

# Kitsu-bot



Proyecto realizado por:  
Santiago Herrera Fernández

# **Índice**

## **1. Introducción**

## **2. Planteamiento inicial**

### **2.1 Tecnologías usadas**

### **2.2 Estructura y base de datos**

## **3. Desarrollo**

## **4. Mejoras**

## **5. Conclusión**

# 1. Introducción

Inicialmente este proyecto nace de mí ganas y mi propio deseo de crear un bot de Discord completamente desarrollado por mí, que fuera plenamente funcional y al cual si yo necesitaba alguna funcionalidad, ya sea de moderación o de comunidad yo se la pudiera implementar sin necesidad de estar añadiendo otros bots y llenando un servidor de Discord de bots a lo mejor solo para una funcionalidad.

Entonces aprovechando ese deseo personal y que para desarrollar el propio bot en sí puedo emplear el conocimiento que he aprendido sobre el curso en gran parte, ya que este proyecto se basa principalmente en JavaScript, aunque si es verdad que no es puro ya que se usará node.js , pero aprovechando estos factores me dije manos a la obra. Además de así presentar un proyecto de fin de grado más único y diferente a mis compañeros.

## 2. Planteamiento inicial

### 2.1 Tecnologías usadas

El planteamiento inicial en mi caso fue más fácil ya que las tecnologías a utilizar en este proyecto las tenía bastante claras las cuales son las siguientes:

#### Node.js

Node.js es un entorno de tiempo de ejecución de JavaScript (de ahí su terminación en .js haciendo alusión al lenguaje JavaScript). Este entorno de tiempo de ejecución en tiempo real incluye todo lo que se necesita para ejecutar un programa escrito en JavaScript.

¿Para que sirve node.js?

Node.js utiliza un modelo de entrada y salida sin bloqueo controlado por eventos que lo hace ligero y eficiente (con entrada nos referimos a solicitudes y con salida a respuestas).



#### Discord.js

Discord.js es un módulo de node.js que nos permite interactuar con la Api de Discord sin complicaciones y de esa forma poder desarrollarnuestro bot de Discord.



#### MongoDB

MongoDB es un sistema de base de datos NoSQL, orientado a documentos y de código abierto.

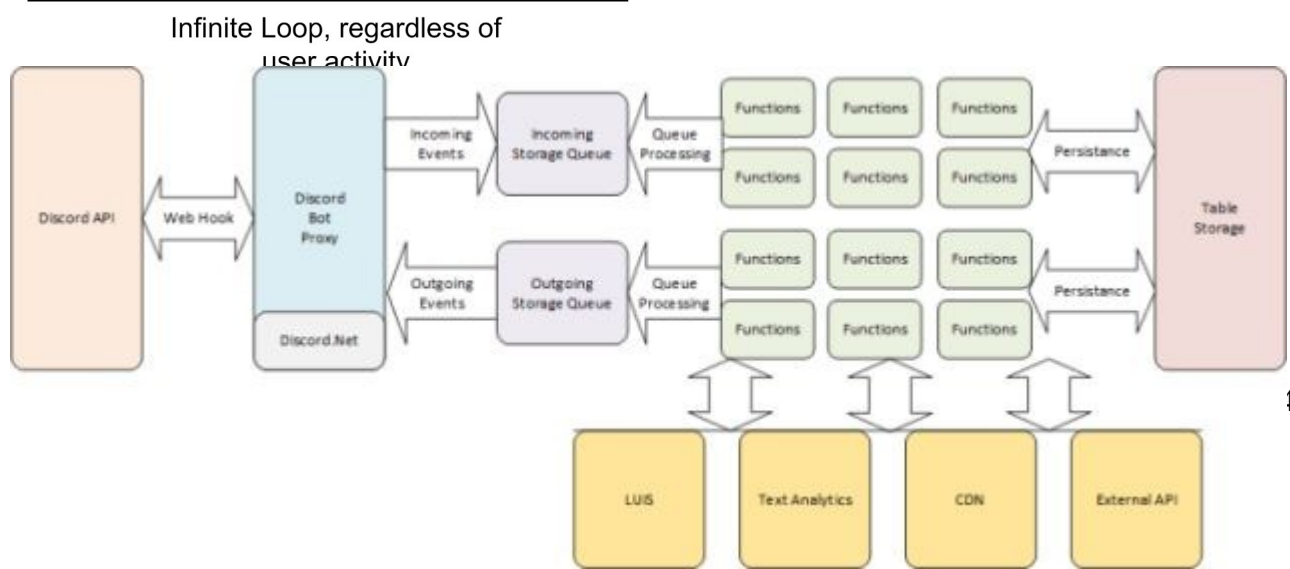
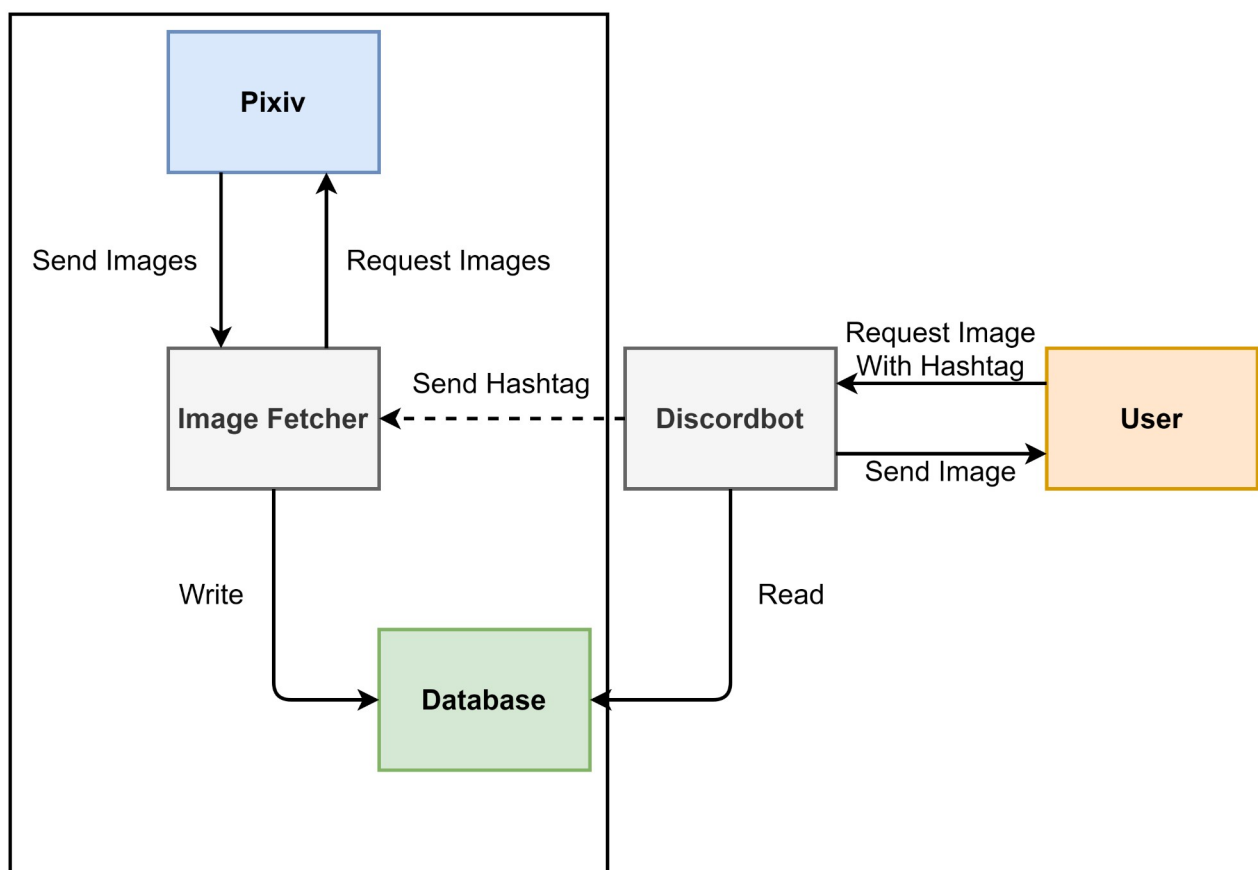


En mí planteamiento y tras informarme vi que lo más lógico para usar en este caso y lo más óptimo venía a ser usar MongoDB como base de datos. Ya que me permite tener una base de datos totalmente gratuita en la nube. Además de ser la más usada para este tipo de proyectos.

## 2.2 Estructura y base de datos

En cuanto a la estructura del Bot inicialmente pensar en como se iba a estructurar en cuanto a la base de datos fue un problema ya que al final cada función que le iba añadiendo al bot podía o no necesitar una tabla para si misma.

Ya que al final la estructura básica de un bot de Discord es la siguiente



### 3. Desarrollo

Esta fase ha sido la más longeva del proyecto ya que en ella es donde he tenido que ir buscando documentación de discord.js, ir desarrollando cada una de las funciones del bot, etc, etc. Donde tras pensar mucho decidí desarrollar las siguientes funcionalidades ya sea en menor o mayor escala

- Sistema de tickets
- Integración con ChatGPT
- Sistema de bienvenida
- Comandos de juegos
- Comandos de utilidad

Una vez tenía claro lo que quería realizar inicialmente con el bot me puse manos a la obra y aquí fue cuando empezaron a surgir los primeros problemas.

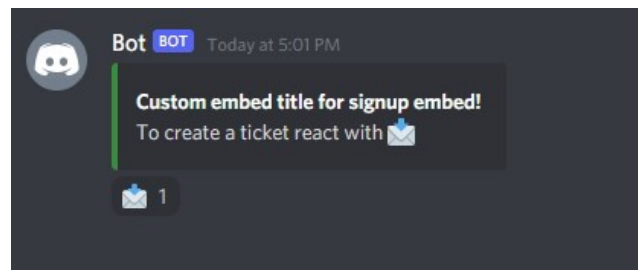
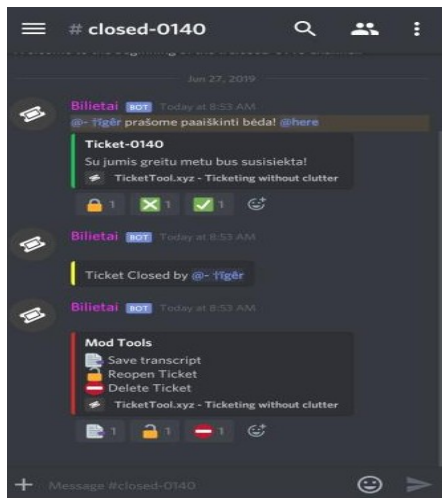
Estos problemas venían en base a discord.js ya que al ser propio de Discord depende únicamente de esta empresa lo cual mientras Discord siga yendo bien con tantos millones de usuarios genial, todo irá hacia delante, pero como deje de tenerlos todo se va al traste.

Además no es el único problema que encontré ya que debía tener especial cuidado con la documentación o información que encontraba ya que discord.js cambia demasiado de una versión a otra en este caso con la versión actual que es la 14 es totalmente diferentes con sus predecesoras por cual si Discord hace cambios durante mi proyecto puede ser que mi bot dejará de funcionar y tuviera que realizar cambios en el arranque y la forma de crear los comandos lo cuál significaría refactorizar casi empezando de 0 en cualquier momento lo que a su vez también me suponía una especie de reto, a la vez que debo de estar en constante información con los cambios realizados por Discord en su propio módulo para poder prevenirlos.

Otro problema que empecé a tener es la versión 14 del propio discord.js era muy reciente por cual todavía su documentación estaba en desarrollo algunas cosas bastante avanzadas, pero para según que cosas me tenía que estar peleando durante horas con la API para a lo mejor sacar la función que realizará X cosa porque tuviera un nombre diferente al habitual.

Entonces tras seguir con mi investigación empecé con las primeras implementaciones con mis propios comandos que realizaban funciones como enviar un link al perfil de la red social, información del servidor, sacar el avatar del usuario, etc.

Luego de tener los primeros comandos más básicos empecé con el desarrollo de los sistemas más complejos como podría ser la creación de un sistema de ticket siendo que mi objetivo era que se pareciese lo más posible a las siguientes imágenes



Una vez logrado este objetivo, empecé con el desarrollo de otras funcionalidades y así progresivamente.

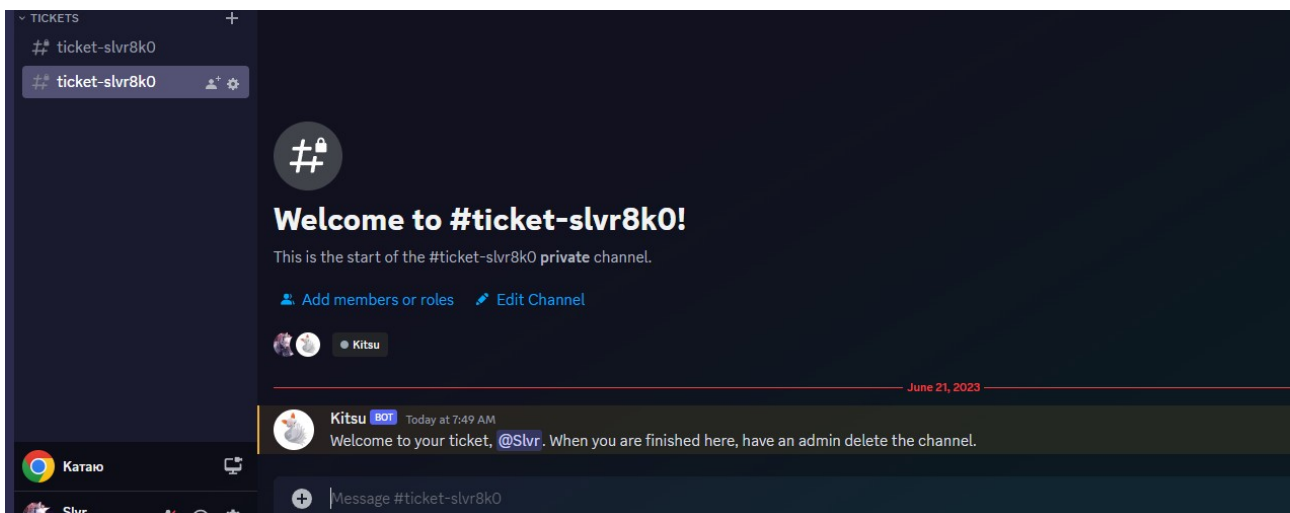
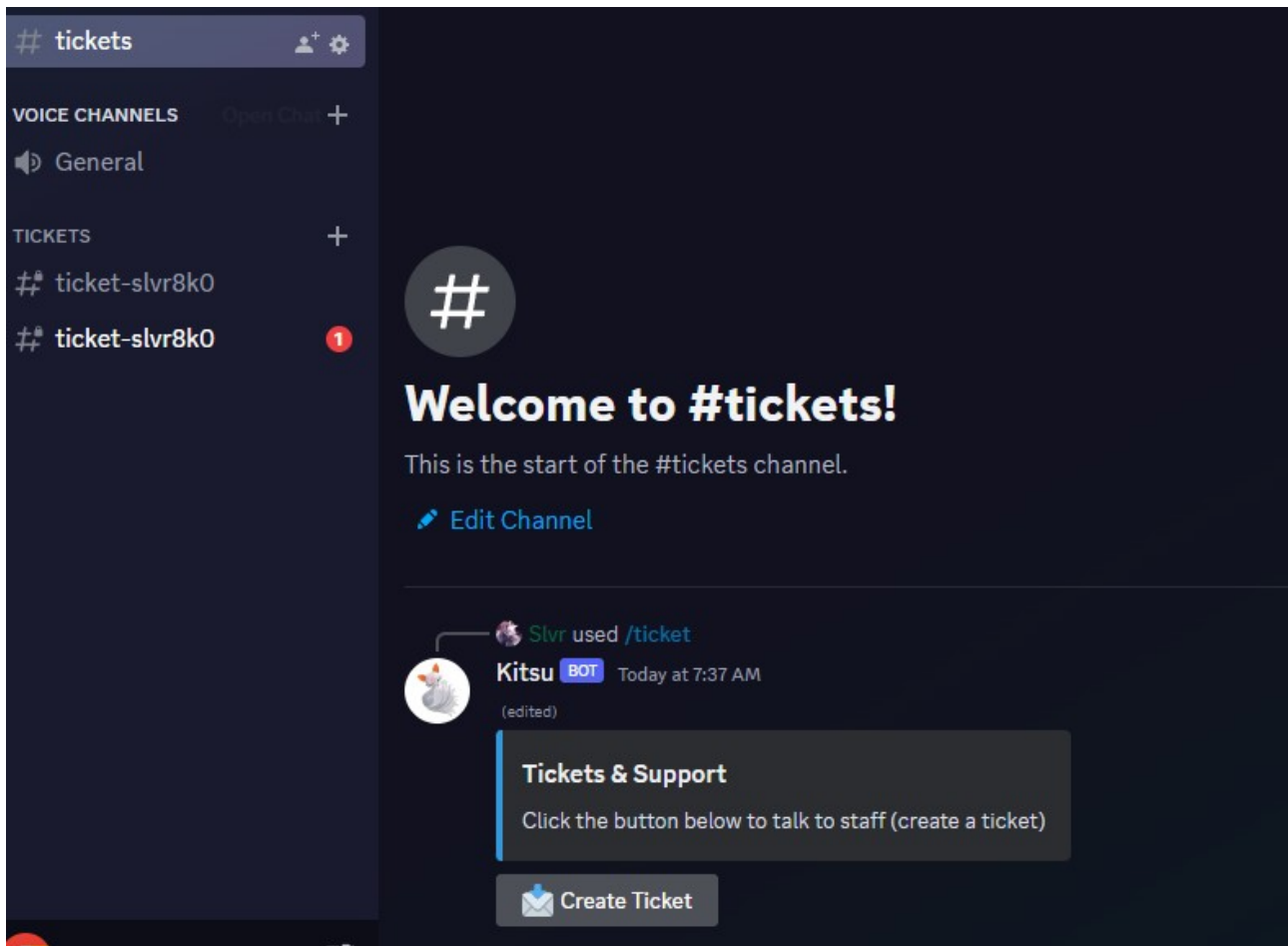
Desgraciadamente mediante iba avanzando en el proyecto me dí cuenta de que debía descartar la creación de la página web del Bot ya que si me ponía a desarrollarla con la escasa documentación que encontraba sobre ese tema me iba a demorar más de lo que había planeado inicialmente. Entonces prioricé el desarrollo del bot a la implementación de la Web del mismo.

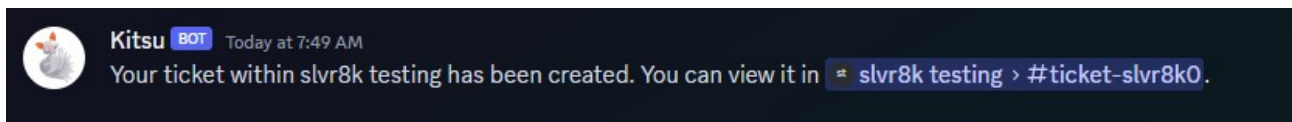
Por lo que actualmente el bot cuenta con las siguientes funcionalidades.

```
-----  
|Funcionalidades |  
-----  
|---- Sistema de ticket  
|---- Sistema de Bienvenida  
|---- Comando para quitar fondo a imágenes  
|---- Comando de chatgpt  
|---- Comandos de juegos (Ahorcado, snake, wordle, bola 8)  
|---- Comando de traducción  
|---- Sistema de bienvenida  
|---- Comando de memes  
|---- Comando creador de anuncios  
|---- Comando de ping  
|---- Comando de avatar  
|---- Comando de chequeador de insignias  
|---- Comandos de redes sociales  
-----
```

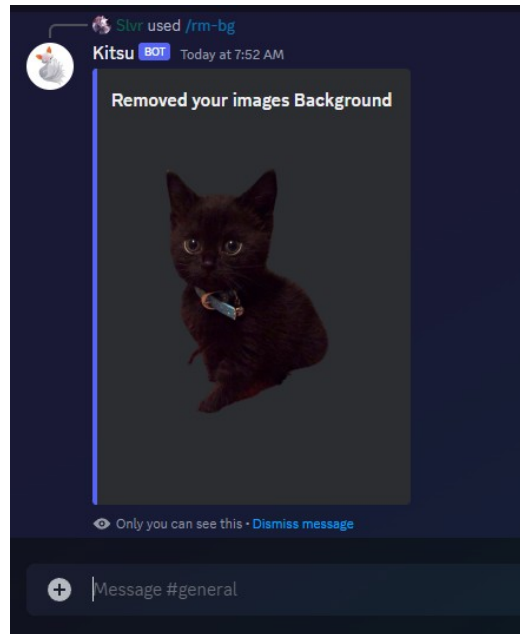
Un ejemplo del funcionamiento de algunas de estas funcionalidades son las siguientes imágenes:

### Sistema de ticket:

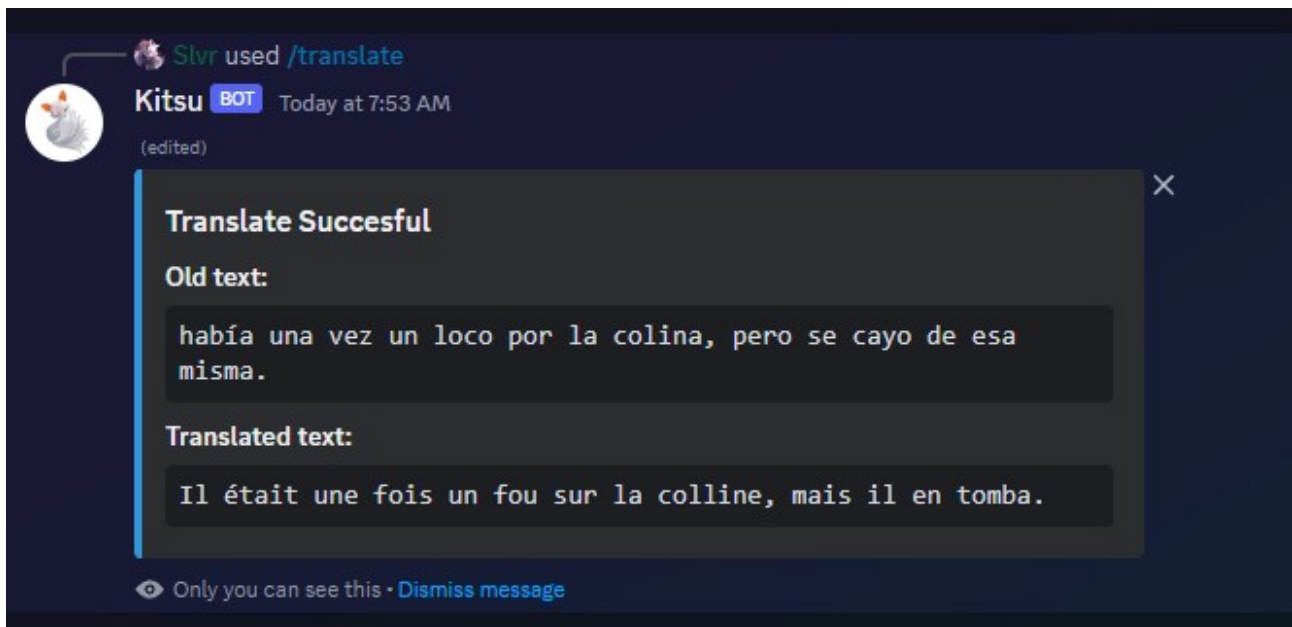




## Removedor de fondo:

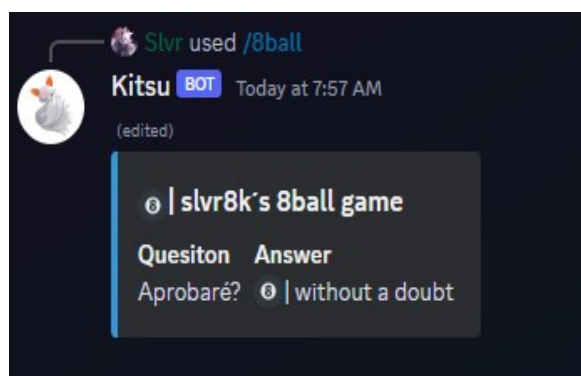
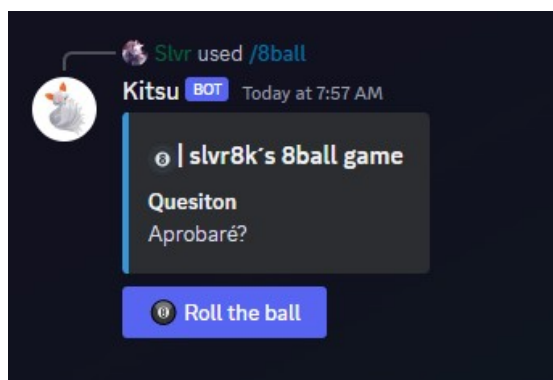
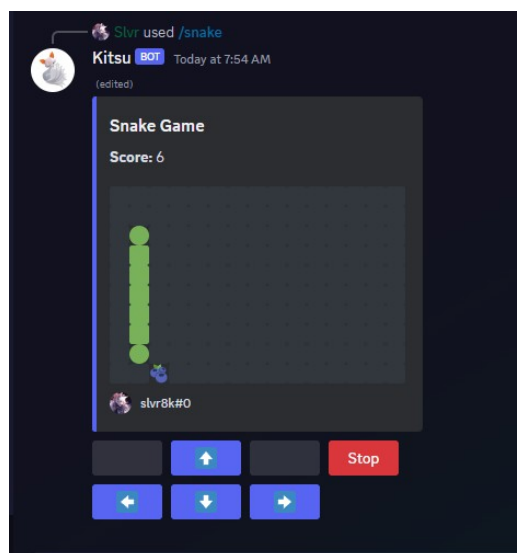
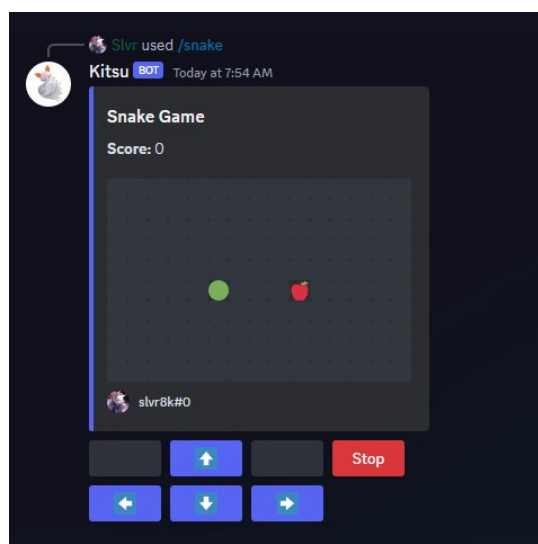
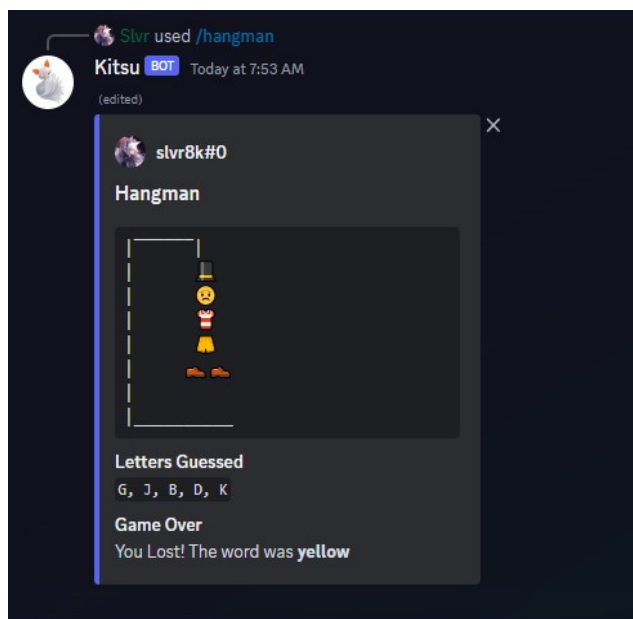
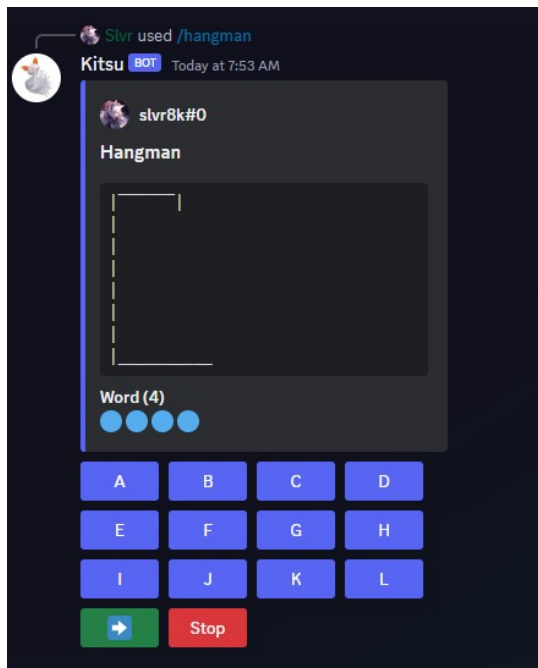


## Traductor:







## Comandos de juegos:




## Comandos sociales:


# general


 <https://twitter.com/slvr8k>


 Slvr used /twitch

 **Kitsu BOT** Today at 7:56 AM  
<https://www.twitch.tv/slvr8k>

Twitch  
[slvr8k - Twitch](#)





 Slvr used /namemc


 **Kitsu BOT** Today at 7:56 AM  
[https://namemc.com/profile/\\_silverX8\\_](https://namemc.com/profile/_silverX8_)

NameMC  
[\\_SilverX8\\_ | Minecraft Profile](#)

Check out \_SilverX8\_'s Minecraft skins, name history, UUID, and much more!




 Slvr used /yt

 **Kitsu BOT** Today at 7:56 AM  
<https://www.youtube.com/@silverx8>

YouTube  
[silverX8](#)

Un simple canal sin más :o  
N c que esperabas x'd



## 4. Mejoras

En este apartado van todas las mejoras que le quiero añadir a este proyecto ya que mi idea es seguir manteniéndolo y desarrollando dentro de este apartado encontramos el siguiente listado con algunas de las mejoras iniciales que he pensado para este proyecto.

1. Creación de la página web
2. Implementación de comandos de moderación (ban, mute, kick, etc)
3. Implementación de sistema de música con la API de Spotify o la propia de YouTube si no monetizo mi bot o no lo contiene la versión de pago con esta última
4. Creador de comandos propios
5. Mejorar rendimiento de los sistemas actuales
6. Reestructuración del bot
7. Sistema de automoderación configurable
8. Sistema de bypass de roles seleccionados

Otra mejora posible es añadir un sistema de monetización al bot, pero para ello debería de crear una versión premium del bot la cual te diera más acceso o más funcionalidades como podría ser acceso a una configuración de los comandos desde un panel de control web lo que permitiría a las personas que tengan mi bot en su servidor poder configurar el bot sin necesidad de estar dentro de Discord o desde el propio móvil ya que se realizaría desde la web del mismo o desde una aplicación móvil que se podría desarrollar. Otra forma de mantener el bot es en base a donaciones.

## 5. Conclusión

Mi conclusión final en base a este proyecto es la siguiente. Es un proyecto bonito de hacer se aprende muchas cosas de JavaScript con él y lo importante de todo es que la propia documentación de incita a llevar buenas prácticas a la hora de escribir código. Pero no es un proyecto que recomendaría hacer ya que las horas que necesitas para buscar documentación que te sirva en ese momento son demasiadas y si o sí necesitas un buen nivel de inglés ya que todo está en inglés si quieres ir más rápido.

Si es verdad que trabajar con este tipo de proyecto te da una perspectiva más única y clara de como es trabajar en una aplicación que se este basando en otra a la vez, lo cual te enseña a estar más al día con las actualizaciones, a estar revisando tu código sobre que mejorar o por si tienes que cambiar algo. Por lo cual te aporta esa constancia de estar refactorizando y mejorando tu propio código.

Otro punto negativo de este proyecto es que te encierra mucho en JavaScript ya que al final solo esta basado en este lenguaje por las tecnologías que necesitas. Lo cual significa que ya te estarías más especializando en ese lenguaje que tiene sus puntos a favor y en contra.