# **📋 Documentación del Sistema - Chatbot INGE LEAN S.A.S.**

## **🏢 Información General**

**Empresa:** INGE LEAN S.A.S.

**Ubicación:** Pereira, Risaralda, Colombia

**Año de fundación:** 2013

**Sistema:** Chatbot de atención al cliente basado en IA

**Fecha de documentación:** Julio 2025

## **📖 Descripción del Sistema**

El sistema de chatbot de INGE LEAN S.A.S. es una aplicación movil desarrollada en App inventor, e implementación de Python con Flask que utiliza la API de Google Gemini para proporcionar respuestas automáticas a consultas frecuentes sobre los servicios de la empresa.

### **🎯 Objetivos del Sistema**

* Automatizar la atención de consultas frecuentes
* Proporcionar información inmediata sobre servicios de la empresa
* Reducir la carga de trabajo del equipo de atención al cliente
* Mejorar la experiencia del usuario con respuestas 24/7

## **️ Arquitectura del Sistema**

### **Componentes Principales**

┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐  
│ Frontend │───▶│ Flask API │───▶│ Google Gemini │  
│ (Cliente) │ │ (Servidor) │ │ (IA) │  
└─────────────────┘ └─────────────────┘ └─────────────────┘  
 │  
 ▼  
 ┌─────────────────┐  
 │ Registro CSV │  
 │ (Histórial) │  
 └─────────────────┘

### **Tecnologías Utilizadas**

* **Backend:** Python 3.11+
* **Framework Web:** Flask 2.3+
* **IA:** Google Gemini API (gemini-2.0-flash)
* **CORS:** Flask-CORS
* **Almacenamiento:** CSV para logs
* **Formato de datos:** JSON
* **Codificación:** UTF-8

## **📁 Estructura del Proyecto**

chatbots/  
├── server.py # Servidor principal Flask  
├── context.txt # Contexto del chatbot  
├── requirements.txt # Dependencias Python  
├── registro\_chat.csv # Log de conversaciones  
├── gemini\_chat\_contexto.py # Script auxiliar (opcional)  
├── hola.py # Script de prueba (opcional)  
└── modelo.pkl # Modelo guardado (opcional)

## **⚙️ Instalación y Configuración**

### **Requisitos del Sistema**

* Python 3.11 o superior
* Conexión a internet
* API Key de Google Gemini
* Sistema operativo: Windows/Linux/macOS

### **Proceso de Instalación**

1. **Clonar o descargar el proyecto**

https://github.com/Valentinavvd/Entrega\_Final.git

1. **Instalar dependencias**

pip install flask flask-cors google-generativeai

1. **Configurar API Key**
   1. Obtener API Key desde [Google AI Studio](https://aistudio.google.com/)
   2. Reemplazar en server.py:

genai.configure(api\_key="TU\_API\_KEY\_AQUI")

1. **Verificar archivo de contexto**
   1. Asegurar que context.txt existe con información de la empresa

## **🚀 Ejecución del Sistema**

### **Iniciar el Servidor**

python server.py

### **Salida Esperada**

🚀 INGE LEAN S.A.S. - Chatbot Server  
✅ Modelo: gemini-2.0-flash (más reciente)  
⏱️ Rate limit: 5 segundos entre requests  
🔗 Endpoints: /chatbot, /health, /test  
💡 Usa /test para probar sin consumir cuota  
 \* Running on <http://127.0.0.1:5000>

## **🔌 API Endpoints**

### **1. Endpoint Principal - Chatbot**

**URL:** POST /chatbot

**Descripción:** Procesa preguntas y devuelve respuestas del chatbot

#### **Request**

{  
 "pregunta": "¿Qué servicios ofrece INGE LEAN S.A.S.?"  
}

#### **Response Exitosa (200)**

{  
 "respuesta": "INGE LEAN S.A.S. ofrece servicios de software, hardware, automatización industrial, inteligencia artificial y mantenimiento..."  
}

#### **Response Error de Cuota (429)**

{  
 "error": "Cuota API excedida. Intenta más tarde.",  
 "solucion": "Espera unas horas o considera un plan de pago."  
}

### **2. Endpoint de Salud**

**URL:** GET /health

**Descripción:** Verifica el estado del servidor

#### **Response**

{  
 "status": "OK",  
 "modelo": "gemini-2.0-flash",  
 "rate\_limit": "5 segundos",  
 "info": "INGE LEAN S.A.S. - Chatbot v2.0"  
}

### **3. Endpoint de Prueba**

**URL:** GET /test

**Descripción:** Respuesta de prueba sin consumir cuota de IA

#### **Response**

{  
 "respuesta": "¡Hola! Soy el asistente de INGE LEAN S.A.S. ¿En qué puedo ayudarte?",  
 "empresa": "INGE LEAN S.A.S.",  
 "ubicacion": "Pereira, Risaralda",  
 "servicios": ["Software", "Hardware", "Automatización", "IA", "Mantenimiento"]  
}...

## **🛠️ Configuración Avanzada**

### **Parámetros del Modelo IA**

generation\_config = {  
 "max\_output\_tokens": 200, # Máximo de tokens en respuesta  
 "temperature": 0.5, # Creatividad (0.0-1.0)  
}

### **Rate Limiting**

* **Intervalo mínimo:** 5 segundos entre requests
* **Propósito:** Evitar exceder cuotas de la API
* **Configurable en:** Variable MIN\_INTERVALO

### **Logging de Conversaciones**

* **Archivo:** registro\_chat.csv
* **Campos:** Timestamp, Pregunta, Respuesta
* **Codificación:** UTF-8

## **🧪 Pruebas del Sistema**

### **Usando Postman**

1. **Configurar Postman**
   1. URL: <http://127.0.0.1:5000/chatbot>
   2. Método: POST
   3. Headers: Content-Type: application/json
2. **JSON de prueba**

{  
 "pregunta": "¿Cuáles son los servicios de INGE LEAN?"  
}

### **Usando cURL**

curl -X POST <http://127.0.0.1:5000/chatbot> \  
 -H "Content-Type: application/json" \  
 -d '{"pregunta": "¿Qué servicios ofrecen?"}'

### **Casos de Prueba Recomendados**

|  |  |  |
| --- | --- | --- |
| **Caso** | **Pregunta** | **Resultado Esperado** |
| Servicios | "¿Qué servicios ofrecen?" | Lista de servicios |
| Ubicación | "¿Dónde están ubicados?" | Pereira, Risaralda |
| Historia | "¿Desde cuándo operan?" | Desde 2013 |
| Contacto | "¿Cómo puedo contactarlos?" | Información de contacto |

## **⚠️ Troubleshooting**

### **Problemas Comunes**

#### **Error 404 - Modelo no encontrado**

**Causa:** Modelo de IA obsoleto

**Solución:** Actualizar a gemini-2.0-flash

#### **Error 429 - Cuota excedida**

**Causa:** Límite de API alcanzado

**Solución:**

* Esperar reset de cuota
* Implementar rate limiting
* Considerar plan de pago

#### **Error 500 - Error interno**

**Causa:** Problema en el código o configuración

**Solución:**

* Verificar API Key
* Revisar logs del servidor
* Validar archivo context.txt

#### **Conexión rechazada**

**Causa:** Servidor no iniciado o puerto ocupado

**Solución:**

* Verificar que el servidor esté corriendo
* Cambiar puerto si es necesario

## **Monitoreo y Métricas**

### **Logs del Sistema**

* **Ubicación:** Consola del servidor
* **Incluye:** Requests, errores, timestamps
* **Rotación:** Manual

### **Métricas Importantes**

* Número de requests por minuto
* Tiempo de respuesta promedio
* Tasa de errores
* Uso de cuota de API

## **🔒 Seguridad**

### **Buenas Prácticas Implementadas**

* Rate limiting para prevenir abuso
* Validación de entrada de datos
* Manejo seguro de errores
* Logs de auditoría

### **Recomendaciones Adicionales**

* Usar variables de entorno para API keys
* Implementar autenticación en producción
* Configurar HTTPS
* Limitar acceso por IP si es necesario

## **🚀 Despliegue en Producción**

### **Consideraciones**

1. **Servidor WSGI:** Usar Gunicorn en lugar del servidor de desarrollo
2. **Proxy Reverso:** Nginx para servir archivos estáticos
3. **Variables de Entorno:** Para configuraciones sensibles
4. **Logging:** Sistema de logs estructurado
5. **Monitoreo:** Herramientas de monitoreo en tiempo real

### **Ejemplo de Despliegue**

# Instalar Gunicorn  
pip install gunicorn  
  
# Ejecutar en producción  
gunicorn -w 4 -b 0.0.0.0:5000 server:app

## **📝 Mantenimiento**

### **Tareas Regulares**

* **Diario:** Revisar logs de errores
* **Semanal:** Verificar uso de cuota API
* **Mensual:** Actualizar dependencias
* **Trimestral:** Revisar y actualizar contexto

### **Actualizaciones**

* Seguir actualizaciones de Google Gemini API
* Actualizar dependencias de Python
* Revisar y mejorar contexto del chatbot

## **👥 Información de Contacto**

**Desarrolladores: Dahiana Osorio, Valentina Vargas, María Alejandra Gonzales**

**Fecha de última actualización:** Julio 2025

## **📄 Licencia y Términos**

Este sistema fue desarrollado específicamente para INGE LEAN S.A.S. y está sujeto a los términos de uso de:

* Google Gemini API
* Flask Framework
* Políticas internas de la empresa

*Documentación v2.0 - Sistema de Chatbot INGE LEAN S.A.S.*