



Institut Puig Castellar  
Projecte / Crèdit de síntesi  
Document funcional

# DOCUMENTO FUNCIONAL DEL PROYECTO

ALUMNO/GRUPO: Iker fernandez rocamora, alejandro medano martinez

---

## 1. Introducción y contexto

Se desarrollará un bot multifuncional para Discord que integre herramientas de inteligencia artificial con el fin de mejorar la organización y la eficiencia en la comunicación dentro de los servidores. Este bot será capaz de resumir conversaciones importantes, generar respuestas automáticas, realizar búsquedas de información y moderar mensajes mediante la detección de spam o lenguaje inapropiado. Además, permitirá publicar automáticamente imágenes o publicaciones de Instagram en canales determinados, crear encuestas y ofrecer un sistema de traducción integrado que facilite la interacción entre usuarios de diferentes idiomas. El propósito principal del proyecto es unificar en un solo bot funciones que generalmente requieren el uso de varios, brindando así una experiencia más cómoda, funcional y accesible para las comunidades que utilizan Discord como medio principal de comunicación.

## 2. Análisis de requisitos

### 2.1. Requisitos funcionales (RF)

Código	Descripción del requisito funcional
RF1	El bot permitirá registrar y autenticar a los servidores en los que se instalará, configurando permisos y canales específicos para su funcionamiento.

Código	Descripción del requisito funcional
RF1	El bot permitirá registrar y autenticar a los servidores en los que se instalará, configurando permisos y canales específicos para su funcionamiento.
RF2	El sistema resumirá automáticamente conversaciones o mensajes importantes, incluyendo los fijados o de canales seleccionados.
RF3	El bot ofrecerá un buscador interno que permitirá localizar mensajes, archivos o temas específicos dentro del servidor.
RF4	El sistema realizará moderación inteligente, detectando y gestionando mensajes con spam, lenguaje inapropiado o contenido no permitido.
RF5	El bot permitirá la creación y gestión de encuestas dentro de los canales de texto para la toma de decisiones de la comunidad.
RF6	El sistema publicará de manera automática imágenes o publicaciones de Instagram en los canales designados.
RF7	El bot integrará un sistema de traducción automática de mensajes en tiempo real para facilitar la comunicación entre usuarios de diferentes idiomas.
RF8	El sistema permitirá a los administradores personalizar la configuración del bot, incluyendo comandos, filtros, idioma y funciones activas.

RF9	El bot podrá generar resúmenes semanales o diarios de la actividad del servidor según la configuración del usuario.
RF10	El sistema ofrecerá respuestas automáticas e información contextual mediante inteligencia artificial para apoyar la gestión del servidor.

## 2.2. Requisitos no funcionales (RNF)

Código	Descripción del requisito no funcional
RNF1	El sistema deberá funcionar de manera estable en servidores de distintos tamaños, garantizando rendimiento óptimo incluso con alto tráfico de mensajes.
RNF2	Las respuestas del bot deberán generarse en menos de tres segundos desde la recepción del comando o solicitud.
RNF3	El sistema deberá ser compatible con versiones actuales de Discord y adaptarse a futuras actualizaciones mediante mantenimiento periódico.
RNF4	Los datos de los usuarios y servidores deberán manejarse de forma segura, cumpliendo con las políticas de privacidad y protección de información.
RNF5	El sistema permitirá una configuración intuitiva a través de comandos simples y menús accesibles, facilitando su uso a administradores y usuarios.
RNF6	El bot deberá tener una arquitectura modular para permitir la integración de nuevas funciones sin afectar su rendimiento general.
RNF7	La interfaz o panel de control del bot deberá ser accesible desde navegadores y dispositivos móviles, manteniendo una experiencia uniforme.
RNF8	El sistema deberá contar con tolerancia a fallos y capacidad de reinicio automático en caso de errores o desconexiones.
RNF9	El procesamiento de IA deberá realizarse de manera eficiente, optimizando el uso de recursos del servidor donde se ejecute el bot.
RNF10	El sistema deberá ofrecer soporte multilingüe tanto en comandos como en mensajes de respuesta.

## 2.3. Restricciones

*Condiciones o limitaciones del proyecto.*

- Lenguajes o tecnologías obligatorias:
- Recursos disponibles (tiempo, equipo, materiales):
- Dependencias o limitaciones técnicas:

### **Lenguajes o tecnologías obligatorias:**

El bot se desarrollará en Python con la librería discord.py, utilizando la API de OpenAI para la inteligencia artificial y SQLite o PostgreSQL como base de datos.

### **Recursos disponibles:**

El proyecto contará con un equipo de tres desarrolladores y un diseñador, trabajará durante ocho a diez semanas usando GitHub, Trello y Discord como herramientas principales.

### **Dependencias o limitaciones técnicas:**

Dependerá de la API de Discord y del servicio de IA externo. Su rendimiento variará según la capacidad del servidor y deberá adaptarse a cambios en las políticas o actualizaciones de Discord.

### 3. Análisis de usuarios y roles

Rol	Descripción	Permisos principales
Administrador	Gestiona usuarios y recursos.	Alta, baja y modificación de datos.
Usuario	Utiliza el sistema para realizar acciones básicas.	Crear, consultar y modificar sus propios datos.
Visitante	Consulta información sin interactuar con el sistema	Sin permisos

Rol	Descripción	Permisos principales
Administrador	Es la persona encargada de gestionar el bot dentro del servidor de Discord. Configura los canales, activa o desactiva funciones y supervisa el comportamiento general del sistema.	Puede administrar usuarios, configurar comandos, gestionar permisos, moderar mensajes y acceder a todas las funciones del bot.
Usuario	Son los miembros del servidor que interactúan con el bot para aprovechar sus herramientas. Pueden realizar consultas, participar en encuestas y hacer uso de sistemas como traducción o resúmenes.	Puede usar comandos básicos, consultar información, crear encuestas o solicitar resúmenes, y reportar contenido inapropiado.
Visitante	Personas que recién se unen al servidor o tienen permisos limitados. Solo pueden observar y recibir mensajes de ciertas funciones del bot.	Puede ver resúmenes, mensajes o encuestas públicas, pero no puede ejecutar comandos ni cambiar configuraciones.

## 4. Casos de uso / Escenarios de uso

Código	Nombre del caso de uso	Actor principal	Descripción	Resultado esperado
CU1	Registrar usuario	Visitante	El usuario introduce sus datos y se crea una cuenta.	Usuario registrado correctamente.
CU2	Iniciar sesión	Usuario	Introduce credenciales y accede al sistema.	Acceso concedido.
CU3	Realizar una reserva	Usuario	Selecciona día y hora para reservar un recurso.	Reserva creada.

Código	Nombre del caso de uso	Actor principal	Descripción	Resultado esperado
CU1	Registrar servidor	Administrador	El administrador configura el bot por primera vez en el servidor, estableciendo permisos y canales.	Bot configurado correctamente en el servidor.
CU2	Resumir conversaciones	Usuario	El bot genera un resumen automático de los mensajes importantes de un canal o conversación.	Resumen generado y disponible para consulta.
CU3	Realizar búsqueda	Usuario	El usuario busca mensajes o información específica dentro del servidor a través del bot.	Resultados relevantes mostrados al usuario.
CU4	Crear encuesta	Usuario	El usuario crea una encuesta en un canal para recolectar opiniones de la comunidad.	Encuesta publicada y accesible para votación.

<b>CU5</b>	Moderar mensajes	Administrador	El bot detecta y elimina mensajes que contienen spam o palabras prohibidas automáticamente.	Mensajes inapropiados eliminados y usuario moderado.
------------	------------------	---------------	---	--

## 5. Modelo de datos o estructura de la información

*Objetivo:* representar la información que gestionará el sistema.\*

Incluye las entidades principales (tablas u objetos) y las relaciones entre ellas.

Ejemplo:

- Entidades: Usuario, Reserva, Recurso.
- Relaciones: un usuario puede tener varias reservas, cada reserva corresponde a un recurso.

### Entidades principales

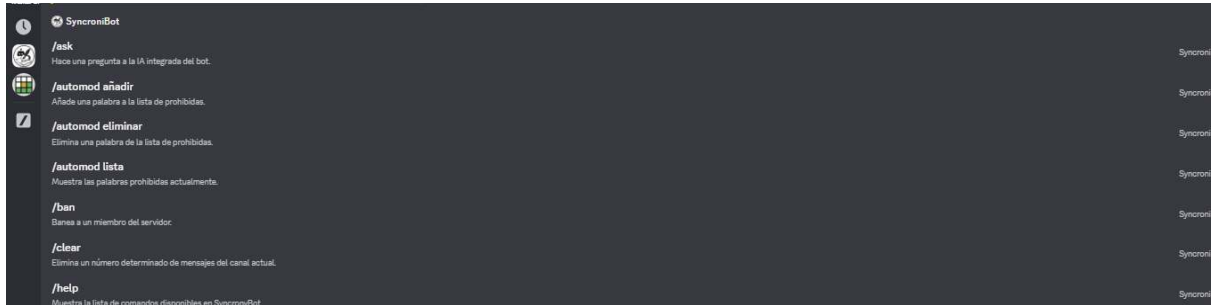
- Servidor: Representa cada comunidad donde el bot está instalado.
- Usuario: Miembro del servidor que interactúa con el bot.
- Mensaje: Mensajes gestionados o procesados por el bot (resumidos, moderados o traducidos).
- Encuesta: Elementos creados por los usuarios para recolectar opiniones de la comunidad.
- Configuración: Preferencias del servidor y permisos del bot.

### Relaciones

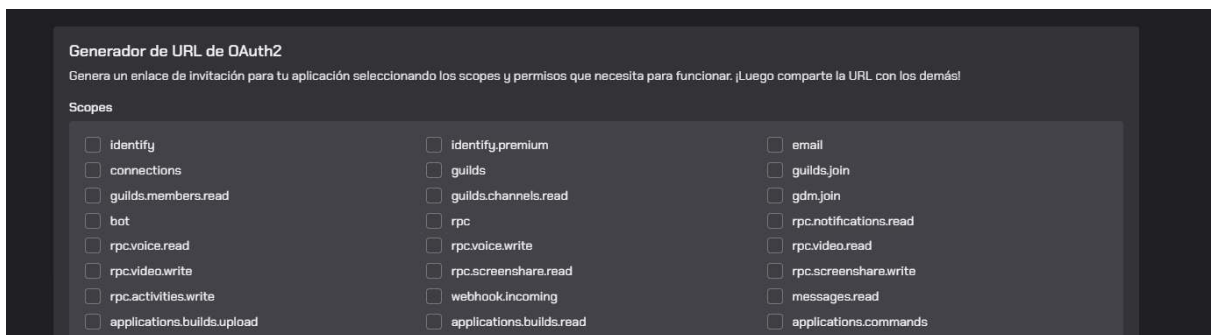
- Un servidor tiene muchos usuarios.
- Un usuario puede crear muchos mensajes y muchas encuestas.
- Un mensaje pertenece a un usuario y a un servidor.
- Una encuesta pertenece a un servidor y puede tener muchas respuestas de usuarios.
- Un servidor tiene una configuración propia.

## 6. Diseño de la interfaz

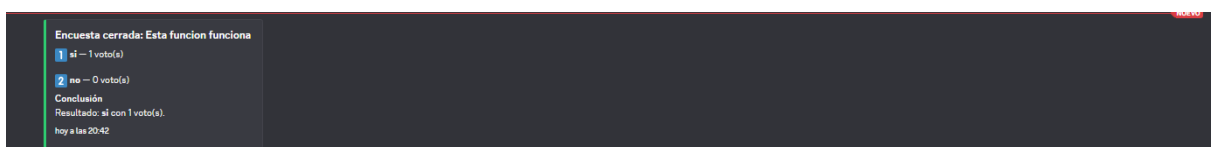
- Pantalla principal (Discord): Muestra comandos, consultas y resúmenes directamente, permitiendo la interacción rápida de usuarios.



- Pantalla de configuración (web): Administradores ajustan funciones y permisos del bot para el servidor.



- Pantalla de encuestas y resultados: Permite crear encuestas y ver los resultados dentro de Discord.



## 7. Planificación técnica

- Lenguajes y frameworks: Python con la librería discord.py para la integración con Discord, y uso de APIs de IA (como OpenAI) para funciones de resumen, traducción y moderación inteligente.
- Base de datos: SQLite o PostgreSQL para almacenar datos de usuarios, mensajes, encuestas y configuraciones de servidores.
- Herramientas de diseño y edición: GitHub para control de versiones; Trello o similar para gestión de tareas; editores de código como VSCode o PyCharm; diseño visual básico para la interfaz web si es requerida (Flask/FastAPI para backend).
- Reparto de tareas: Un equipo pequeño (3-4 personas), con roles asignados a backend (bot y base de datos), IA e integración, y frontend (panel web/configuración).
- Cronograma: Planificación de 8 a 10 semanas con fases: análisis y diseño (1-2 semanas), desarrollo (4-5 semanas), pruebas y ajustes (2 semanas), y despliegue/documentación (1 semana).

## 8. Análisis de riesgos

- Riesgo de ataques o exploits si el bot no está bien protegido; se mitigará con control de accesos y actualizaciones constantes.
- Posible exposición de datos sensibles; se usará cifrado y restricciones de acceso.
- Abuso o spam en el bot; se implementarán filtros automáticos y moderación.
- Errores o respuestas no deseadas de la inteligencia artificial; se supervisará y corregirá manualmente.
- Dependencia de servicios externos como la API de Discord o IA, con planes para mitigar interrupciones.
- Sobrecarga o mal uso que pueda afectar rendimiento; se aplicarán límites y optimización.

### 8.1. Identificación de riesgos

- Falta de tiempo o mala planificación que afecte la entrega del proyecto.
- Problemas técnicos o incompatibilidades con la API de Discord o las librerías usadas.
- Pérdida de datos importantes por fallos en la base de datos o errores en el código.
- Abandono o baja disponibilidad de miembros del equipo durante el desarrollo.
- Cambios en los requisitos que pueden requerir reajustes o reescritura de funciones.

## 8.2. Valoración y respuesta

Riesgo	Probabilidad	Impacto	Plan de prevención o contingencia
Falta de tiempo	Alta	Alta	Dividir tareas y fijar entregas intermedias.
Problemas técnicos	Media	Media	Probar tecnologías antes de programar.
Pérdida de datos	Baja	Alta	Hacer copias de seguridad semanales.

Riesgo	Probabilidad	Impacto	Plan de prevención o contingencia
Falta de tiempo	Alta	Alta	Dividir tareas, fijar entregas intermedias y seguimiento constante.
Problemas técnicos	Media	Media	Probar tecnologías antes de la implementación y mantener comunicación fluida.
Pérdida de datos	Baja	Alta	Realizar copias de seguridad periódicas y usar almacenamiento robusto.
Abandono de miembros	Media	Alta	Mantener motivación, documentar progresos y repartir tareas de forma equilibrada.
Cambios en requisitos	Media	Media	Aplicar metodologías ágiles para adaptarse y validar cambios con el equipo.

## 9. Validación y criterios de éxito

- Criterios de aceptación:

El bot debe registrar usuarios y configurarse correctamente, responder a comandos en menos de 3 segundos, mostrar resúmenes y resultados de búsqueda, permitir creación de encuestas, y aplicar moderación eficaz. Además, debe funcionar establemente en distintos servidores y ser accesible desde dispositivos móviles.

- Pruebas previstas:

Pruebas funcionales para verificar cada comando y función principal, pruebas de usuario para valorar usabilidad y experiencia, y pruebas de rendimiento para asegurar tiempos de respuesta y estabilidad bajo carga.

- Indicadores de calidad:

Para asegurar la calidad del bot en Discord, se busca que funcione de manera confiable, respondiendo rápido y sin errores frecuentes. También es importante que los usuarios estén satisfechos con su experiencia y que el servicio esté disponible sin interrupciones. Estos aspectos garantizan que el bot sea útil y efectivo para quienes lo usan.

## 10. Conclusión

Las principales decisiones del análisis funcional incluyen:

- El sistema tendrá funciones para gestionar usuarios, resumir y buscar mensajes, crear encuestas y moderar contenido en servidores de Discord.
- Se utilizarán tecnologías como Python con discord.py, bases de datos SQL, APIs de inteligencia artificial y herramientas de gestión de código y tareas.
- El proyecto aportará valor al mejorar la administración y la experiencia de usuarios en Discord con un bot eficiente y personalizado.
- Próximos pasos: preparar el entorno de desarrollo, crear estructura inicial del repositorio y base de datos, y comenzar con la implementación de los casos de uso prioritarios.
- El análisis funcional proporciona una visión clara y realista del proyecto. A partir de aquí, comienza la fase de desarrollo.

El análisis funcional es un paso fundamental que permite comprender a fondo qué debe hacer el sistema y cómo debe comportarse, sentando las bases para un desarrollo organizado y alineado con las necesidades reales del usuario. Proporciona una guía clara y realista, minimizando riesgos y asegurando que el producto final cumpla con las expectativas. Con esta visión definida, comienza la fase de desarrollo, que convertirá en realidad las soluciones planificadas. Finaliza con una breve reflexión o frase de cierre, por ejemplo: