto bien manejado gracias a los tokens que otorga el API de spotify	
RNF 7: El sistema debe garantizar integridad de datos y evitar pérdidas	<b>SI.</b> La conexión a la base de datos es sólida, sin embargo en caso de perder la conexión con el servidor pueden ocurrir pérdidas de información	
RNF 8: La experiencia de usuario debe ser intuitiva y fácil de usar	<b>SI.</b> La interfaz es muy fácil de usar ya se que se asemeja a la app de spotify	

## **Dashboards y Análisis**

### ***Herramienta a utilizar***

En nuestro proyecto, elegimos utilizar Python para el desarrollo del backend debido a su versatilidad y facilidad de uso, que nos permite escribir código claro y eficiente, acelerando el proceso de desarrollo. Python ofrece una amplia variedad de frameworks y herramientas que facilitan la integración con bases de datos NoSQL como MongoDB, la cual seleccionamos por su flexibilidad para manejar grandes volúmenes de datos y su capacidad para almacenar información no estructurada, ideal para nuestro proyecto. Además, implementamos la integración con la API de Spotify para acceder a datos musicales en tiempo real, lo cual nos proporciona una fuente confiable y dinámica para enriquecer nuestro sistema.

Para el frontend, optamos por tecnologías clásicas como JavaScript, HTML y CSS, que nos permiten construir una interfaz de usuario interactiva y fácil de usar, garantizando que los usuarios puedan navegar y explotar las funcionalidades del sistema de manera intuitiva. Esta división clara entre backend y frontend facilita la escalabilidad y el mantenimiento del proyecto, permitiendo que cada parte evolucione según sus necesidades específicas.

Como respaldo adicional para la visualización y análisis de datos, implementamos un dashboard en Power BI. Esta herramienta nos ofrece una plataforma robusta y visualmente atractiva para explorar y presentar datos, complementando el trabajo realizado en el backend y frontend y aportando valor con un enfoque analítico y visual para la toma de decisiones.

### ***Conectar la base de datos***

Para conectar nuestra base de datos MongoDB, primero creamos una cuenta en MongoDB Atlas, una plataforma en la nube que permite gestionar bases de datos de manera sencilla y segura. Luego, dentro de la plataforma, creamos un proyecto nuevo y configuramos un clúster seleccionando el plan gratuito, lo cual nos permite contar con un entorno de base de datos escalable y accesible desde la nube.

Una vez creado el clúster, generamos un usuario de base de datos con permisos para leer y escribir, asegurándonos de guardar sus credenciales de acceso de manera segura. Para conectar nuestra aplicación al clúster, obtenemos la URI de conexión desde el panel de Atlas, la cual contiene la información necesaria para establecer la comunicación con la base de datos en la nube.

Para mantener la seguridad y flexibilidad, almacenamos esta URI en un archivo de configuración de entorno (.env) dentro de nuestro proyecto, lo que facilita el manejo de credenciales sin exponerlas directamente en el código fuente. Finalmente, utilizando Python, nos conectamos a MongoDB mediante librerías especializadas que leen esta URI desde el archivo .env, permitiendo así una conexión segura y eficiente con la base de datos para realizar operaciones de consulta, inserción y actualización de datos.

Este proceso garantiza que nuestro backend gestione la información de manera confiable y que pueda escalar según las necesidades del proyecto, manteniendo buenas prácticas de seguridad y organización en el desarrollo.

### **Conexión tablero Power BI**

El tablero que realizamos con Power BI, tiene precargada la conexión, lo único que hace falta es instalar el controlador, el cual se puede encontrar en el siguiente enlace:

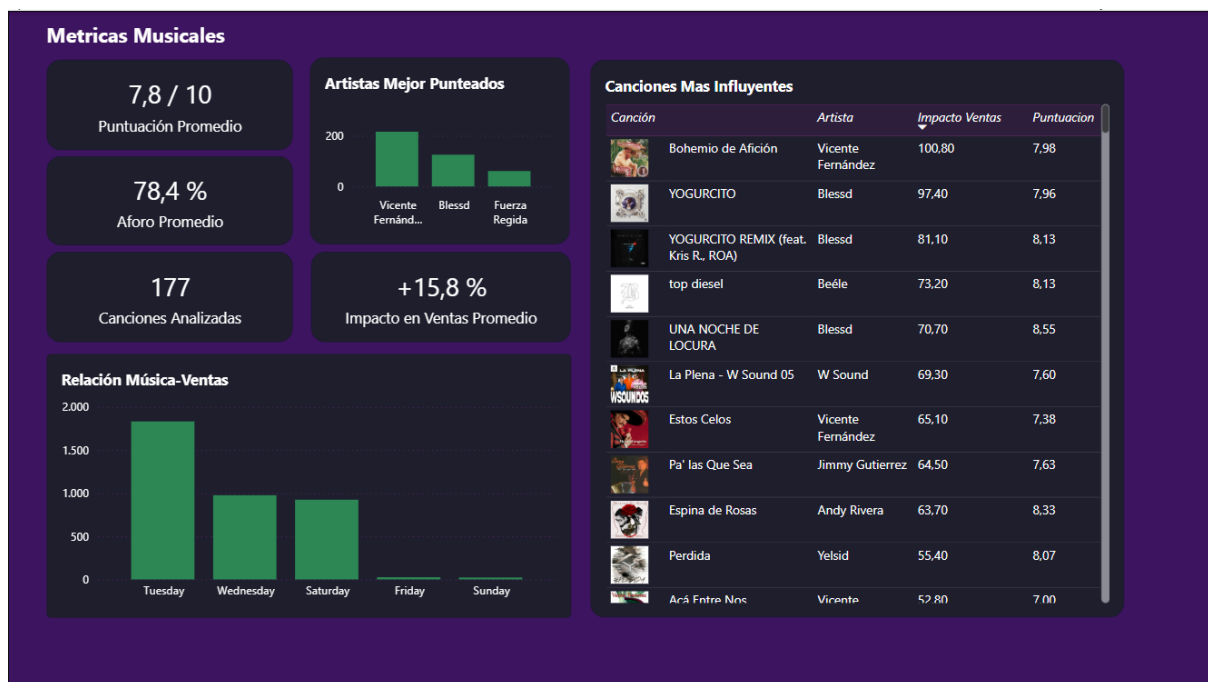
<https://www.mongodb.com/try/download/odbc-driver>, una vez instalado el ODBC para mongo, es abrir el archivo .pbix y oprimir el botón de actualizar, para obtener los datos más recientes de la db.

Nota: El archivo .pbix se puede encontrar en:

<https://github.com/SebastianSLJ/ProyectoFinalSI2025-II/tree/main/Power%20bi>

## **Dashboards**

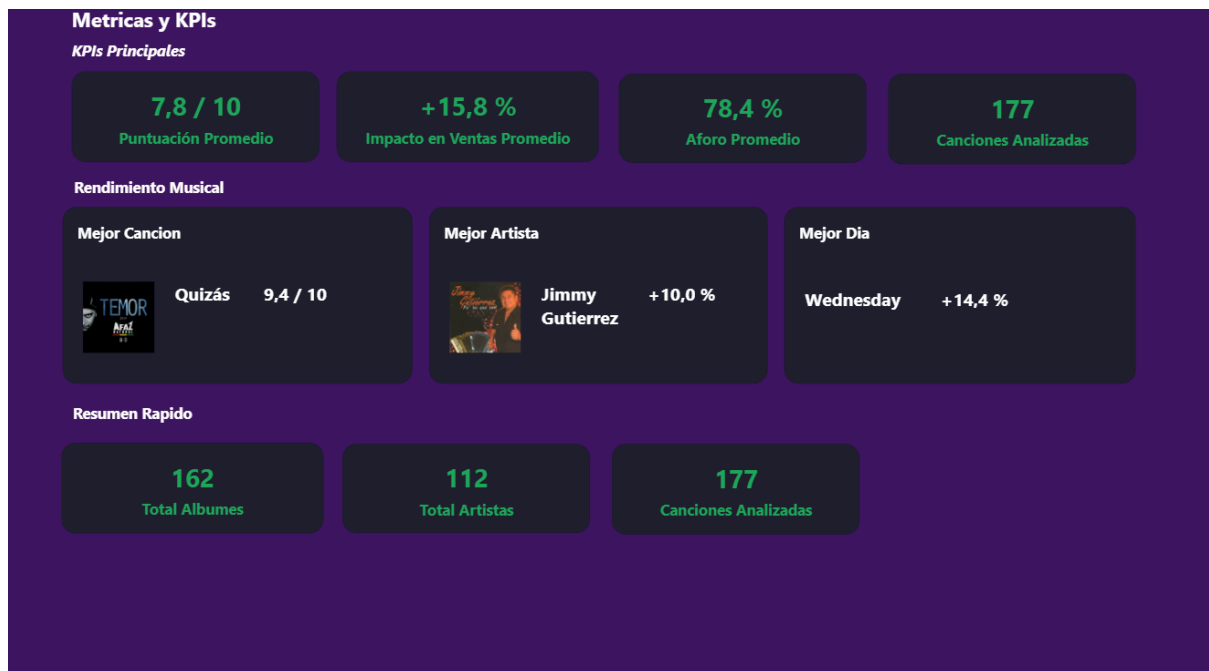
### **Dashboard 1**



- Puntuación promedio de la música.
- Aforo promedio del bar.
- Número total de canciones analizadas.
- Impacto promedio de la música en las ventas (porcentaje).

- Artistas mejor puntuados.
- Canciones más influyentes (impacto en ventas y puntuación).
- Relación música–ventas por día de la semana.

### *Dashboard 2*



- Puntuación promedio de la música.
- Impacto promedio de la música en las ventas (porcentaje).
- Aforo promedio del bar.
- Total de canciones analizadas.
- Mejor canción (título y calificación).
- Mejor día (día de la semana e incremento porcentual).
- Total de álbumes.
- Total de artistas.

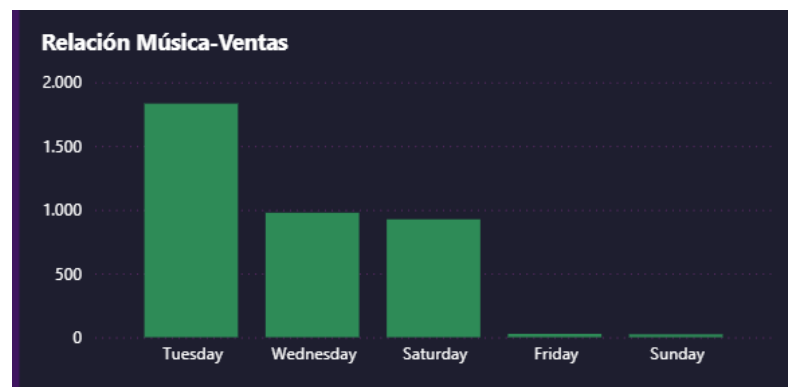


## Análisis de Datos:

- **Mejor día para ventas y aforo**

### ¿Qué encontramos?

El miércoles presenta el mejor rendimiento global: es el día con mayor mejora relativa en ventas (alrededor de +14–15%) y un aforo promedio alto, mientras que otros días como viernes o domingo tienen un impacto mucho menor.



### ¿Qué recomendaría al negocio?

Concentrar las estrategias comerciales (promociones, eventos temáticos, activaciones de marca) en los miércoles y días con comportamiento similar, y revisar por qué otros días tienen bajo rendimiento para ajustar horarios, comunicación o tipo de música.

- **Artistas y canciones con mayor impacto en ventas**

### ¿Qué encontramos?

Algunos artistas específicos (por ejemplo Jimmy Gutiérrez, Vicente Fernández, Blessd, entre otros) y canciones concretas se asocian con un impacto positivo en ventas y mejores puntuaciones promedio en la experiencia musical.

<i>Canción</i>	<i>Artista</i>	<i>Impacto Ventas</i>	<i>Puntuacion</i>
 Bohemio de Afición	Vicente Fernández	100,80	7,98
 YOGURCITO	Blessd	97,40	7,96
 YOGURCITO REMIX (feat. Kris R., ROA)	Blessd	81,10	8,13
 top diesel	Beéle	73,20	8,13
 UNA NOCHE DE LOCURA	Blessd	70,70	8,55
 La Plena - W Sound 05	W Sound	69,30	7,60
 Estos Celos	Vicente Fernández	65,10	7,38
 Pa' las Que Sea	Jimmy Gutierrez	64,50	7,63

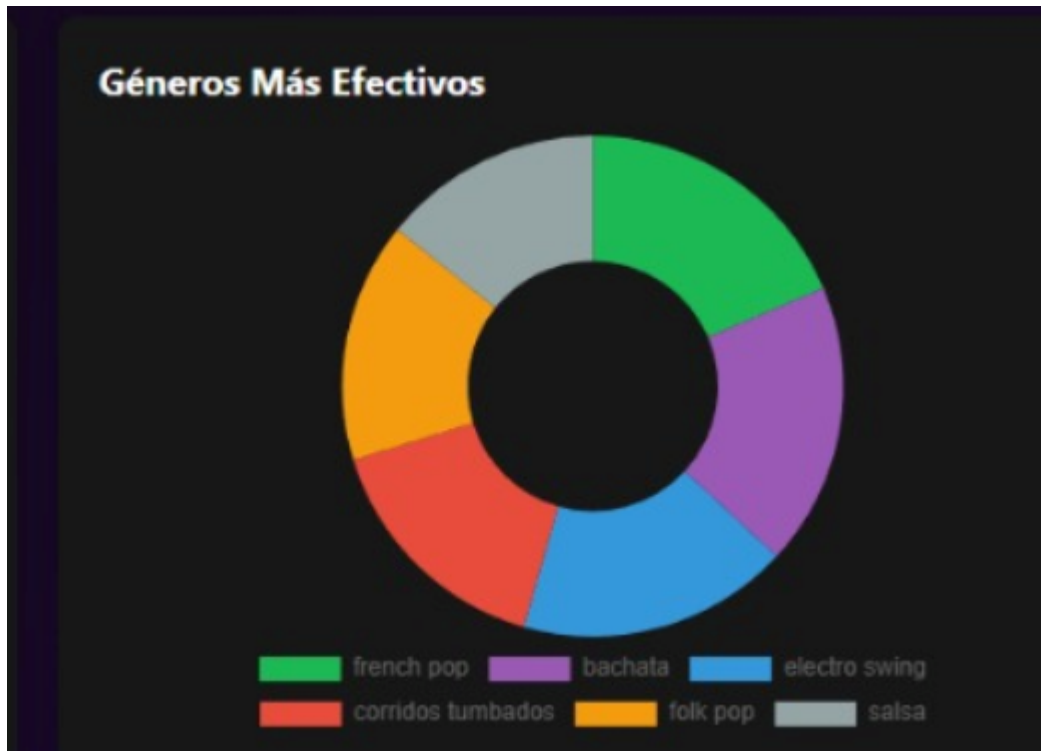
### ¿Qué recomendamos al negocio?

Priorizar estas canciones y artistas en las playlists de los momentos de mayor afluencia, crear bloques musicales especiales alrededor de ellos y probar campañas promocionales ligadas a estos nombres para maximizar consumo y permanencia.

- **Géneros musicales más efectivos**

### ¿Qué encontramos?

Ciertos géneros, como french pop, bachata, corridos tumbados, electro swing, folk pop o salsa, muestran mejor desempeño que otros en términos de impacto en ventas y/o valoración de la experiencia.



### **¿Qué recomendamos al negocio?**

Diseñar playlists segmentadas por horario usando principalmente los géneros más efectivos, testear combinaciones entre ellos y reducir el uso de géneros con bajo impacto, evaluando de forma continua si las ventas y el aforo mejoran con estos ajustes.

### ***Resumen del proyecto***

El proyecto desarrolla una solución de apoyo gerencial para el bar "La Iglesia Bar" en Bogotá, orientada a analizar cómo la música y los géneros musicales influyen en las ventas y el aforo. A partir de la API de Spotify, el sistema obtiene métricas musicales (artistas, canciones, géneros y valoraciones) que se almacenan en una base de datos y se relacionan con los datos de ventas simulados del bar, permitiendo identificar qué tipo de música favorece más el consumo en un ambiente donde las personas principalmente se reúnen a conversar y compartir.

Aunque no se trata de un ERP ni de un CRM en sentido estricto, la solución funciona como un sistema de información gerencial enfocado en la analítica de ventas y la ambientación musical del bar. Su rol es ofrecer indicadores y reportes para la toma de decisiones del jefe de barra, de forma similar a un módulo de Business Intelligence dentro de un ERP ligero especializado en bares y restaurantes.

La arquitectura propuesta separa claramente la capa operativa de la capa de análisis: el jefe de barra es el único usuario autorizado para acceder al sistema, autenticarse, consultar los

resultados y tomar decisiones sobre la programación musical, mientras que el equipo de TI se encarga del mantenimiento técnico y la evolución de la herramienta. A través de un dashboard gerencial, se presentan KPIs como impacto porcentual de la música en las ventas, mejores días, géneros más efectivos y artistas o canciones con mayor contribución, lo que convierte la solución en un sistema de información que apoya decisiones tácticas y estratégicas sobre la ambientación musical del bar y que, en perspectiva, puede adaptarse fácilmente a otros establecimientos similares