UNIVERSIDAD AUSTRAL DE CHILE

Facultad de Ciencias de la Ingeniería, Ingeniería Civil Informática

Bot de Discord: WabMusicBot

Arquitectura de Software

Segundo Semestre, 2024

Docente responsable
Matthieu Vernier

Alumnos
Francisco Morales, francisco.morales02@alumnos.uach.cl
Alann Kahler, alann.kahler@alumnos.uach.cl
Esperanza Tejeda, esperanza.tejeda@alumnos.uach.cl

29 de septiembre de 2024

Índice

1. Introducción

El presente proyecto consiste en el desarrollo de un bot de Discord, que permite reproducir música personalizada en llamadas de voz. Los usuarios que son administradores de un servidor podrán gestionar las canciones desde una plataforma web, añadiendo, eliminando o editando las canciones. El bot se controlará mediante comandos de Discord, permitiendo a los usuarios interactuar directamente desde el chat para reproducir la música en tiempo real en los canales de voz.

El bot se diseñará considerando tanto los requisitos funcionales, como la capacidad de reproducir, pausar, y gestionar listas de reproducción; y no funcionales, como la disponibilidad continua y la capacidad. Este informe posee además, la documentación necesaria para entender la arquitectura del proyecto, mediante el uso de diagramas que lo representan.

2. Requisitos Funcionales y no Funcionales

En esta sección se definirán los requisitos funcionales y no funcionales del proyecto.

2.1. Requisitos Funcionales

- El bot debe ser capaz de unirse a canales de voz en Discord cuando sea invocado por un comando del usuario.
- El sistema debe permitir a los usuarios reproducir música a través de comandos como `!play`, `!pause`, y `!stop`.
- Los usuarios de algún servidor deben poder gestionar la cola de reproducción, añadiendo, eliminando o reordenando canciones mediante comandos específicos.
- La plataforma web debe permitir a los administradores gestionar canciones listas de reproducción personalizadas y asociarlas a los canales de Discord.
- El bot debe poder acceder a múltiples servidores y operar de manera independiente en cada uno.

2.2. Requisitos no Funcionales

Los requisitos no funcionales son cruciales para garantizar la calidad y el rendimiento del sistema. A continuación, se detallan los aspectos más importantes:

- **Disponibilidad**: Es fundamental que el bot funcione de manera continua y sin interrupciones, asegurando que no se presenten fallos durante la reproducción de canciones. Debe estar disponible en todo momento para garantizar una experiencia fluida.
- **Capacidad**: El bot debe ser capaz de atender a una gran cantidad de usuarios simultáneamente sin comprometer el rendimiento, ya que muchos servidores de Discord pueden querer utilizar el bot al mismo tiempo. La escalabilidad es clave para asegurar un funcionamiento óptimo en entornos de alta demanda.
- **Comportamiento Temporal**: El tiempo de respuesta debe ser lo más bajo posible, sin demoras perceptibles. Un retraso considerable en la ejecución de los comandos podría afectar significativamente la experiencia del usuario.
- **Seguridad**: Dado que el bot interactúa con cuentas de Discord y puede manejar información sensible, se deben implementar medidas de seguridad robustas para proteger los datos de los usuarios, tanto en la plataforma web como en el bot mismo.

3. Diagramas

En la siguiente sección se muestran los diagramas de casos de uso, clases, componentes y despliegue, que representan la arquitectura del proyecto y ayudan con la comprensión de este mismo.

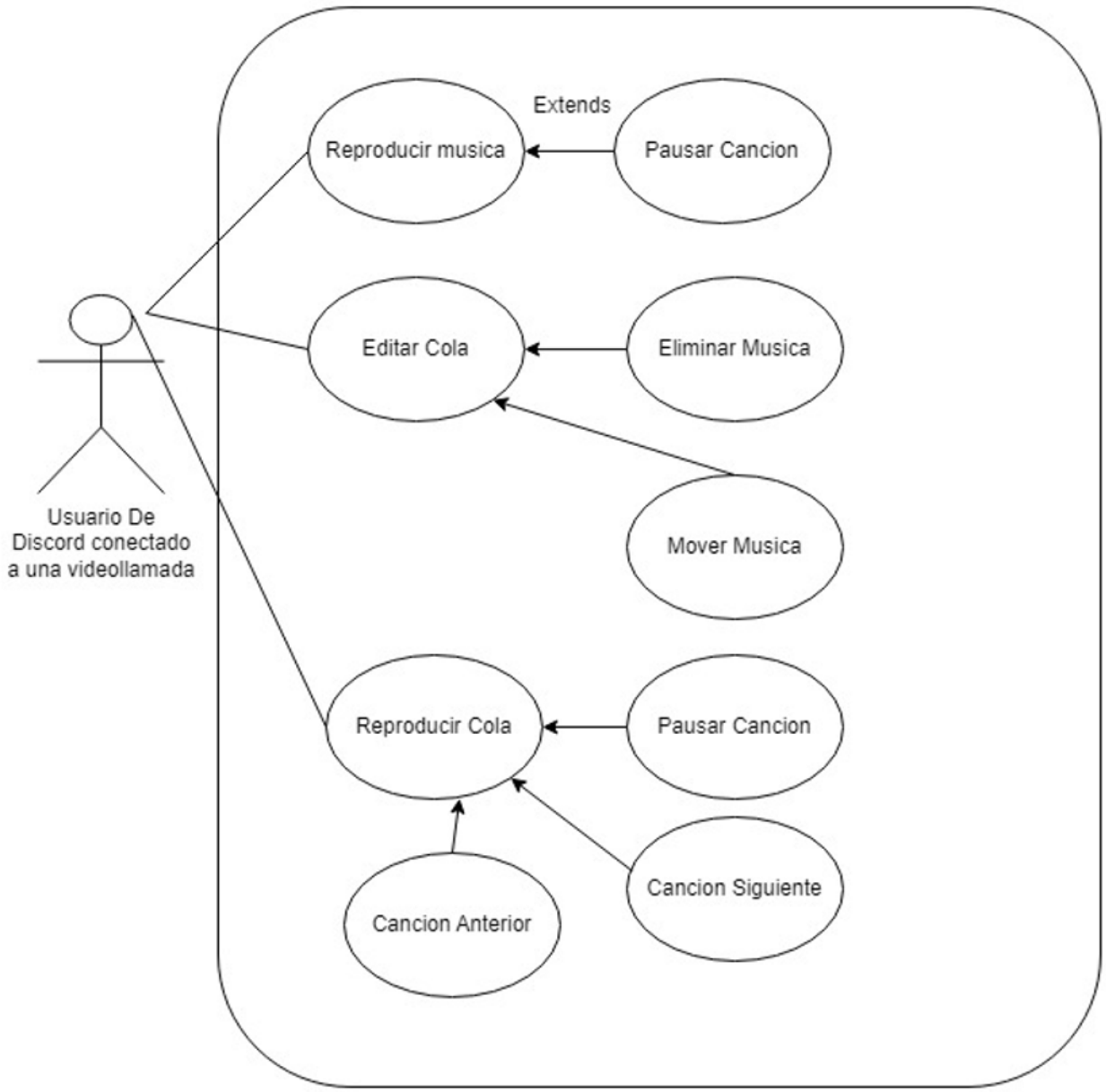

Figura 1: Casos de Uso Bot de Discord

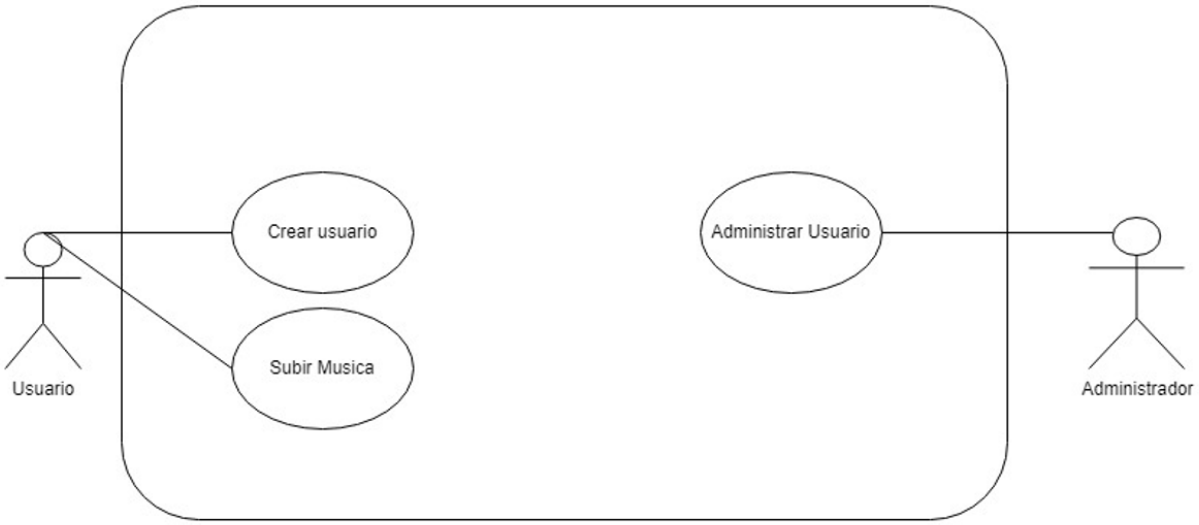

Figura 2: Casos de Uso Página Web

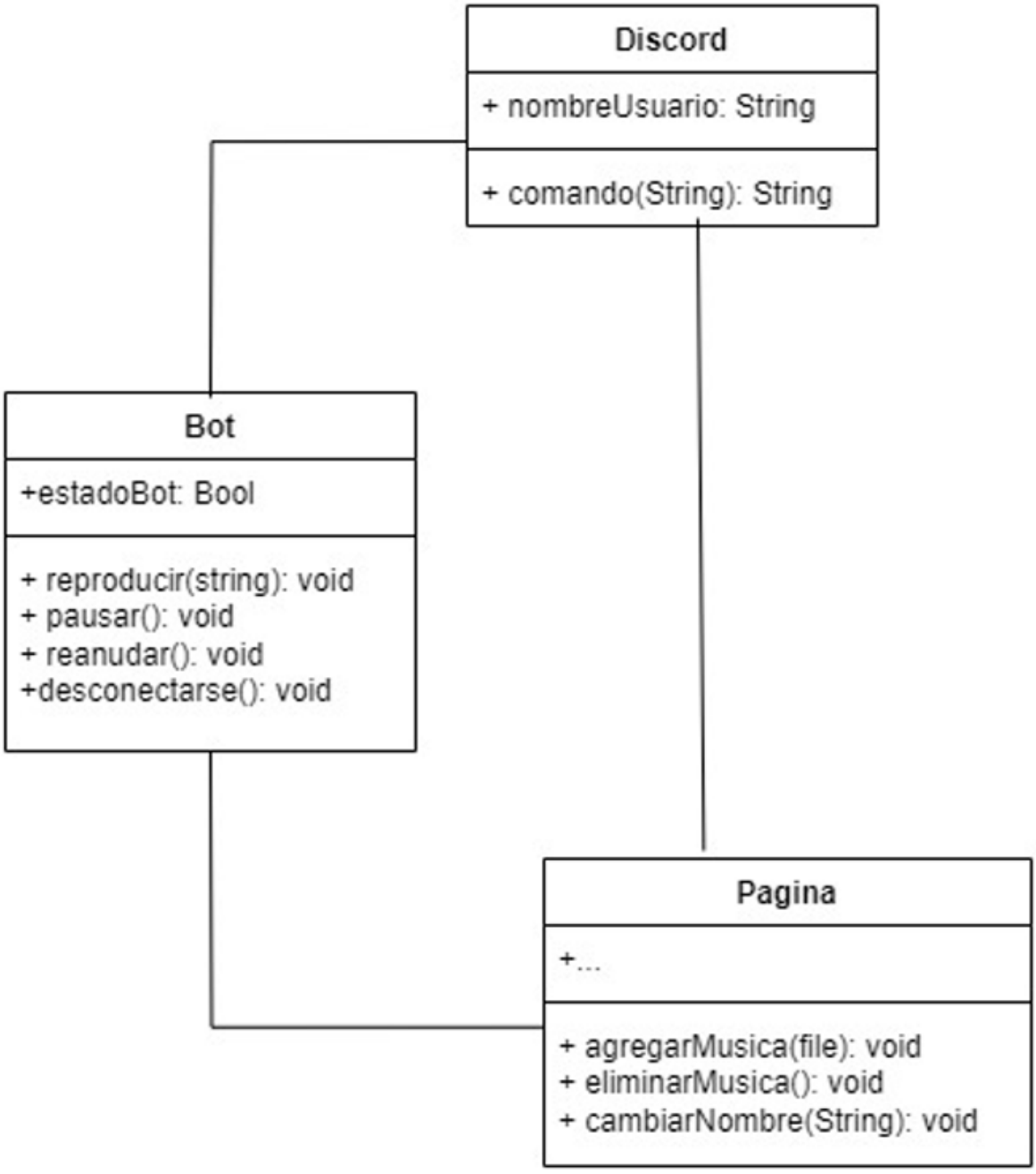

Figura 3: Diagrama de Clases

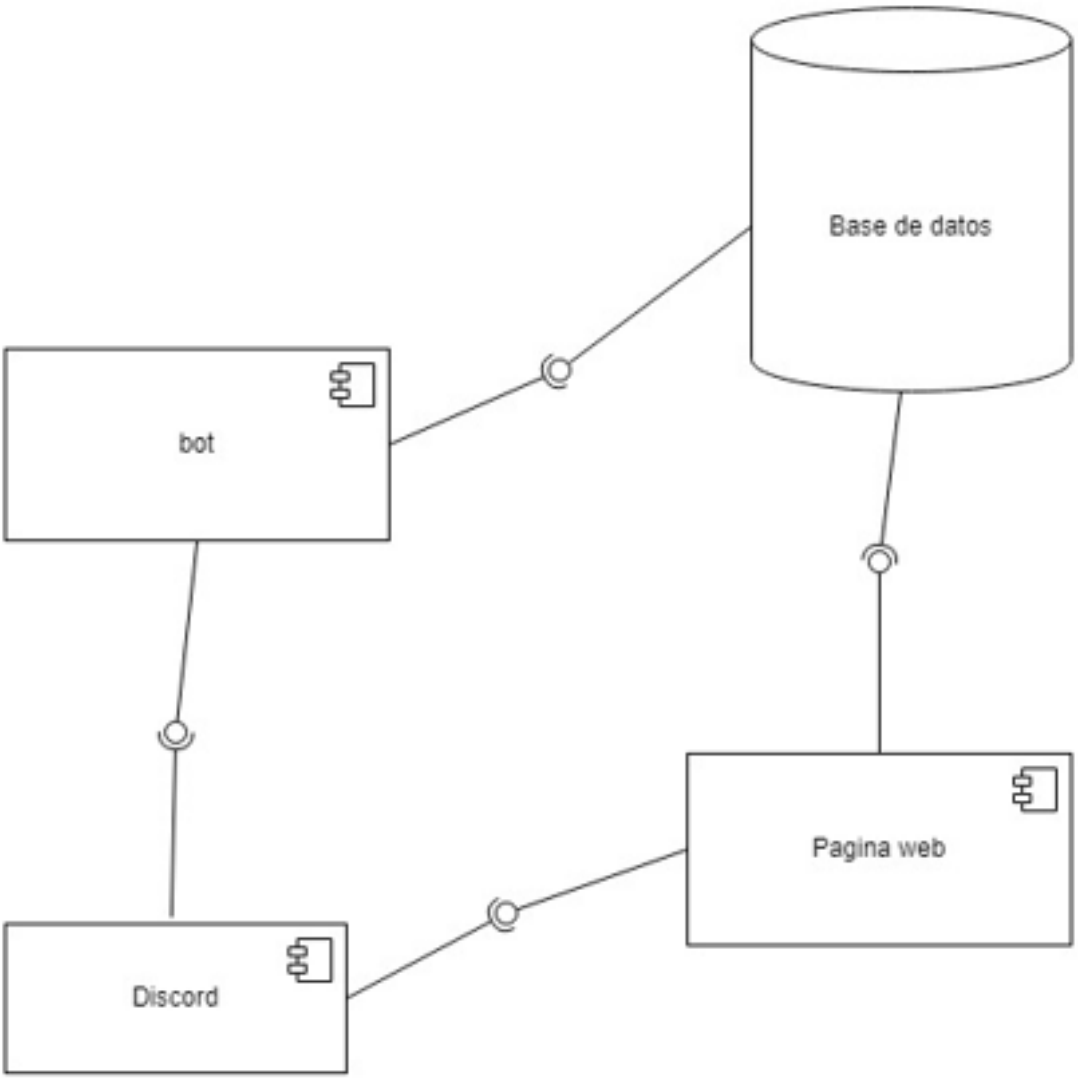

Figura 4: Diagrama de Componentes

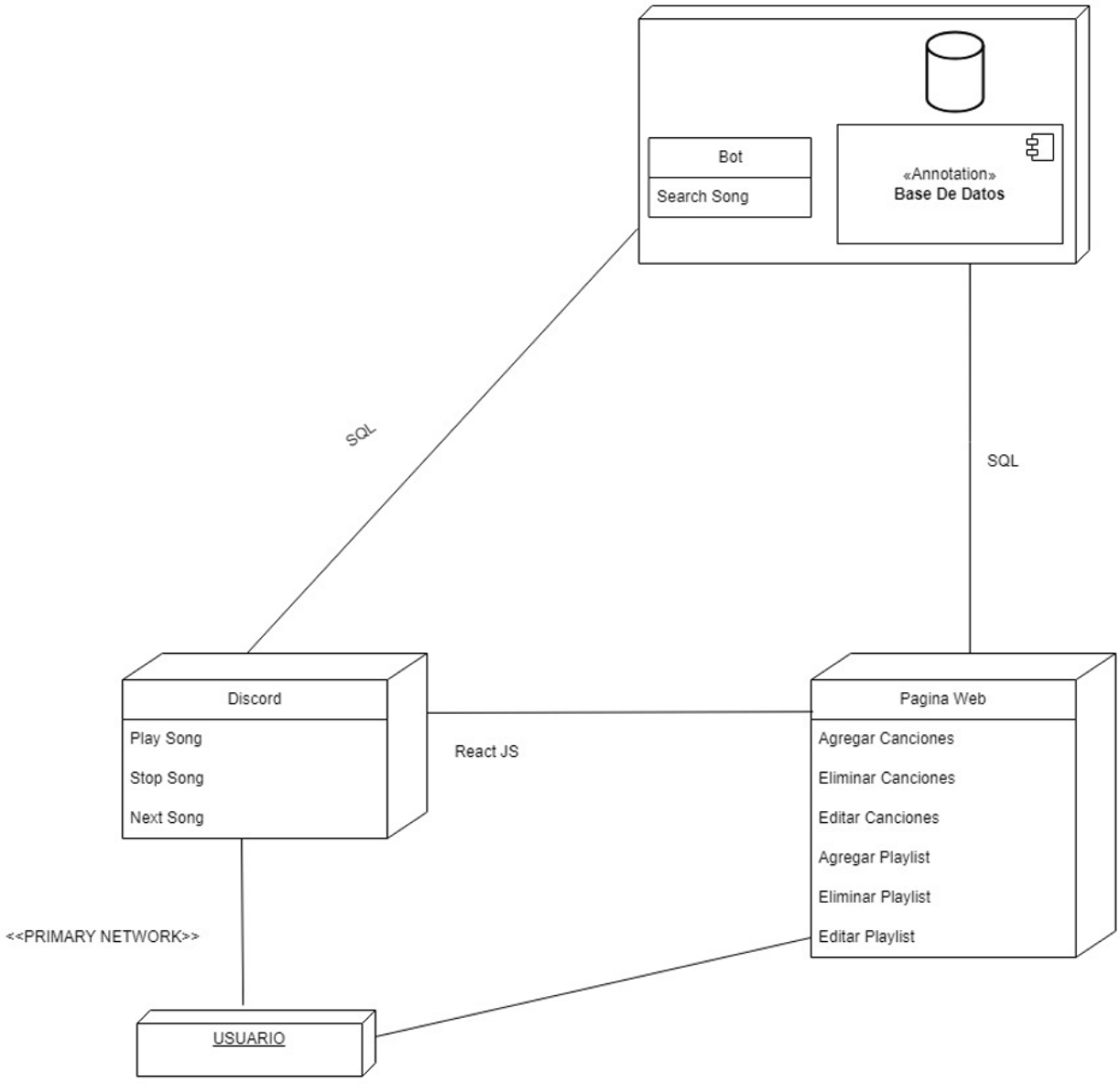

Figura 5: Diagrama de Despliegue