SYNCRONYBOT

Bot multifuncional para Discord con Inteligencia Artificial

Iker Fernandez Rocamora | Alejandro Medrano Martinez
SMX Administració de Sistemes micro Informátics y xarxes · Grup 2C
Institut Puig Castellar · Santa Coloma de Gramenet · 2026

Contenido de la presentación

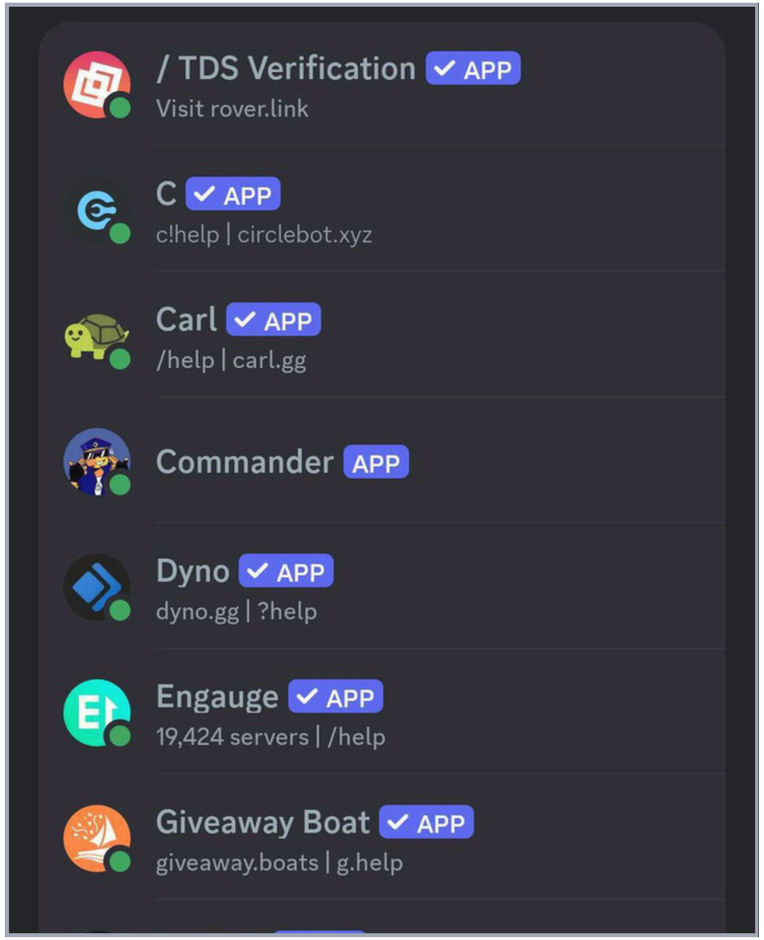
Contexto del proyecto y que problema solucionamos

1. El uso de Discord ha crecido exponencialmente en entornos profesionales y comunidades digitales.

2. Los servidores utilizan múltiples bots independientes para tareas diversas, generando fragmentación e ineficiencia.

3. La moderación de grandes volúmenes de mensajes es un reto creciente para los administradores.

4. Las empresas que ofrecen soporte técnico vía Discord necesitan herramientas integradas.



Objetivos del proyecto

Objetivo general: Desarrollar SyncronyBot, un bot para Discord con funcionalidades avanzadas basadas en IA y automatización.

Resúmenes automáticos

Implementar NLP para condensar conversaciones largas en puntos clave.

Moderación inteligente

Detectar spam, contenido inadecuado y palabras prohibidas.

Sincronización externa

Investigacion rapida y sin errores en paginas como Wikipedia o Fandom.

Traducción en tiempo real

Permitir comunicación fluida en servidores multilingüe.

Interfaz de administración

Panel web para gestionar la configuración sin acceder a Discord.

Despliegue cloud

Garantizar estabilidad y disponibilidad 24/7 con arquitectura robusta.

Metodología de trabajo

Metodología Ágil (Scrum)

- Codigos resumidos, cortos pero los mas pulido posibles.
- Adaptación continua al feedback
- Comunicación fluida entre los dos desarrolladores
- Priorización de funcionalidades por valor

Herramientas utilizadas

Trello

Gestión de tareas y sprint backlog

GitHub

Control de versiones y colaboración

Docker

Contenedores para el despliegue

VS Code

Editor principal de código

Stack tecnológico elegido

Discord.js

Bot principal

Python

Programacion Bot y p.web

Express.js

Panel web admin

MySQL

Base de datos

Docker

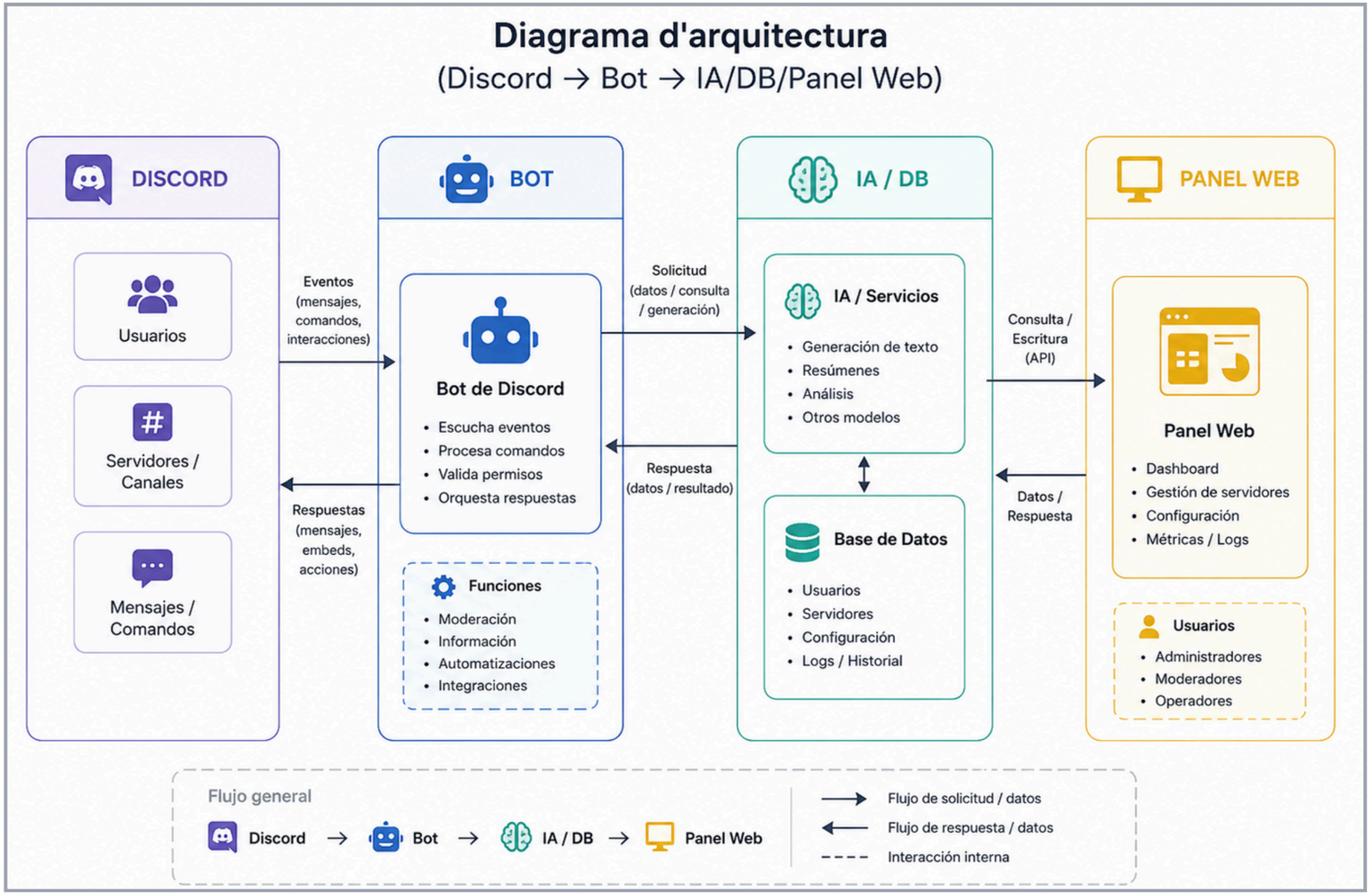
Contenedores

Node.js + pm2

Despliegue 24/7

Arquitectura del sistema

SyncronyBot sigue una arquitectura modular: cada componente tiene una responsabilidad clara y se comunica con los demás a través de API REST.



Bot de Discord

Discord.js

Servicio de IA

Python + spaCy

Base de datos

MySQL

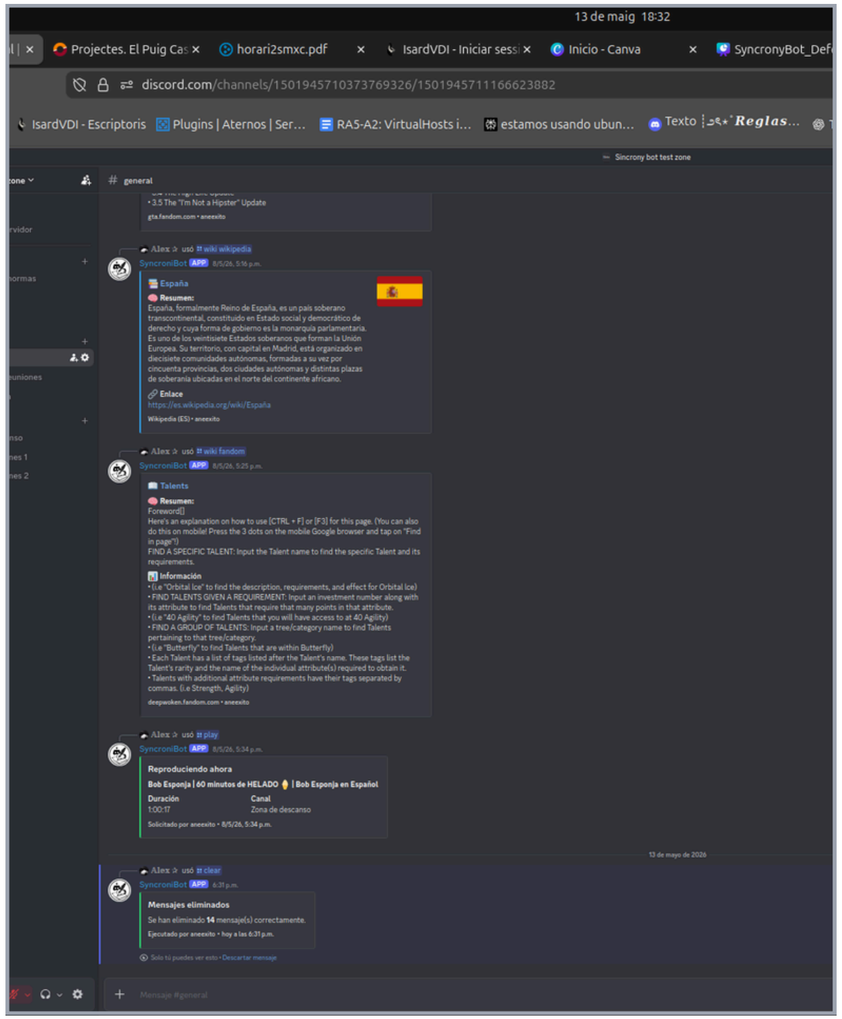
Panel web

Express.js / Discord.dev

Despliegue

Docker + Cloud

Componente 1: Bot de Discord (Discord.js)



Recepción de mensajes y comandos

Detecta y procesa mensajes de usuarios, identificando estructuras y comandos específicos.

Acciones dentro del servidor

Crea tickets, genera resúmenes, publica contenido y gestiona canales.

Integración con la IA

Este envia los datos atraves de una API-Key de IA

Conexión con la base de datos

Guarda configuraciones, preferencias y registros de actividad permitiendo resumir

Moderación automática

Analiza mensajes en tiempo real y actúa según las reglas del servidor especificadas

Componente 2: Servicio de Inteligencia Artificial

Resumen automático de conversaciones

Analiza grandes volúmenes de mensajes y genera un resumen con los puntos más relevantes.

Traducción de mensajes

Traducción automática entre idiomas para mejorar la comunicación en servidores multilingüe.

Moderación inteligente

Detecta spam, contenido inadecuado y lenguaje ofensivo mediante análisis de texto.

Capacidad para investigar entre diferentes paginas

Interpreta las palabras adjuntadas, tras ello usa un inventario de paginas web que tiene almacenadas a su disposicion o aprovecha la capacidad de la IA para encontrarlas.

1	→	2	→	3	→	4	→	5
Usuario envía el comando /resum		Bot recoge los mensajes del canal (Base de datos)		Envío al servicio de IA (API-KEY)		IA analiza los umensajes y hace un resumen		La devuelve y la muestra en el servidor atraves del bot.

Funcionalidad: Investigación Wiki + Fandom

Usuario escribe

/wiki “Tema” “detalles”

Bot envía la respuesta

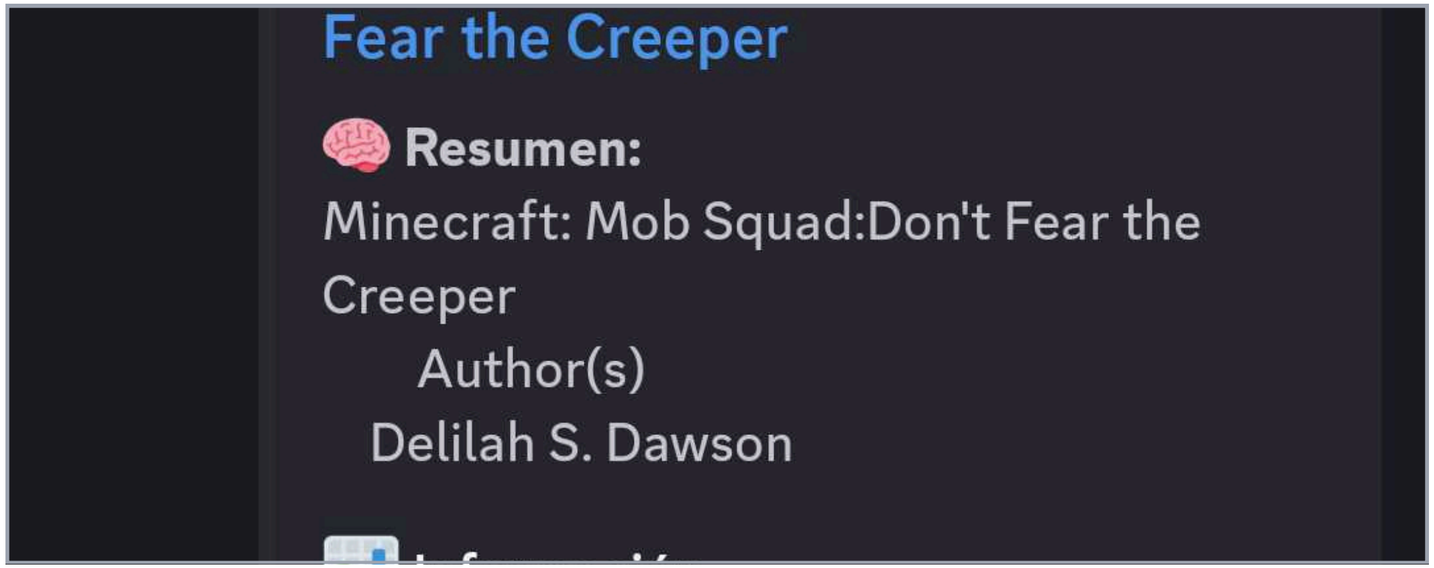
Texto + detalle

Se busca en las páginas para recibir la respuesta

puigcastellar.wikipedia.es

Resultado en el canal

Resumen + enlace de la página



Carateristiques

- Muestra un pre-resumen de la página
- Está enlazado directamente con las páginas
- El tiempo de respuesta es rápido ya que estas paginas estan almacenadas.
- Si tienes faltas ortográficas es capaz de leer igualmente

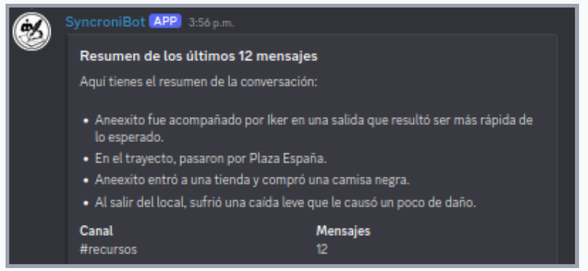
Pruebas y validación del sistema

Prova 1

Resumen de conversaciones

Método: Conversación real de 50+ mensajes enviada al servicio de IA. Se valora, la claridad, la capacidad de resumir y la velocidad

Resultado esperado: Resumen claro generado en < 3 seg. Información principal preservada.

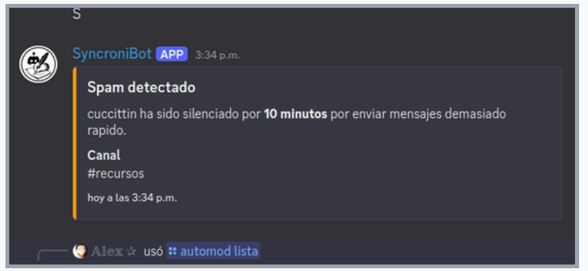


Prova 2

Detección de spam y palabras prohibidas

Método: Mensajes de prueba con spam y contenido inadecuado enviados al canal. El bot los analiza y actúa.

Resultado esperado: Detección > 90% correcta. Pocos falsos positivos. Respuesta rápida.

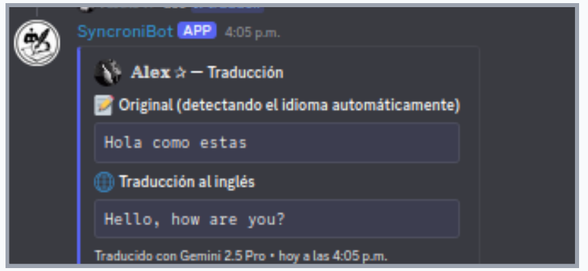


Prova 3

Traducción entre idiomas

Método: Comando /translate ejecutado con diversas parejas de idiomas (ES-EN, CA-ES, EN-FR).

Resultado esperado: Traducciones precisas y rápidas. Tiempo de respuesta cambia dependiendo de la longitud.



Errores encontrados y soluciones aplicadas

Errores del servidor web

ERR_CONNECTION_REFUSED

Node.js no se ejecutaba. Solución: corregir el proceso.

ERR_CONNECTION_TIMED_OUT

El básico Isard solo admite el puerto 80. Solución: proxy inverso.

Sesión no persistente

Cada HTML era independiente. Solución: comprobar el token JWT.

Conflicto Apache/Node puerto 80

Solución: proxy inverso con Apache, Node en el puerto 3000.

Errores del bot

Rate Limits

Demasiadas peticiones rápidas. Solución: cola de mensajes.

Error 4000-4009 (Gateway)

Token caducado. Solución: refresco automático del token.

Compatibilidad dependencias

Versión de Discord.js incompatible. Solución: fijar la versión.

API Key

Una API-Key gratuita sufre de limitaciones

Errores de base de datos y depuración

SyntaxError: Unexpected end of input

Causa: El server.js quedaba incompleto al editarlo. Faltaba el cierre final.

Solución: Revisar el archivo después de cada edición. Usar un linter

Unknown column 'usuarios' in field list

Causa: La columna INSERT se llamaba 'usuarios' pero en la tabla era 'nombre'.

Solución: Revisar el nombre exacto de las columnas en la DB y sincronizar el código.

Unknown column 'gmail' in field list

Causa: El campo INSERT decía 'gmail' pero en la tabla era 'email'.

Solucion: Asegurar la coherencia entre el esquema de la base de datos y las consultas SQL.

Cannot find module 'mysql2'

Causa: El paquete mysql2 no estaba instalado en el servidor.

Solución: Ejecutar npm install mysql2 y verificar el package.json.

Página web: estructura y páginas

La web de SyncronyBot es una plataforma completa con login, dashboard, documentación y soporte — construida con Express.js y MySQL.

Inicio

index.html

Presentación general de SyncroniBot, explicación rápida de sus funciones principales.

El Proyecto

proyecto.html

Información sobre cómo nació el proyecto, objetivos principales y tecnologías utilizadas durante el desarrollo.

Características

caracteristicas.html

Apartado donde se muestran las funciones más importantes del bot, como moderación, resúmenes automáticos y soporte para Discord.

Donaciones

donaciones.html

Sección para apoyar el proyecto y ayudar a mantener el desarrollo y las futuras mejoras de SyncroniBot.

Contacto

contacto.html

Formulario y métodos de contacto para resolver dudas, enviar sugerencias o comunicar problemas relacionados con la web o el bot.

Funcionalidades de la plataforma web

Sistema de autenticación

- Login y registro con correo/nombre de usuario/contraseña
- Sesiones persistentes con auth.js

Backend (server.js)

- API REST con Node.js y Express
- Base de datos MySQL (usuarios, configuración, logs)
- Proxy inverso Apache → Node puerto 3000

Tecnologías usadas

- HTML5, CSS3 y JavaScript vanilla para el frontend
- Node.js + Express para el backend
- MySQL para la base de datos
- Apache como proxy inverso hacia el puerto 3000

Intregaciones

Integraciones

- Autenticación social con Discord OAuth y Google planificada como alternativa al registro manual aunque al final no se ha implementado, pero es algo que esta a punto de salir a la luz y es algo que esta claro para desarrollarse en el futuro.

- Página de donaciones dedicada para apoyar el mantenimiento y desarrollo continuo del proyecto

Funciones

- Traductor
- Formulario de contacto
- Navegación adaptada para móvil
- Menú de usuario con cerrar sesión
- Sistema de donaciones
- Modo oscuro

Conclusiones del proyecto

SyncronyBot centraliza múltiples utilidades en un solo bot, mejorando la experiencia de usuario en Discord y reduciendo la carga operativa de los administradores.

Objetivos alcanzados

El bot integra IA, moderación, traducción y automatización en una sola herramienta. La mayoria de objetivos han sido cumplidos sin ningun problema.

Aprendizaje tecnológico

El proyecto ha permitido profundizar en Node.js, Python, MySQL, Docker y despliegue de procesos, herramientas reales de uso profesional.

Metodología válida

El enfoque Scrum ha facilitado la adaptación continua y la comunicación entre el equipo. Discord y GitHub han sido herramientas indispensables.

Limitaciones detectadas

Algunas funciones dependen de modelos de IA externos con límites de uso. El panel web requiere de mucho mas trabajo para poder aplicarse al bot de forma publica.

Visión de futuro

Panel web completo

Desarrollar un panel web para el publico donde sean capaces de gestionar y colocar sus propias reglas y personalizacion.

Modelos de IA más avanzados

Integrar mas de 1 solo modelo de IA que cumpla mas de una sola funcion. Como seria IA generativa o IA de investigacion

Nuevas integraciones

Añadir conectores con Twitter/X, YouTube, Twitch y he instragram otras plataformas de contenido.

Sistema de tickets avanzados

Mejorar el flujo de soporte técnico de forma que el bot sea capaz de gestionar automaticamente los tickets con la IA

App móvil de administración

Permitir a los administradores gestionar el bot desde el móvil en tiempo real.

Multiservidor con suscripciones

Monetizar el bot ofreciendo planes de servicio a comunidades y empresas dependiendo del modelo IA que quieran utilizar.

Gracias por vuestra atención

Turno de preguntas

Iker Fernandez Rocamora
Alejandro Medrano Martinez
CFGM · Institut Puig Castellar · 2025