Bitácora Semanal – Estancia Profesional ESIT

BITÁCORA DEL PROYECTO

Fase: **2 – Diseño de flujos conversacionales y arquitectura de integración**

Nombre del proyecto: **Chatbot Asistente en la Nube para Atención Automatizada (Lite)**

Equipo: **20**

Tutor: **CARLOS GUILLERMO RODRIGUEZ ALVAREZ**

Ciclo: **6**

1. Información general

Semana No. 5A (Se dividió para entregar Fase 3 según calendario)
Fecha (inicio–fin): 08 – 11 de febrero 2026
Integrantes presentes:
1. ADA LISSETTE GONZALEZ CASTRO
2. KAREN SORAYA MARTÍNEZ CORBERA
Roles activos en la semana: • **Líder de Proyecto Jr.** • **Ingeniera de Integración Cloud Jr.** • **Diseñadora de Conversaciones Jr.** • **Desarrolladora de Automatización Jr.** • **QA / Documentador Técnico Jr.**

2. Actividades realizadas

Actividad realizada	Descripción técnica	Responsable	Evidencia
Configuración del entorno de desarrollo	Creación de entorno virtual en Python, instalación de dependencias (Flask, gspread, google-auth) y verificación de ejecución del servidor local.	Karen Martínez	Evidencia 1
Implementación del webhook en Flask	Desarrollo del endpoint `/webhook` para recibir solicitudes desde la plataforma conversacional y enrutar mensajes a los manejadores de flujos.	Karen Martínez	Evidencia 2
Diseño de Flujos conversacionales	Creación de Diagramas para explicar los flujos lógicos deseados en las interacciones con el chatbot	Ada González	Evidencia
Desarrollo de flujos conversacionales	Implementación de flujos multi-turno para consulta de requisitos y registro de solicitudes con manejo de estados por usuario.	Karen Martínez	Evidencia 1 y Evidencia 2

Refactorización y modularización del código	Separación del proyecto en módulos (`app.py`, `flows.py`, `data.py`) para mejorar mantenibilidad y claridad.	Karen Martínez	Evidencia 3
Integración con Google Sheets	Configuración de cuenta de servicio en Google Cloud, habilitación de APIs e implementación de persistencia de datos en Sheets.	Karen Martínez	Evidencia 4 y Evidencia 5
Integración con Dialogflow (webhook)	Configuración de intents que delegan procesamiento al webhook y ajuste del evento WELCOME para la bienvenida.	Karen Martínez	Evidencia 6 y Evidencia 7
Exposición del webhook con Ngrok	Publicación temporal del endpoint local para permitir la comunicación con Dialogflow durante pruebas.	Karen Martínez	Evidencia 8
Pruebas funcionales end-to-end	Validación del flujo completo (inicio → requisitos/registro → cierre) y verificación de persistencia en Sheets.	Karen Martínez	Evidencia 5 y 9
Documentación de puntos de integración	Redacción y explicación de los puntos de integración entre la plataforma conversacional, el backend del chatbot y los servicios externos, detallando el rol de cada componente dentro de la arquitectura.	Ada González	Evidencia 10
Justificación de flujos y decisiones de diseño	Descripción del propósito de cada flujo conversacional y su relación con los objetivos del chatbot, proporcionando el contexto conceptual para la fase de implementación.	Ada González	Evidencia 10

3. Herramientas utilizadas

Herramienta / Servicio	Propósito	Observaciones
Python	Lenguaje principal para implementar el backend del chatbot y la lógica de los flujos conversacionales.	Permite un control total del flujo, manejo de estados y fácil integración con servicios externos.
Flask	Framework web para exponer el webhook que recibe las solicitudes desde la plataforma conversacional.	Ligero y adecuado para prototipos funcionales y servicios web sencillos.
Dialogflow	Plataforma conversacional para la detección inicial de intenciones y	Facilita la integración con canales de mensajería y

	gestión de mensajes de bienvenida, ayuda y consultas simples.	reduce la complejidad en la clasificación inicial de mensajes.
Postman	Herramienta de pruebas para simular solicitudes HTTP al webhook y validar respuestas del backend.	Permite pruebas controladas sin depender del canal final (Telegram).
Google Cloud Platform	Plataforma en la nube utilizada para la gestión de APIs y credenciales de acceso a Google Sheets.	Se utilizó una cuenta de servicio para autenticación segura.
Google Sheets	Almacenamiento ligero de las solicitudes registradas por el chatbot.	Solución accesible y adecuada para prototipos y fases iniciales del proyecto.
Ngrok	Exposición temporal del servidor local para permitir la comunicación con Dialogflow durante pruebas.	Útil en entornos de desarrollo; no recomendado para producción.
Visual Studio Code	Editor de código utilizado para el desarrollo del proyecto.	Facilita la organización del proyecto y la depuración del código.
GitHub	Control de versiones y repositorio del proyecto.	Permite el seguimiento de cambios y el trabajo colaborativo del equipo.
Google Docs / Word	Documentación del proyecto.	Útil para la elaboración de entregables académicos y registro del proceso.

4. Problemas detectados

Problema	Impacto	Causa probable	Evidencia
Conflictos en la activación de intents en la plataforma conversacional	Interrupción de flujos conversacionales y respuestas incorrectas del chatbot durante interacciones multi-turno.	Configuración inicial de intents con training phrases muy genéricas que interferían con flujos activos.	Evidencia 11
Errores en la interpretación de respuestas del usuario	Fallos en el reconocimiento de confirmaciones (“sí” / “no”) o selección de opciones del menú.	Variaciones en la forma de escribir del usuario (uso de tildes, mayúsculas o texto libre).	Evidencia 12
Pérdida de estado en pruebas aisladas	Reinicio inesperado de flujos conversacionales durante pruebas manuales.	Uso de sesiones distintas en las pruebas con Postman, lo que provocaba la pérdida del contexto de conversación.	Evidencia 12

5. Soluciones aplicadas

Solución aplicada	Problema asociado	Responsable	Evidencia
Ajuste de configuración de intents y uso del evento WELCOME para la bienvenida	Conflictos en la activación de intents en la plataforma conversacional	Karen Martínez	Evidencia 13
Normalización de texto y validación de entradas del usuario	Errores en la interpretación de respuestas del usuario	Karen Martínez	Evidencia 14
Unificación del identificador de sesión en pruebas	Pérdida de estado en pruebas aisladas	Karen Martínez	Evidencia 15

6. Estado del proyecto esta semana

ÁREA	ESTADO	COMENTARIOS
FASE ACTUAL	Completada	La fase se finalizó oficialmente el 11 de agosto.
CUMPLIMIENTO/ RIESGOS	Pendiente a Documentar	Recopilaremos toda la evidencia de la fase 3 donde haremos una versión más estable del bot.

7. Plan de trabajo para semana 5B (12 – 15 de febrero)

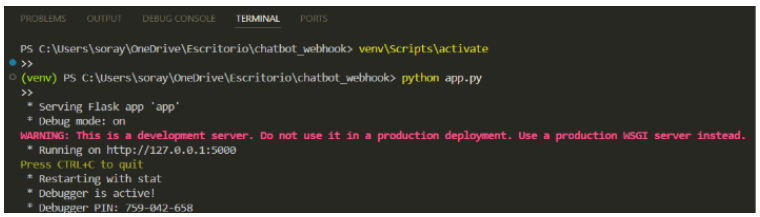
TAREA PRÓXIMA SEMANA	RESPONSABLE	FASE	ENTREGABLE ESPERADO
MEJORAR CONVERSACIÓN	Karen Martínez	3	Flujos estables en Telegram
DOCUMENTAR ACTIVIDADES	Ada González	3	Bitácora detallada
AJUSTAR ESTILO DE MENSAJES, TIEMPOS, VALIDACIONES	Karen Martínez	3	Capturas de escenarios conversacionales
REGISTRAR PROCEDIMIENTO SEGUIDO Y HACER PRUEBAS	Ada González	3	Evidencias de funcionamiento mejorado

8. Reflexión individual por rol

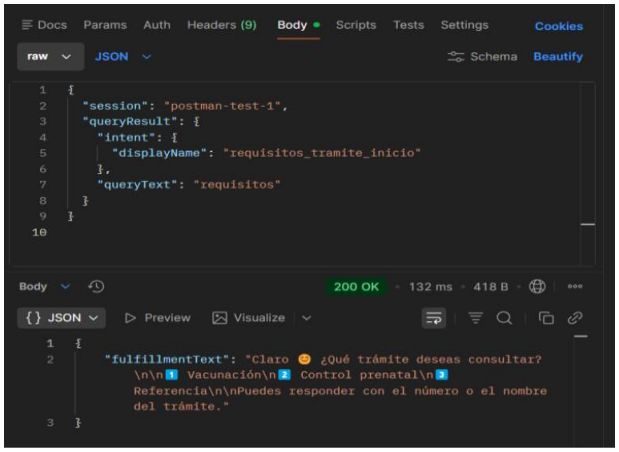
INTEGRANTE	ROL	REFLEXIÓN DE LA SEMANA
ADA GONZÁLEZ	**Diseñadora de Conversaciones Jr.**	En esta fase se aprendió a entender mejor el funcionamiento del sistema y sus procesos. El diseño de los diagramas y flujos ayudó a visualizar cómo operará el sistema y a tener mayor claridad antes de continuar con las siguientes etapas.
KAREN MARTÍNEZ	**Líder de proyecto Jr.**	Coordinar la Fase 2 implicó equilibrar el diseño conceptual del chatbot con los retos técnicos de su implementación, así como organizar el trabajo del equipo para que los aportes individuales se integraran en una solución coherente.
	Ingeniera de Integración Cloud Jr.	Este proceso permitió reconocer la importancia de una comunicación clara, la definición de responsabilidades y la toma de decisiones oportunas para avanzar pese a las dificultades técnicas que surgieron durante la implementación.

9. Registro de evidencias

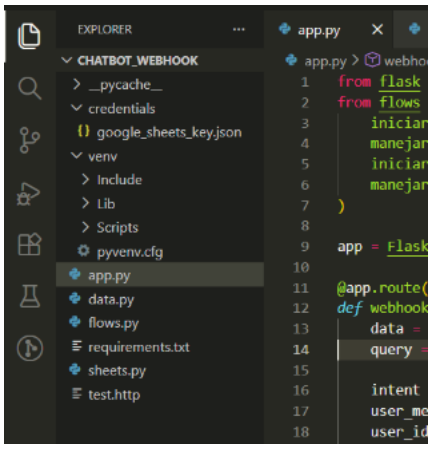
Evidencia 1: Terminal con entorno activado



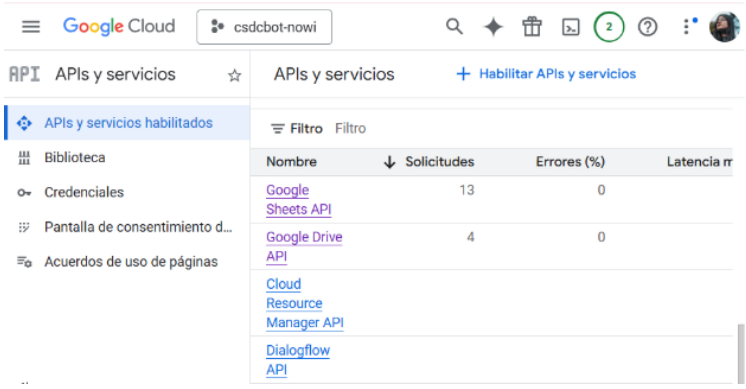
Evidencia 2: Postman enviando POST al endpoint y Respuesta JSON



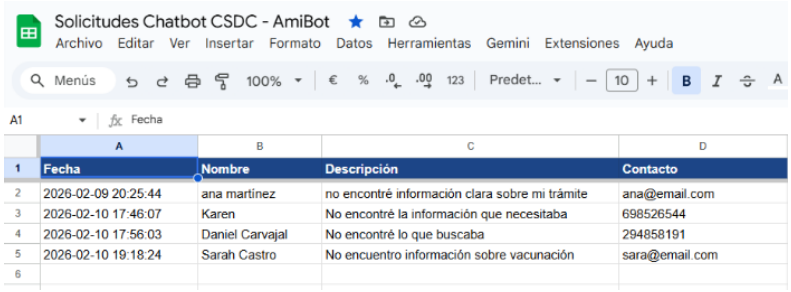
Evidencia 3: Estructura del proyecto



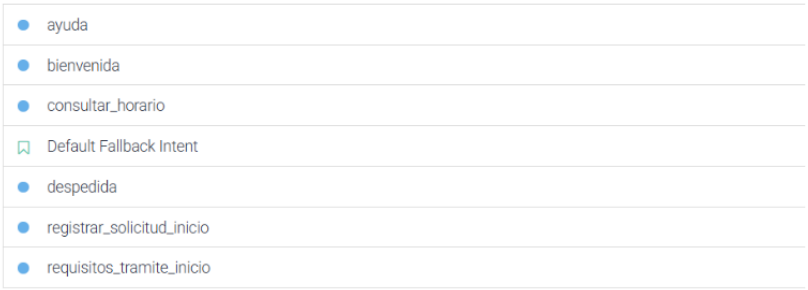
Evidencia 4: Google Cloud con APIs habilitadas



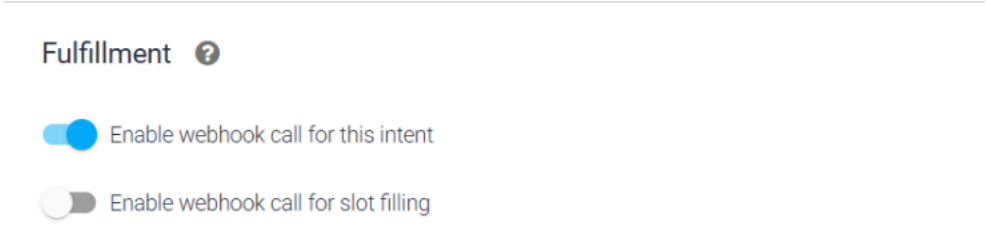
Evidencia 5: Hoja de cálculo con registros creados por el bot.



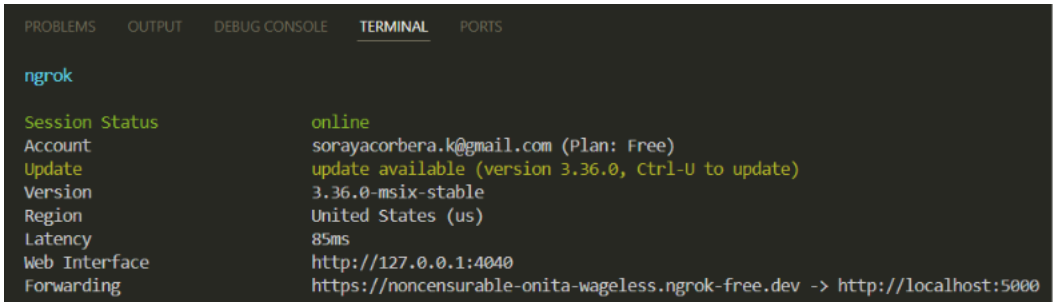
Evidencia 6: Configuración de intents en Dialogflow



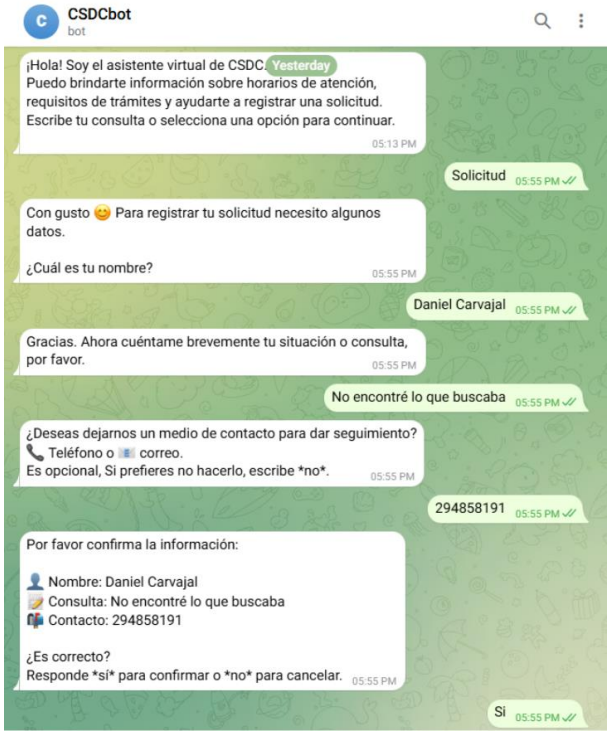
Evidencia 7: Webhook habilitado



Evidencia 8: Ngrok mostrando la URL pública activa.



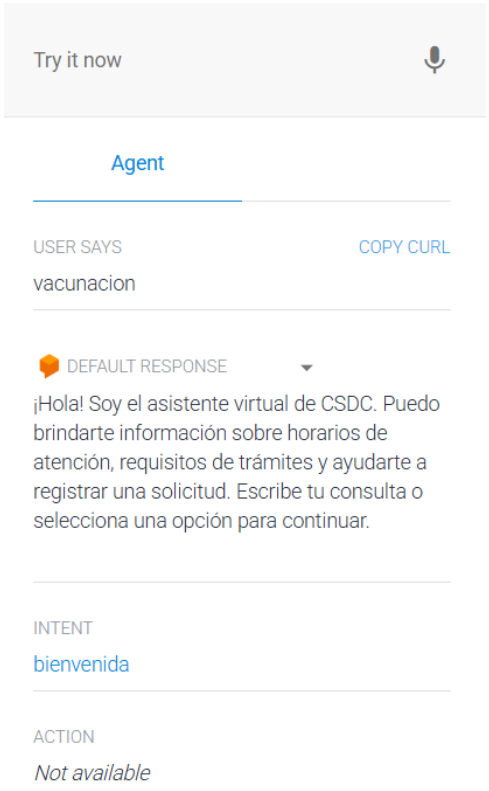
Evidencia 9: Prueba en Telegram



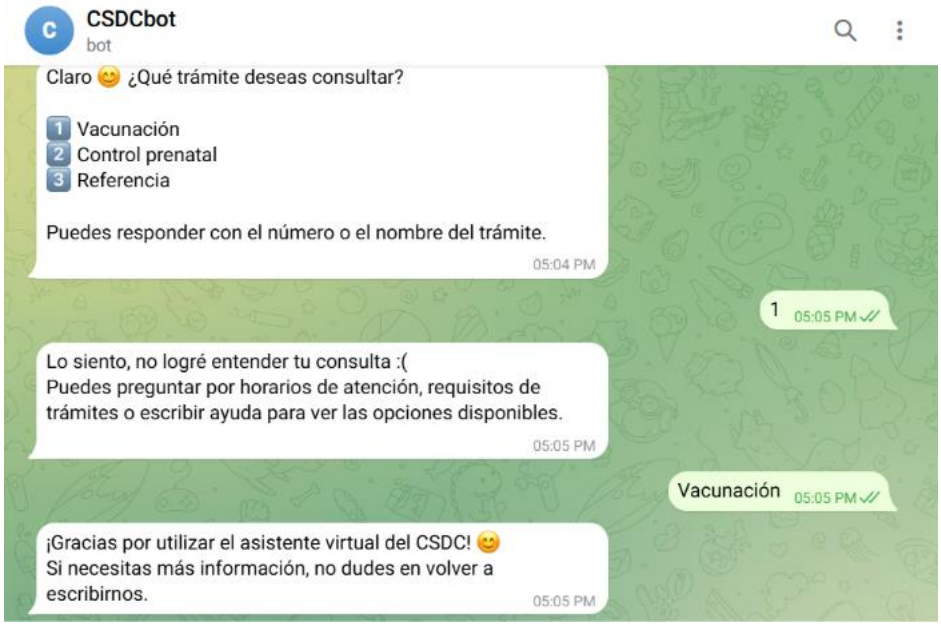
Evidencia 10: Documento en repositorio

https://github.com/corberaks/grupo20_Chatbot/blob/872c71c6f6e4b086ea949bb5ce34e485725c6bf3/docs/Fase%202%20-%20Flujos%20y%20Arquitectura.pdf

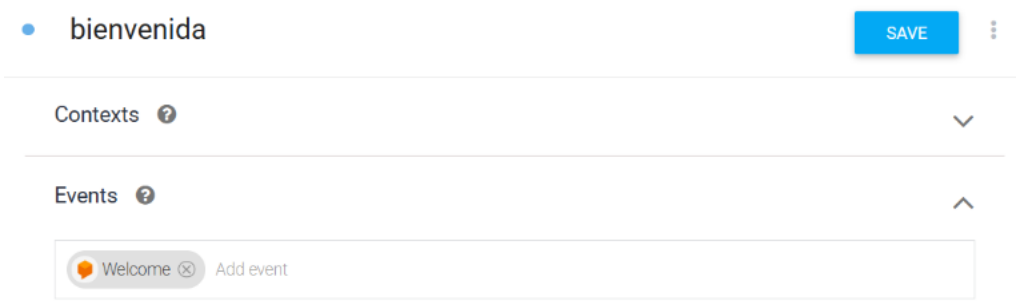
Evidencia 11: Intent activado por error



Evidencia 12: Falta de normalización y la conversación se reinicia al cambiar de estado de intent



Evidencia 13: Modificación de Intent Bienvenida, ahora solo relacionado con el evento "WELCOME"



Evidencia 14: Normalización de texto

```python
#todo el texto en minúsculas, sin tildes ni símbolos
def normalizar_texto(texto):
    texto = texto.lower()
    texto = unicodedata.normalize("NFD", texto)
    texto = "".join(c for c in texto if unicodedata.category(c) != "Mn")
    texto = re.sub(r"[^a-z0-9\s]", "", texto)
    return texto.strip()
```

Evidencia 15: Manejo de estados con Python

```python
user_states = {}

##INTENCION 3: requisitos_tramite##
#Menu de tramites disponibles
def iniciar_requisitos(user_id):
    user_states[user_id] = {
        "flow": "requisitos",
        "step": "esperando_tramite"
    }
```