



SECRETARÍA
DE INNOVACIÓN

Escuela Superior de Innovación y Tecnología



Técnico Superior Universitario Servicios en la Nube

Estancia Profesional

Entregable Fase 3

Matriz y Documentación de Pruebas

Chatbot Asistente en la Nube para Atención Automatizada (Lite)

Docente: **Carlos Guillermo Rodríguez**

Integrantes Grupo SN20

Ada Lissette Gonzalez Castro

Karen Soraya Martínez Corbera

Fecha: **15 de febrero de 2026**



SECRETARÍA
DE INNOVACIÓN

ESIT - TSU EN SERVICIOS EN LA NUBE



Tabla de contenido

Resumen ejecutivo.....	3
1. Matriz de Pruebas	3
1.1 Flujo de Intención: requisitos_tramite	3
1.2 Flujo de Intención: registrar_solicitud	5
1.3 Intenciones en Dialogflow.....	6
2. Problemas y Correcciones.....	7
2.1 Continuidad de flujos (Simulando Dialogflow)	7
2.2 Credenciales como variables de entorno	8
3. Ajustes en el contenido conversacional.....	12
Evidencias.....	13
Pruebas en REST (Entorno Local).....	13
Pruebas en REST (Entorno Cloud).....	17
Pruebas en Telegram	21
Flujos completos en Telegram	26
Conclusiones	30



Resumen ejecutivo

El presente documento corresponde a la Fase 3 del proyecto del chatbot CSDC, enfocada en la ejecución de pruebas funcionales, validación de los flujos conversacionales y aplicación de correcciones técnicas. En este informe se documenta la matriz de pruebas utilizada para evaluar los flujos de intención de consulta de requisitos de trámites y registro de solicitudes, así como los resultados obtenidos durante las pruebas en entorno local (REST Client), entorno cloud y en el canal de Telegram.

El documento incluye el análisis de escenarios de funcionamiento normal y escenarios de fallo, a partir de los cuales se identificaron problemas en la continuidad de los flujos, interpretación de intenciones y configuración del entorno. Con base en estos hallazgos, se describen las correcciones aplicadas y los ajustes en el contenido conversacional del chatbot.

Finalmente, se presentan evidencias visuales con análisis y una comparación del comportamiento antes y después de las correcciones, con el objetivo de respaldar el proceso de validación y mejora continua del asistente virtual.

1. Matriz de Pruebas

Se diseñó y ejecutó una matriz de pruebas para validar los flujos principales del chatbot, contemplando escenarios normales y de fallo. Las pruebas incluyen la consulta de requisitos, el registro de solicitudes en diferentes entornos: local con pruebas en REST y la integración de extremo a extremo entre Telegram, Dialogflow, el backend en la nube (Render) y el repositorio de datos (Google Sheets). Para cada caso se documentó el resultado, las correcciones aplicadas y la evidencia correspondiente.

1.1 Flujo de Intención: requisitos_tramite

Descripción de Intención: Brinda información sobre los requisitos necesarios para realizar trámites específicos. (vacunación, control prenatal, referencia médica).



Intent	Entrada	Salida esperada	Resultado	Fix aplicado	Evidencia
requisitos_tramite_inicio	"quiero consultar un trámite"	Muestra menú de trámites (Vacunación, Control prenatal, Referencia)	OK	N/A	Evidencia 1
requisitos_tramite_detalle	"1"	Muestra requisitos de vacunación	FAIL Volvió a mostrar el menú de trámites	Ajuste en los escenarios de prueba para simular el comportamiento de Dialogflow y que no se reinicie el intent.	Evidencia 2
requisitos_tramite_detalle	"1"	Muestra requisitos de Vacunación	OK	En pruebas con REST se utilizan intents genéricos en los turnos de continuación.	Evidencia 3
requisitos_tramite_inicio	"requisitos"	Muestra menú de trámites (Vacunación, Control prenatal, Referencia)	OK	Integración en entorno Cloud.	Evidencia 12
requisitos_tramite_detalle	"control"	Muestra requisitos de Control prenatal	OK	Integración en entorno Cloud.	Evidencia 13
requisitos_tramite_inicio	"Quiero saber los requisitos de un trámite"	Muestra menú de trámites (Vacunación, Control prenatal, Referencia)	OK	Integración a Telegram.	Evidencia 20
requisitos_tramite_detalle	"Referencia por fi"	Muestra requisitos para Referencia médica	OK	Integración a Telegram.	Evidencia 20
requisitos_tramite_inicio	"saber los requisitos de un trámite"	Muestra menú de trámites (Vacunación, Control prenatal, Referencia)	OK	Mensaje de respuesta mejorado.	Evidencia 31
requisitos_tramite_detalle	"4"	Solicita nuevamente elegir entre las opciones válidas	OK	Mensaje de respuesta mejorado.	Evidencia 31



1.2 Flujo de Intención: registrar_solicitud

Descripción de Intención: Permite registrar una solicitud ciudadana básica cuando la consulta no puede resolverse automáticamente.

Intent	Entrada	Salida esperada	Resultado	Fix aplicado	Evidencia
registrar_solicitud_inicio	"quiero registrar una solicitud"	Bot solicita nombre del usuario	OK	N/A	Evidencia 4
registrar_solicitud_descripcion	"María Pérez"	Bot solicita descripción del caso	OK	N/A	Evidencia 5
registrar_solicitud_descripcion	"No pude agendar una cita"	Bot solicita contacto opcional	OK	N/A	Evidencia 6
registrar_solicitud_contacto	"5555-1234"	Bot muestra mensaje de confirmación	OK	N/A	Evidencia 7
registrar_solicitud_confirmacion	"sí"	Registro exitoso y guardado en Google Sheets	FAIL Fallback en pantalla, sin acceso a Google Sheets	Obtener token JSON de la carpeta oculta con credenciales (Solo para prueba REST en entorno local)	Evidencia 8 y 9
registrar_solicitud_confirmacion	"sí"	Registro exitoso y guardado en Google Sheets	OK	Para pruebas locales se utilizaron variables de entorno de sesión en PowerShell.	Evidencia 10 y 11
registrar_solicitud_inicio	"quiero registrar una solicitud"	Bot solicita nombre del usuario	OK	Integración en entorno Cloud.	Evidencia 14
registrar_solicitud_descripcion	"Soraya Corbera"	Bot solicita descripción del caso	OK	Integración en entorno Cloud.	Evidencia 15
registrar_solicitud_descripcion	"No me han atendido en el centro de salud desde ayer"	Bot solicita contacto opcional	OK	Integración en entorno Cloud.	Evidencia 16
registrar_solicitud_contacto	"no"	Bot muestra mensaje de confirmación	OK	Integración en entorno Cloud.	Evidencia 17
registrar_solicitud_confirmacion	"sí"	Registro exitoso y guardado en Google Sheets	OK	Integración en entorno Cloud.	Evidencia 18 y 19
registrar_solicitud_inicio	"Debo registrar una solicitud"	Bot solicita nombre del usuario	OK	Integración a Telegram.	Evidencia 21



registrar_solicitud_descripcion	“Sofia Gonzales”	Bot solicita descripción del caso	FAIL Mensaje de despedida activado	Editar ajustes de ML en Dialogflow	Evidencia 21 y 22
registrar_solicitud_inicio	“Debo registrar una solicitud”	Bot solicita nombre del usuario	OK	Entrenar agente directamente en Dialogflow	Evidencia 23
registrar_solicitud_descripcion	“Sofia Gonzales”	Bot solicita descripción del caso	FAIL Mensaje de despedida activado	Eliminar todos los parámetros del intent “despedida”	Evidencia 24 y 25
registrar_solicitud_descripcion	“Sofia Gonzales”	Bot solicita descripción del caso	OK	N/A	Evidencia 26
registrar_solicitud_descripcion	“No me dieron mis medicinas”	Bot solicita contacto opcional	OK	N/A	Evidencia 27
registrar_solicitud_contacto	“sofi@mail.org”	Bot muestra mensaje de confirmación	OK	N/A	Evidencia 28
registrar_solicitud_confirmacion	“si”	Registro exitoso y guardado en Google Sheets	OK	N/A	Evidencia 29 y 30
registrar_solicitud_confirmacion	“23”	Bot solicita que responda “sí o no” nuevamente	OK	Mensaje de respuesta mejorado.	Evidencia 32
registrar_solicitud_confirmacion	“no”	Cancelar registro y ofrecer ayuda adicional	OK	Mensaje de respuesta mejorado.	Evidencia 32

1.3 Intenciones en Dialogflow

Los siguientes datos corresponden a las pruebas realizadas a las intenciones que son procesadas por Dialogflow directamente: bienvenida, consultar_horarios, fallback, ayuda y despedida.

Intent	Entrada	Salida esperada	Resultado	Fix aplicado	Evidencia
bienvenida	“Hola!”	Bot saluda y muestra menú principal	OK	Mensaje más claro y amigable	Evidencia 33
consultar_horario	“Quiero saber los horarios de la clínica”	Mostrar horarios	OK	Mensaje más claro y amigable	Evidencia 33



consultar_horario	"A qué hora abren?"	Mostrar horarios	OK	Mensaje más claro y amigable	Evidencia 33
consultar_horario	"Los horarios de atención"	Mostrar horarios	OK	Mensaje más claro y amigable	Evidencia 34
fallback	Mensaje fuera de contexto	Informar error y sugerir alternativas	OK	N/A	Evidencia 34
fallback	"asdfghj"	Informar error y sugerir alternativas	OK	N/A	Evidencia 35
ayuda	"Qué puedo hacer aquí?"	Mostrar menú de consultas soportadas	OK	Mensaje más claro y empático	Evidencia 36
ayuda	"ayuda"	Mostrar menú de consultas soportadas	OK	N/A	Evidencia 36
despedida	"Gracias, eso es todo"	Mensaje de cierre	OK	Mensaje de cierre más amable	Evidencia 37

2. Problemas y Correcciones

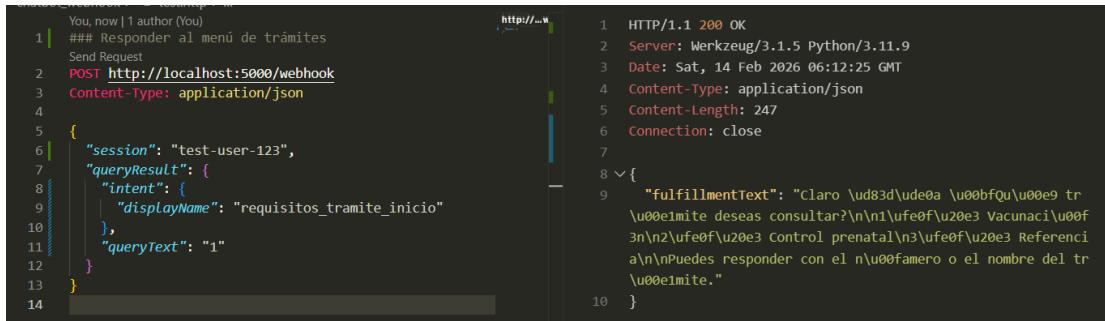
2.1 Continuidad de flujos (Simulando Dialogflow)

Problema detectado:

Durante las pruebas manuales con REST Client para validar el flujo de consulta de requisitos de trámite, se observó que al reenviar el intent de inicio *requisitos_tramite_inicio* en turnos posteriores, el flujo conversacional se reiniciaba y el chatbot volvía al mensaje inicial en lugar de continuar el diálogo.

Este comportamiento no se presentaba en Telegram ni en el uso real del bot, ya que Dialogflow gestiona correctamente los contextos entre turnos. La situación se originó por la forma en que se construían los requests en REST Client, donde se enviaba repetidamente el intent de inicio, provocando un reinicio artificial del flujo durante las pruebas.





```

1 You, now | 1 author (You)
2     ### Responder al menú de trámites
3 Send Request
4 POST http://localhost:5000/webhook
5 Content-Type: application/json
6
7 {
8     "session": "test-user-123",
9     "queryResult": {
10         "intent": {
11             "displayName": "requisitos_tramite_inicio"
12         },
13         "queryText": "1"
14     }
15 }
```

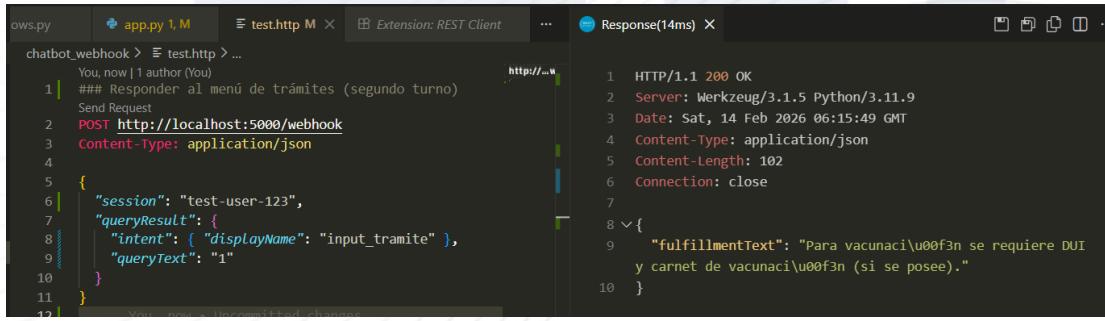
```

1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.5 Python/3.11.9
3 Date: Sat, 14 Feb 2026 06:12:25 GMT
4 Content-Type: application/json
5 Content-Length: 247
6 Connection: close
7
8 {
9     "fulfillmentText": "Claro \ud83d\ude0a \u00bfQu\u00e9 tr\u00e1mite deseas consultar?\n\n1\uufe0f\u20e3 Vacunaci\u00f3n\n2\uufe0f\u20e3 Control prenatal\n3\uufe0f\u20e3 Referencia\n\nPuedes responder con el n\u00famero o el nombre del tr\u00e1mite."
10 }
```

Corrección aplicada:

Se ajustaron los escenarios de prueba en REST Client para enviar el intent de inicio únicamente en el primer turno de la conversación. En los turnos siguientes, se utilizaron intents genéricos o mensajes de continuación, permitiendo que el flujo avanzara según la lógica definida en el backend y los contextos activos.

Con este ajuste, las pruebas reflejaron de forma más fiel el comportamiento real del chatbot en producción y se validó correctamente la continuidad de los flujos conversacionales, confirmando que el problema no estaba en el código sino en la configuración de las pruebas.



```

1 You, now | 1 author (You)
2     ### Responder al menú de trámites (segundo turno)
3 Send Request
4 POST http://localhost:5000/webhook
5 Content-Type: application/json
6
7 {
8     "session": "test-user-123",
9     "queryResult": {
10         "intent": {
11             "displayName": "input_tramite"
12         },
13         "queryText": "1"
14     }
15 }
```

```

1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.5 Python/3.11.9
3 Date: Sat, 14 Feb 2026 06:15:49 GMT
4 Content-Type: application/json
5 Content-Length: 102
6 Connection: close
7
8 {
9     "fulfillmentText": "Para vacunaci\u00f3n se requiere DUI y carnet de vacunaci\u00f3n (si se posee)."
10 }
```

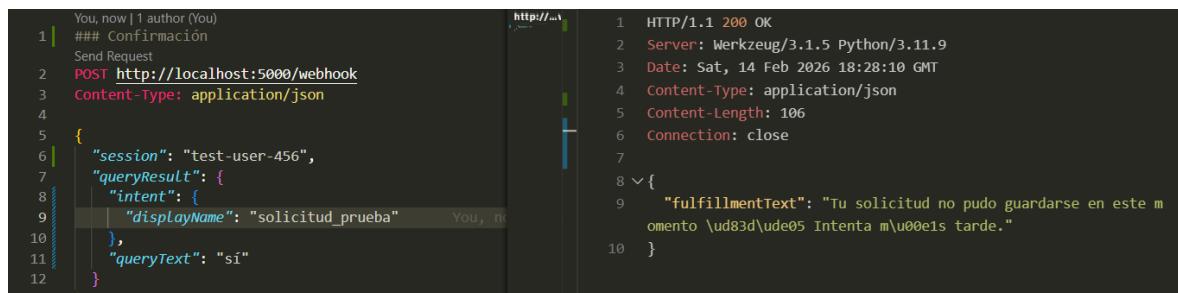
2.2 Credenciales como variables de entorno

Problema detectado:

Durante las pruebas del flujo *registrar_solicitud* en el entorno local, se presentó un fallo en la integración con Google Sheets / Google Drive, ya que el sistema no lograba encontrar el archivo de credenciales JSON necesario para autenticarse con las APIs de Google.



Esto ocurrió porque, por buenas prácticas de seguridad, el archivo de credenciales no se incluye ni se sincroniza en el repositorio de GitHub. En consecuencia, al preparar el proyecto para su posterior migración a la nube, la carpeta local de credenciales no estaba disponible en el entorno de ejecución. Además, se identificó que la herramienta **setx** en Windows presenta limitaciones al definir variables de entorno con valores extensos, como archivos JSON de credenciales, lo que dificultó su uso como solución directa en el entorno local.

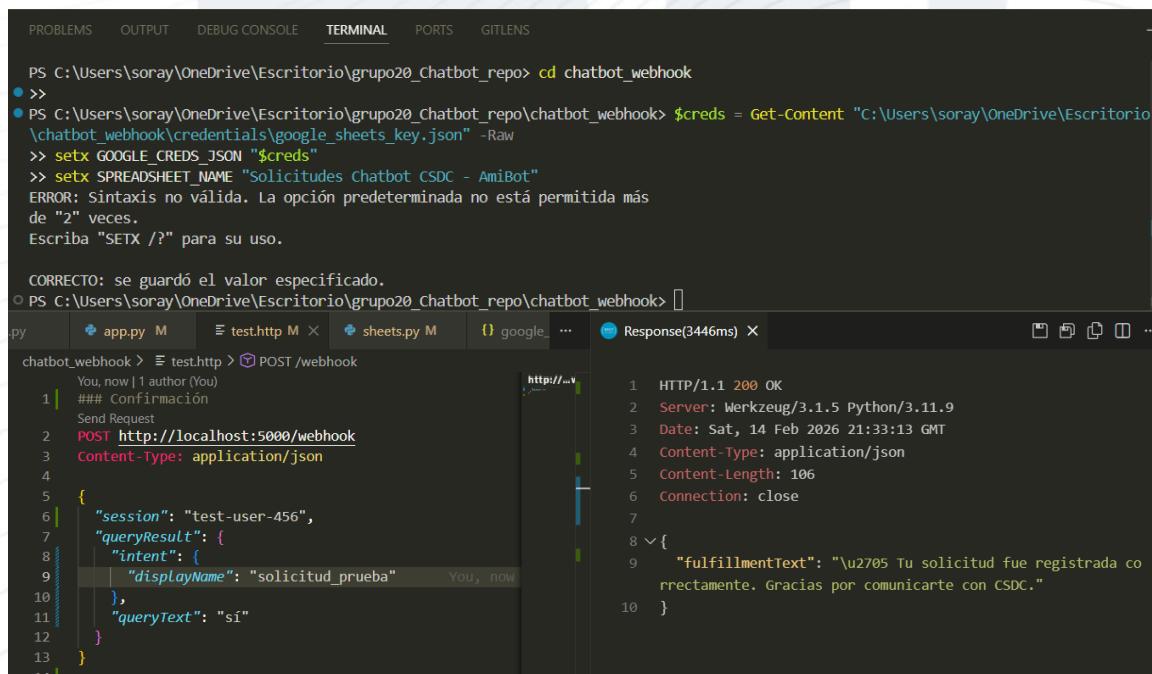


```
You, now | 1 author (You)
1  ### Confirmación
2  Send Request
3  POST http://localhost:5000/webhook
4  Content-Type: application/json
5
6  {
7    "session": "test-user-456",
8    "queryResult": {
9      "intent": {
10        "displayName": "solicitud_prueba"
11      },
12      "queryText": "sí"
13    }
14
```

```
HTTP/1.1 200 OK
1  Server: Werkzeug/3.1.5 Python/3.11.9
2  Date: Sat, 14 Feb 2026 18:28:10 GMT
3  Content-Type: application/json
4  Content-Length: 106
5  Connection: close
6
7
8  {
9    "fulfillmentText": "Tu solicitud no pudo guardarse en este momento \ud83d\ude05 Intenta más tarde."
10 }
```

Corrección aplicada:

Como solución **temporal** para validar el funcionamiento del flujo, se configuraron las credenciales mediante variables de entorno de sesión en PowerShell, apuntando al archivo JSON ubicado localmente (fuera del repositorio), lo que permitió confirmar que la integración con Google Sheets funcionaba correctamente.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS C:\Users\soray\OneDrive\Escritorio\grupo20_Chatbot_repo> cd chatbot_webhook
>>
● PS C:\Users\soray\OneDrive\Escritorio\grupo20_Chatbot_repo\chatbot_webhook> $creds = Get-Content "C:\Users\soray\OneDrive\Escritorio\chatbot_webhook\credentials\google_sheets_key.json" -Raw
>> setx GOOGLE_CREDS_JSON "$creds"
>> setx SPREADSHEET_NAME "Solicitudes Chatbot CSDC - AmiBot"
ERROR: Sintaxis no válida. La opción predeterminada no está permitida más de "2" veces.
Escriba "SETX /?" para su uso.

CORRECTO: se guardó el valor especificado.
○ PS C:\Users\soray\OneDrive\Escritorio\grupo20_Chatbot_repo\chatbot_webhook>
```

```
app.py test.http M test.http M sheets.py google ... Response(3446ms) X
chatbot_webhook > test.http > POST /webhook
You, now | 1 author (You)
1  ### Confirmación
2  Send Request
3  POST http://localhost:5000/webhook
4  Content-Type: application/json
5
6  {
7    "session": "test-user-456",
8    "queryResult": {
9      "intent": {
10        "displayName": "solicitud_prueba"
11      },
12      "queryText": "sí"
13    }
14
```

```
HTTP/1.1 200 OK
1  Server: Werkzeug/3.1.5 Python/3.11.9
2  Date: Sat, 14 Feb 2026 21:33:13 GMT
3  Content-Type: application/json
4  Content-Length: 106
5  Connection: close
6
7
8  {
9    "fulfillmentText": "\u2708 Tu solicitud fue registrada correctamente. Gracias por comunicarte con CSDC."
10 }
```



Posteriormente, en el entorno de despliegue en la nube (Render), las credenciales se definieron como **variables de entorno persistentes** proporcionadas por el proveedor, aprovechando sus mecanismos de seguridad para el manejo de secretos. Esta configuración permitió restablecer el funcionamiento del flujo *registrar_solicitud* sin exponer credenciales en el repositorio y alineando la solución con buenas prácticas de seguridad para entornos productivos.



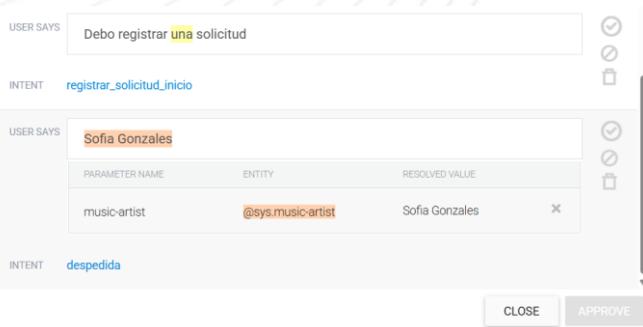
The screenshot shows a service configuration page for a Python 3 application named "grupo20_Chatbot". The service is currently running on a free tier. The URL listed is <https://grupo20-chatbot.onrender.com>. There are "Connect" and "Manual Deploy" buttons at the top right.

2.3 Parámetros e interpretaciones de Dialogflow

Problema detectado:

Durante las pruebas del flujo *registrar_solicitud*, se identificó un comportamiento inesperado del motor de NLU de Dialogflow al procesar ciertos nombres propios. En particular, al ingresar el nombre “Sofía Gonzales”, el chatbot interrumpía el flujo y activaba respuestas de despedida, en lugar de continuar con el registro de la solicitud.

Aunque el flujo funcionaba correctamente con la mayoría de nombres, este caso evidenció que Dialogflow estaba asociando erróneamente el término “Sofía” con una entidad automática del sistema (parámetro *music_artist*), generada por defecto por la plataforma. Esta clasificación incorrecta provocaba que se dispararan intenciones no relacionadas con el dominio del chatbot, afectando la continuidad del flujo conversacional.



The screenshot shows the Dialogflow interface. At the top, a user says "Debo registrar una solicitud" and the intent is "registrar_solicitud_inicio". Below, a user says "Sofia Gonzales" and the system identifies it as a music artist with the resolved value "Sofia Gonzales". The intent for this response is "despedida". At the bottom, there are "CLOSE" and "APPROVE" buttons.



Corrección aplicada:

Se revisó y depuró la configuración de parámetros y entidades automáticas creadas por defecto en Dialogflow, eliminando aquellas que no eran relevantes para el dominio del chatbot. Adicionalmente, se reforzaron los ejemplos de entrenamiento de las intenciones del flujo registrar_solicitud, asegurando que el modelo priorizara el contexto del registro de datos del usuario por encima de interpretaciones genéricas.

General Languages **ML Settings** Export and Import Environments Speech Share Advanced

ML CLASSIFICATION THRESHOLD
Define the threshold value for the confidence score. If the returned value is less than the threshold value, then a fallback intent will be triggered, or if there is no fallback intents defined, no intent will be triggered.

0.5

AUTOMATIC SPELL CORRECTION

Allow ML to correct spelling of query during request processing.

AUTOMATIC TRAINING

Disable automatic re-training the agent after every agent modification.

Automatic training may slow UI responsiveness and is not recommended for large agents. You can kick off agent training manually through [API](#) or by clicking the 'TRAIN' button below.

AGENT VALIDATION

Automatically validate the agent when agent training is performed.

TRAIN

Con estos ajustes, el chatbot dejó de interrumpir creativamente el flujo al procesar nombres propios como “Sofía Gonzales” y se logró una mayor estabilidad en el reconocimiento de intención durante el proceso de registro de solicitudes.

USER SAYS

Sofia Gonzales

COPY CURL

DEFAULT RESPONSE

Gracias. Ahora cuéntame brevemente tu situación o consulta, por favor.

CONTEXTS

RESET CONTEXTS

_system_counters_



3. Ajustes en el contenido conversacional

Durante la Fase 3 se realizó un refinamiento del diseño conversacional (UX writing), mejorando la claridad, tono empático y estructura de los mensajes presentados al usuario. Se incorporaron llamadas a la acción claras, formato en listas y mensajes de error más orientados al usuario, logrando una experiencia más fluida y comprensible en los flujos de consulta y registro de solicitudes.

Flujo / Mensaje	Fase 2 (Antes)	Fase 3 (Después)	Mejora aplicada
Bienvenida	“¡Hola! Soy el asistente virtual del CSDC... ¿En qué puedo ayudarte hoy?”	“¡Hola! 😊 Soy el asistente virtual del CSDC. Puedo ayudarte con horarios, requisitos de trámites o registrar una solicitud. ¿En qué puedo ayudarte hoy?”	Tono más cercano, estructura por opciones, CTA claro
Ayuda	“Con gusto puedo ayudarte a: Consultar horarios... Solo escribe tu consulta...”	“¡Con gusto te ayudo! 😊 Puedo apoyarte con: horarios, requisitos, registrar una solicitud. Solo dime qué necesitas.”	Lenguaje natural, formato en lista, CTA directo
Menú de trámites	“Claro 😊 ¿Qué trámite deseas consultar? 1 Vacunación...”	“¡Con gusto te ayudo! 😊 ¿Qué trámite deseas consultar? 1 Vacunación, 2 Control prenatal, 3 Referencia médica...”	Claridad, emojis moderados, mejor guía de acción
Error al identificar el trámite	“No logré identificar el trámite :(Responde con el número...”	“Lo siento, no pude identificar el trámite. Por favor indícame cuál deseas consultar...”	Mensaje empático, tono no punitivo, instrucción clara
Solicitud - pedir nombre	“¿Cuál es tu nombre?”	“Para comenzar, ¿me indicas tu nombre completo?” (o alternativa elegida)	Mayor claridad, tono respetuoso
Solicitud - descripción	“Ahora cuéntame brevemente tu situación...”	“Gracias. Ahora cuéntame brevemente tu situación o consulta para poder orientarte mejor.”	Contexto del porqué se pide el dato
Solicitud contacto opcional	“¿Deseas dejarnos un medio de contacto...? Es opcional...”	“¿Deseas dejarnos un medio de contacto para dar seguimiento? Puedes escribir un teléfono o correo. Es opcional...”	Claridad de propósito, ejemplos explícitos
Confirmación de datos	“Por favor confirma la información...”	“Por favor verifica que la información sea correcta: Nombre, Consulta, Contacto. ¿Todo está bien?”	Lenguaje más humano, formato más legible
Éxito al guardar	“Tu solicitud fue registrada correctamente.”	“✅ ¡Listo! Tu solicitud fue registrada correctamente. Gracias por comunicarte con CSDC.”	Refuerzo positivo, cierre empático
Error guardar (Sheets)	“Tu solicitud no pudo guardarse en	“Tuvimos un problema al registrar tu solicitud 😬 Por favor, intenta nuevamente en unos minutos...”	Mensaje más empático y orientado a acción



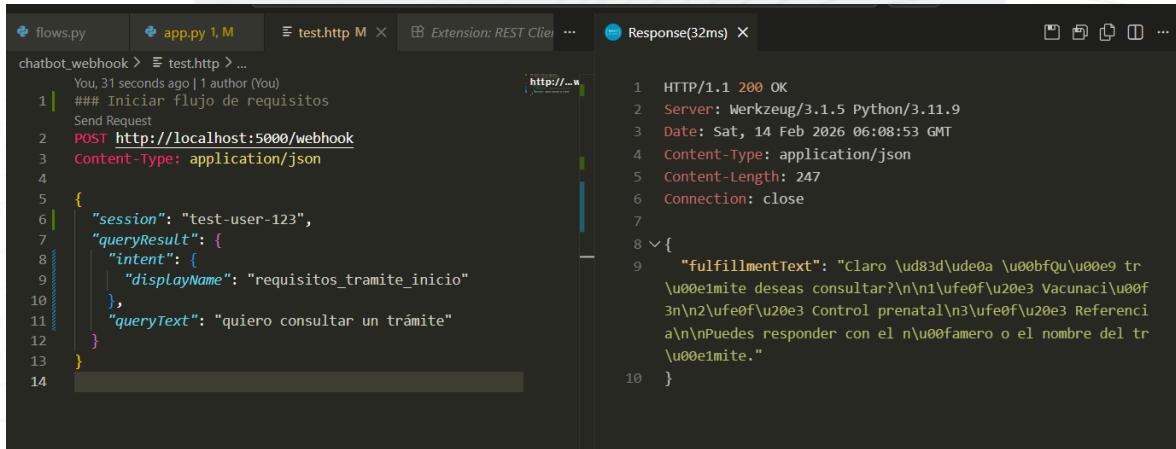
	este momento 😅 Intenta más tarde."		
Fallback / No entendido	(Deshabilitado o genérico)	"Lo siento, no logré entender tu mensaje. Puedes escribir <i>ayuda</i> para ver opciones."	UX responsable ante entradas no comprendidas
Despedida	"¡Gracias por utilizar el asistente virtual del CSDC!"	"¡Gracias por comunicarte con el asistente virtual del CSDC! 😊 Si necesitas algo más, aquí estaré."	Cierre más humano, puerta abierta

Evidencias

Pruebas en REST (Entorno Local)

Las siguientes evidencias corresponden a las pruebas realizadas con REST antes de migrar al entorno Cloud, a modo de documentar el funcionamiento de los flujos que se gestionan con Python.

Evidencia 1: Iniciar flujo de requisitos_trámite



```

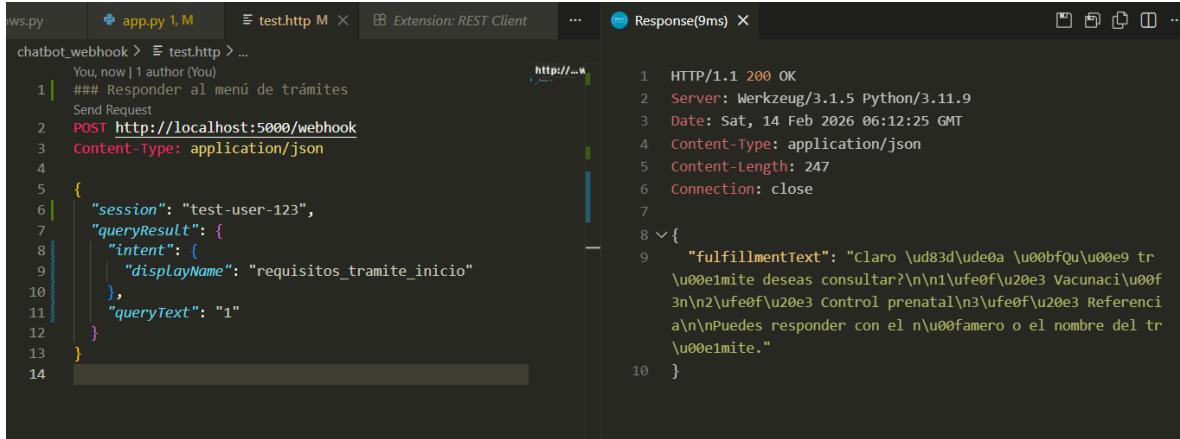
flows.py app.py 1.M test.http Extension: REST Client ...
chatbot_webhook > test.http > ...
You, 31 seconds ago | 1 author (You)
    You, 31 seconds ago | 1 author (You)
    1  ### Iniciar flujo de requisitos
    2  Send Request
    3  POST http://localhost:5000/webhook
    4  Content-Type: application/json
    5  {
    6      "session": "test-user-123",
    7      "queryResult": {
    8          "intent": {
    9              "displayName": "requisitos_trámite_inicio"
   10         },
   11         "queryText": "quiero consultar un trámite"
   12     }
   13 }
   14

http://...w
Response(32ms) X
1  HTTP/1.1 200 OK
2  Server: Werkzeug/3.1.5 Python/3.11.9
3  Date: Sat, 14 Feb 2026 06:08:53 GMT
4  Content-Type: application/json
5  Content-Length: 247
6  Connection: close
7
8  {
9      "fulfillmentText": "Claro \ud83d\ude0a \u200bfQu\u200e9 tr\u00fame deseas consultar?\n\n1\uufe0f\u20e3 Vacunaci\u00f3n\n2\uufe0f\u20e3 Control prenatal\n3\uufe0f\u20e3 Referenci\u00e1l\n\nPuedes responder con el n\u00famero o el nombre del tr\u00f3mite."
10 }

```



Evidencia 2: Error en mostrar información específica del trámite indicado



```

ws.py app.py 1, M test.http M Extension: REST Client ...
chatbot_webhook > test.http > ...
You, now | 1 author (You)
    ### Responder al menú de trámites
Send Request
2 POST http://localhost:5000/webhook
Content-Type: application/json
4
{
    "session": "test-user-123",
    "queryResult": {
        "intent": {
            "displayName": "requisitos_tramite_inicio"
        },
        "queryText": "1"
    }
}

```

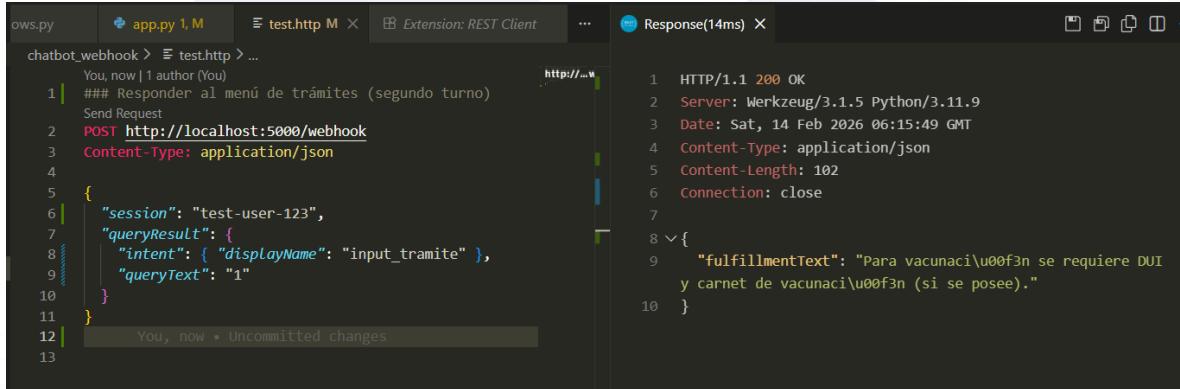
Response(9ms) X

```

1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.5 Python/3.11.9
3 Date: Sat, 14 Feb 2026 06:12:25 GMT
4 Content-Type: application/json
5 Content-Length: 247
6 Connection: close
7
8 {
9     "fulfillmentText": "Claro \ud83d\ude0a \u00bfQu\u00e9 tr\u00e1mite deseas consultar?\n\n1. Vacunaci\u00f3n prenatal\n2. Control prenatal\n3. Referencia\n\nPuedes responder con el n\u00famero o el nombre del tr\u00e1mite."
10 }

```

Evidencia 3: El bot responde satisfactoriamente el trámite específico



```

ws.py app.py 1, M test.http M Extension: REST Client ...
chatbot_webhook > test.http > ...
You, now | 1 author (You)
    ### Responder al menú de trámites (segundo turno)
Send Request
2 POST http://localhost:5000/webhook
Content-Type: application/json
4
{
    "session": "test-user-123",
    "queryResult": {
        "intent": {
            "displayName": "input_tramite"
        },
        "queryText": "1"
    }
}

```

You, now * Uncommitted changes

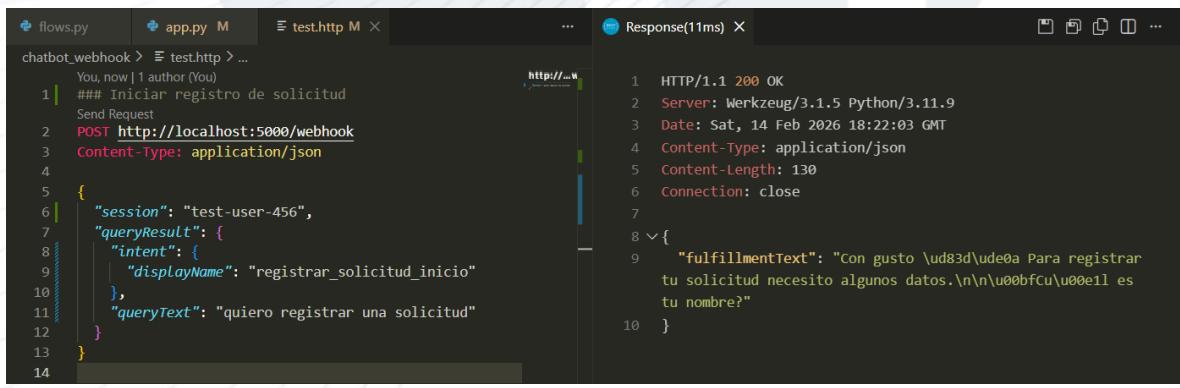
Response(14ms) X

```

1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.5 Python/3.11.9
3 Date: Sat, 14 Feb 2026 06:15:49 GMT
4 Content-Type: application/json
5 Content-Length: 102
6 Connection: close
7
8 {
9     "fulfillmentText": "Para vacunaci\u00f3n se requiere DUI y carnet de vacunaci\u00f3n (si se posee)."
10 }

```

Evidencia 4: Iniciar flujo registro_solicitud



```

ws.py app.py M test.http M Extension: REST Client ...
chatbot_webhook > test.http > ...
You, now | 1 author (You)
    ### Iniciar registro de solicitud
Send Request
2 POST http://localhost:5000/webhook
Content-Type: application/json
4
{
    "session": "test-user-456",
    "queryResult": {
        "intent": {
            "displayName": "registrar_solicitud_inicio"
        },
        "queryText": "quiero registrar una solicitud"
    }
}

```

Response(11ms) X

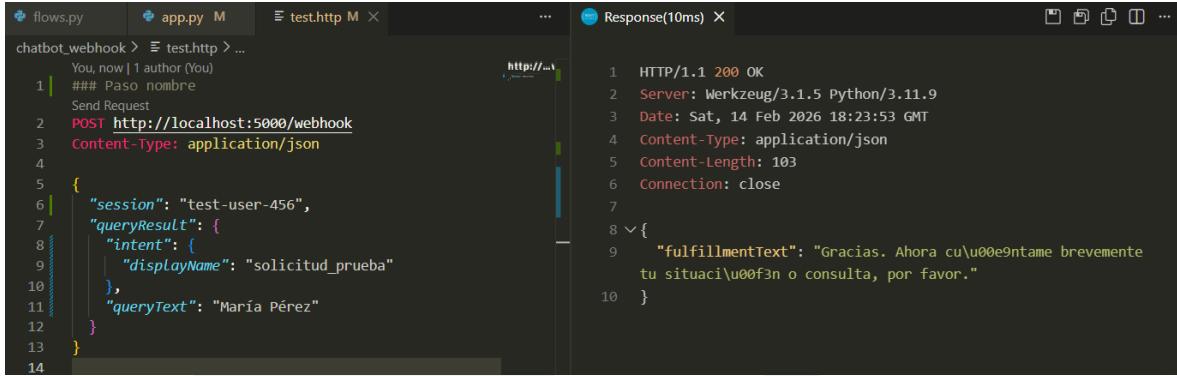
```

1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.5 Python/3.11.9
3 Date: Sat, 14 Feb 2026 18:22:03 GMT
4 Content-Type: application/json
5 Content-Length: 130
6 Connection: close
7
8 {
9     "fulfillmentText": "con gusto \ud83d\ude0a Para registrar tu solicitud necesito algunos datos.\n\n\u00bfCual es tu nombre?"
10 }

```



Evidencia 5: Almacenar nombre y solicitar descripción



The terminal shows the execution of a Python script named `flows.py`. The script sends a POST request to `http://localhost:5000/webhook` with the following JSON payload:

```

chatbot_webhook > test.http > ...
You, now | 1 author (You)
### Paso nombre
Send Request
POST http://localhost:5000/webhook
Content-Type: application/json

{
    "session": "test-user-456",
    "queryResult": {
        "intent": {
            "displayName": "solicitud_prueba"
        },
        "queryText": "María Pérez"
    }
}

```

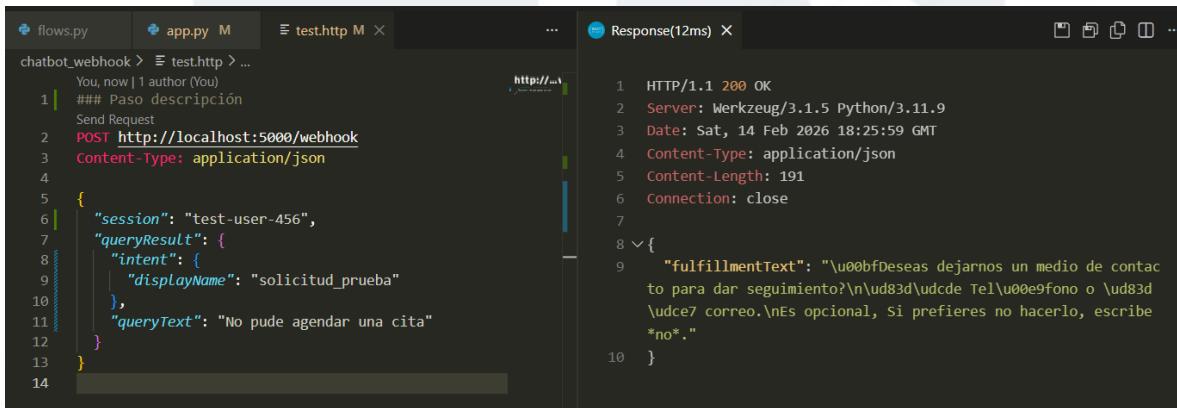
The response is a `HTTP/1.1 200 OK` with the following headers and fulfillment text:

```

1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.5 Python/3.11.9
3 Date: Sat, 14 Feb 2026 18:23:53 GMT
4 Content-Type: application/json
5 Content-Length: 103
6 Connection: close
7
8 {
9     "fulfillmentText": "Gracias. Ahora cuéntame brevemente tu situación o consulta, por favor."
10 }

```

Evidencia 6: Solicitar información de contacto (opcional)



The terminal shows the execution of a Python script named `flows.py`. The script sends a POST request to `http://localhost:5000/webhook` with the following JSON payload:

```

chatbot_webhook > test.http > ...
You, now | 1 author (You)
### Paso descripción
Send Request
POST http://localhost:5000/webhook
Content-Type: application/json

{
    "session": "test-user-456",
    "queryResult": {
        "intent": {
            "displayName": "solicitud_prueba"
        },
        "queryText": "No pude agendar una cita"
    }
}

```

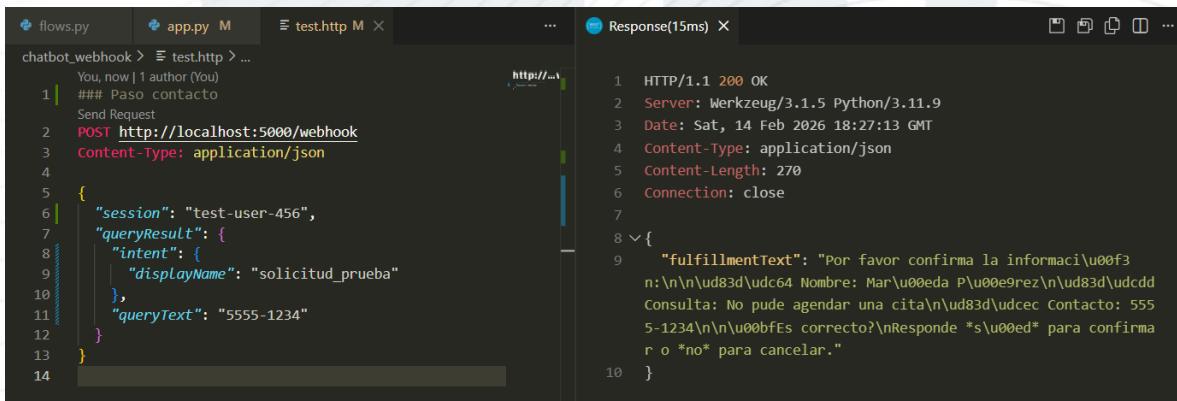
The response is a `HTTP/1.1 200 OK` with the following headers and fulfillment text:

```

1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.5 Python/3.11.9
3 Date: Sat, 14 Feb 2026 18:25:59 GMT
4 Content-Type: application/json
5 Content-Length: 191
6 Connection: close
7
8 {
9     "fulfillmentText": "\u00bfDeseas dejarnos un medio de contacto para dar seguimiento?\n\nNombre: Mar\u00eda P\u00f3rez\nTel\u00edfono o \ud83d\udcde correo.\nEs opcional, si prefieres no hacerlo, escribe *no*."
10 }

```

Evidencia 7: Solicitar confirmación de solicitud



The terminal shows the execution of a Python script named `flows.py`. The script sends a POST request to `http://localhost:5000/webhook` with the following JSON payload:

```

chatbot_webhook > test.http > ...
You, now | 1 author (You)
### Paso contacto
Send Request
POST http://localhost:5000/webhook
Content-Type: application/json

{
    "session": "test-user-456",
    "queryResult": {
        "intent": {
            "displayName": "solicitud_prueba"
        },
        "queryText": "5555-1234"
    }
}

```

The response is a `HTTP/1.1 200 OK` with the following headers and fulfillment text:

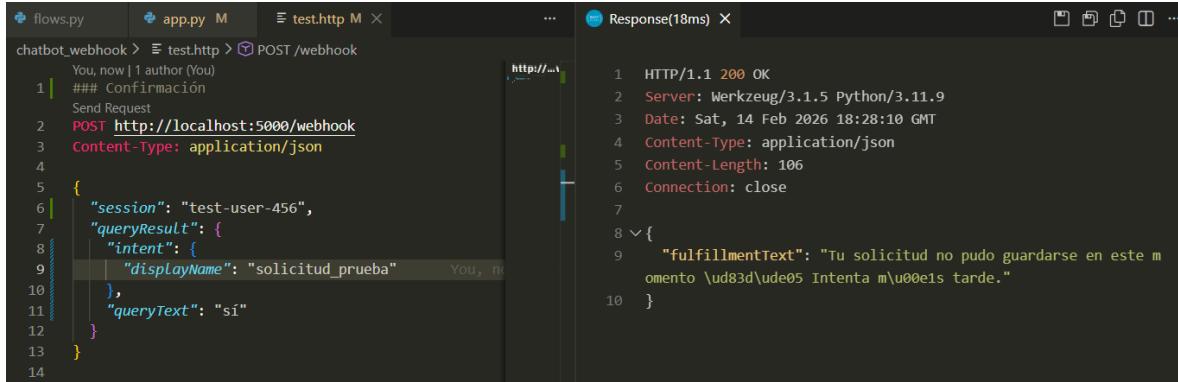
```

1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.5 Python/3.11.9
3 Date: Sat, 14 Feb 2026 18:27:13 GMT
4 Content-Type: application/json
5 Content-Length: 270
6 Connection: close
7
8 {
9     "fulfillmentText": "Por favor confirma la informaci\u00f3n:\nNombre: Mar\u00eda P\u00f3rez\nTel\u00edfono o \ud83d\udc64 Consulta: No pude agendar una cita\nContacto: 555-5-1234\nEs correcto?\nResponde *s\u00f3lo* para confirmar o *no* para cancelar."
10 }

```



Evidencia 8: Fallback indicando que no se pudo almacenar los datos de la solicitud



flows.py

```

chatbot_webhook > test.http > POST /webhook
    You, now | 1 author (You)
    1 |     ### Confirmación
    2 |     Send Request
    3 |     POST http://localhost:5000/webhook
    4 |     Content-Type: application/json
    5 |
    6 |     {
    7 |         "session": "test-user-456",
    8 |         "queryResult": {
    9 |             "intent": {
   10 |                 "displayName": "solicitud_prueba"
   11 |             },
   12 |             "queryText": "sí"
   13 |         }
   14 |

```

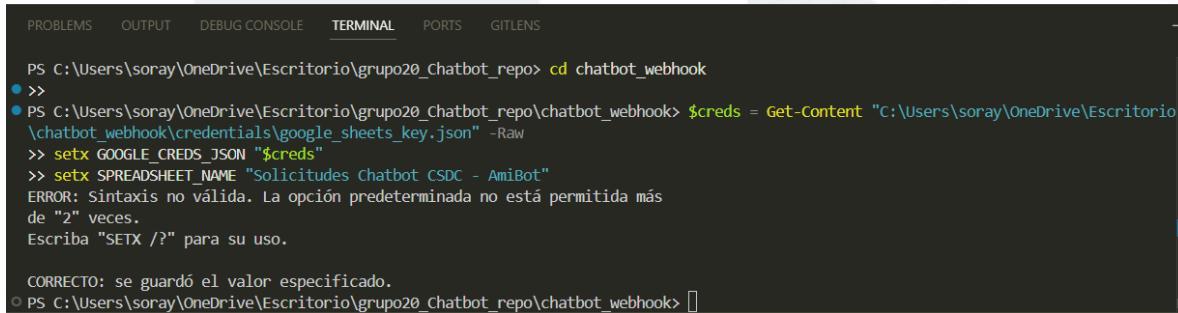
Response(18ms)

```

1  HTTP/1.1 200 OK
2  Server: Werkzeug/3.1.5 Python/3.11.9
3  Date: Sat, 14 Feb 2026 18:28:10 GMT
4  Content-Type: application/json
5  Content-Length: 106
6  Connection: close
7
8  {
9      "fulfillmentText": "Tu solicitud no pudo guardarse en este momento \ud83d\ude05 Intenta más tarde."
10 }

```

Evidencia 9: Terminal mostrando conexión a la ruta de la carpeta con credenciales y error de limitaciones de SETX



```

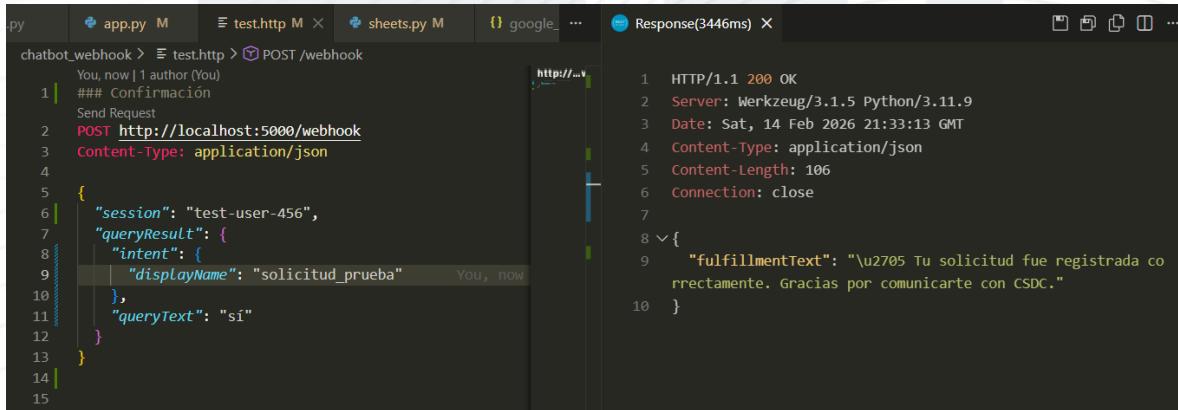
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\soray\OneDrive\Escritorio\grupo20_chatbot_repo> cd chatbot_webhook
>>
● PS C:\Users\soray\OneDrive\Escritorio\grupo20_chatbot_repo\chatbot_webhook> $creds = Get-Content "C:\Users\soray\OneDrive\Escritorio\chatbot_webhook\credentials\google_sheets_key.json" -Raw
>> setx GOOGLE_CREDS_JSON "$creds"
>> setx SPREADSHEET_NAME "Solicitudes Chatbot CSDC - AmiBot"
ERROR: Sintaxis no válida. La opción predeterminada no está permitida más de "2" veces.
Escriba "SETX /?" para su uso.

CORRECTO: se guardó el valor especificado.
○ PS C:\Users\soray\OneDrive\Escritorio\grupo20_chatbot_repo\chatbot_webhook>

```

Evidencia 10: Respuesta satisfactoria después del fix.



py app.py M test.http M sheets.py M google...

chatbot_webhook > test.http > POST /webhook

```

    You, now | 1 author (You)
    1 |     ### Confirmación
    2 |     Send Request
    3 |     POST http://localhost:5000/webhook
    4 |     Content-Type: application/json
    5 |
    6 |     {
    7 |         "session": "test-user-456",
    8 |         "queryResult": {
    9 |             "intent": {
   10 |                 "displayName": "solicitud_prueba"
   11 |             },
   12 |             "queryText": "sí"
   13 |         }
   14 |

```

Response(3446ms)

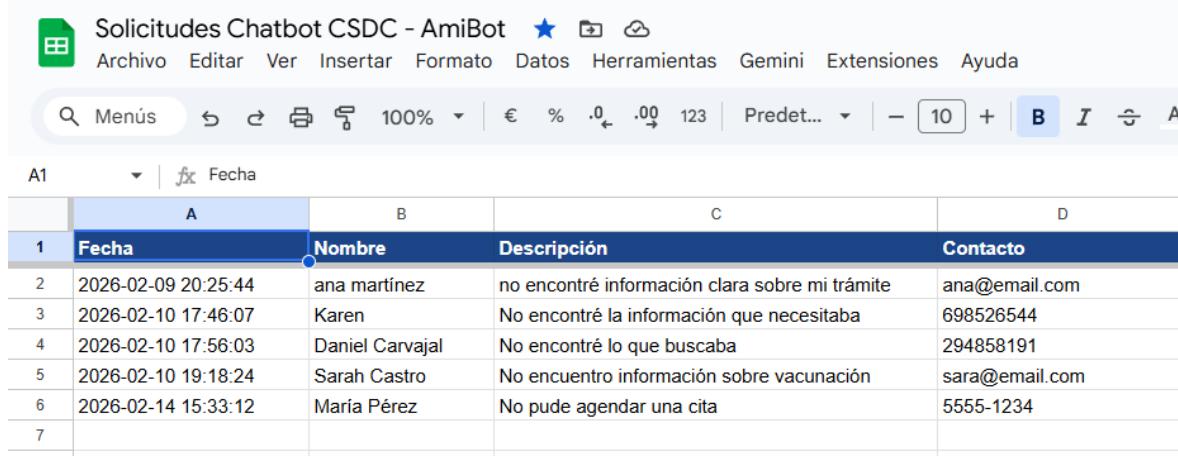
```

1  HTTP/1.1 200 OK
2  Server: Werkzeug/3.1.5 Python/3.11.9
3  Date: Sat, 14 Feb 2026 21:33:13 GMT
4  Content-Type: application/json
5  Content-Length: 106
6  Connection: close
7
8  {
9      "fulfillmentText": "\u2705 Tu solicitud fue registrada correctamente. Gracias por comunicarte con CSDC."
10 }

```



Evidencia 11: Solicitud registrada en Google Sheets

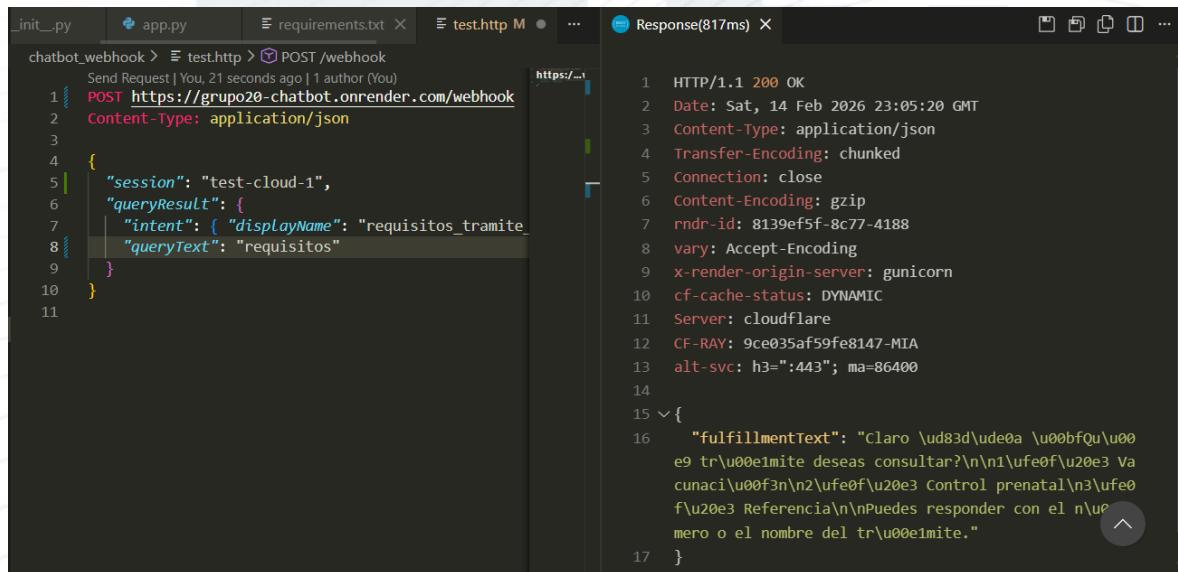


	A	B	C	D
1	Fecha	Nombre	Descripción	Contacto
2	2026-02-09 20:25:44	ana martínez	no encontré información clara sobre mi trámite	ana@email.com
3	2026-02-10 17:46:07	Karen	No encontré la información que necesitaba	698526544
4	2026-02-10 17:56:03	Daniel Carvajal	No encontré lo que buscaba	294858191
5	2026-02-10 19:18:24	Sarah Castro	No encuentro información sobre vacunación	sara@email.com
6	2026-02-14 15:33:12	Maria Pérez	No pude agendar una cita	5555-1234
7				

Pruebas en REST (Entorno Cloud)

Se ejecutaron pruebas REST sobre el backend desplegado en la nube para verificar la disponibilidad del servicio y la correcta operación del webhook del chatbot en un entorno productivo simulado.

Evidencia 12: Iniciar flujo de requisitos_tramite



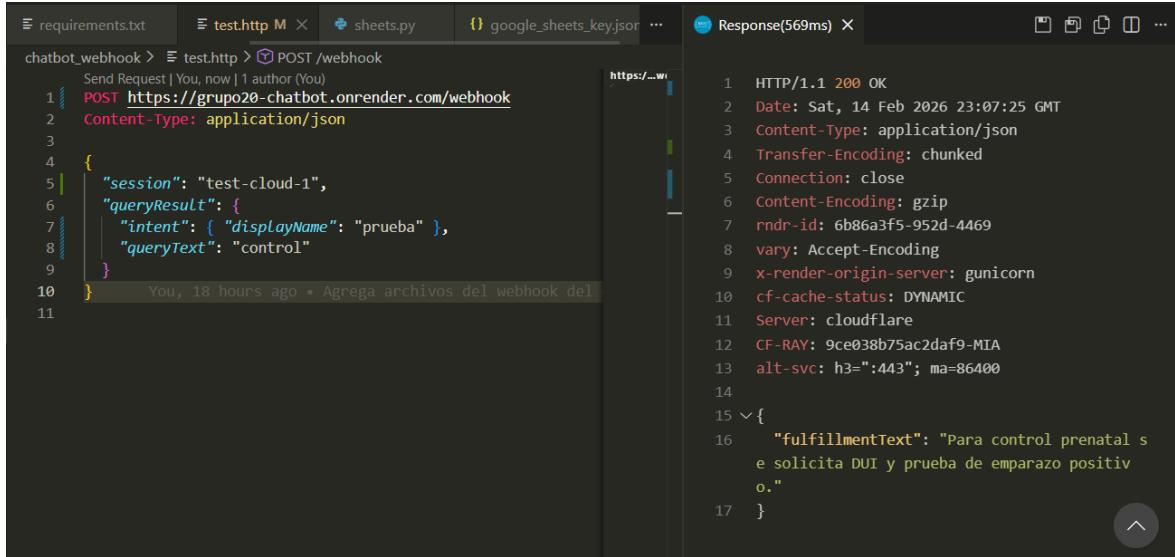
```

chatbot_webhook > test.http > POST /webhook
Send Request | You, 21 seconds ago | 1 author (You)
1 POST https://grupo20-chatbot.onrender.com/webhook
2 Content-Type: application/json
3
4 {
5   "session": "test-cloud-1",
6   "queryResult": {
7     "intent": { "displayName": "requisitos_tramite" },
8     "queryText": "requisitos"
9   }
10 }
11
https://.../...
1  HTTP/1.1 200 OK
2  Date: Sat, 14 Feb 2026 23:05:20 GMT
3  Content-Type: application/json
4  Transfer-Encoding: chunked
5  Connection: close
6  Content-Encoding: gzip
7  rndr-id: 8139efff-8c77-4188
8  vary: Accept-Encoding
9  x-render-origin-server: gunicorn
10 cf-cache-status: DYNAMIC
11 Server: cloudflare
12 CF-RAY: 9ce035af59fe8147-MIA
13 alt-svc: h3=":443"; ma=86400
14
15 {
16   "fulfillmentText": "Claro \ud83d\ude0a \u00bfQu\u00f3  
eres tr\u00fame? deseas consultar?\n\n1\uufe0f\u20e3 Va  
cunaci\u00f3n\n2\uufe0f\u20e3 Control prenatal\n3\uufe0f  
f\u20e3 Referencia\nPuedes responder con el n\u00famero o el nombre del tr\u00fame."
17 }

```



Evidencia 13: El bot responde satisfactoriamente el trámite específico



```

chatbot_webhook > test.http > POST /webhook
Send Request | You, now | 1 author (You)
1 POST https://grupo20-chatbot.onrender.com/webhook
2 Content-Type: application/json
3
4 {
5   "session": "test-cloud-1",
6   "queryResult": {
7     "intent": { "displayName": "prueba" },
8     "queryText": "control"
9   }
10 } You, 18 hours ago • Agrega archivos del webhook del
11

```

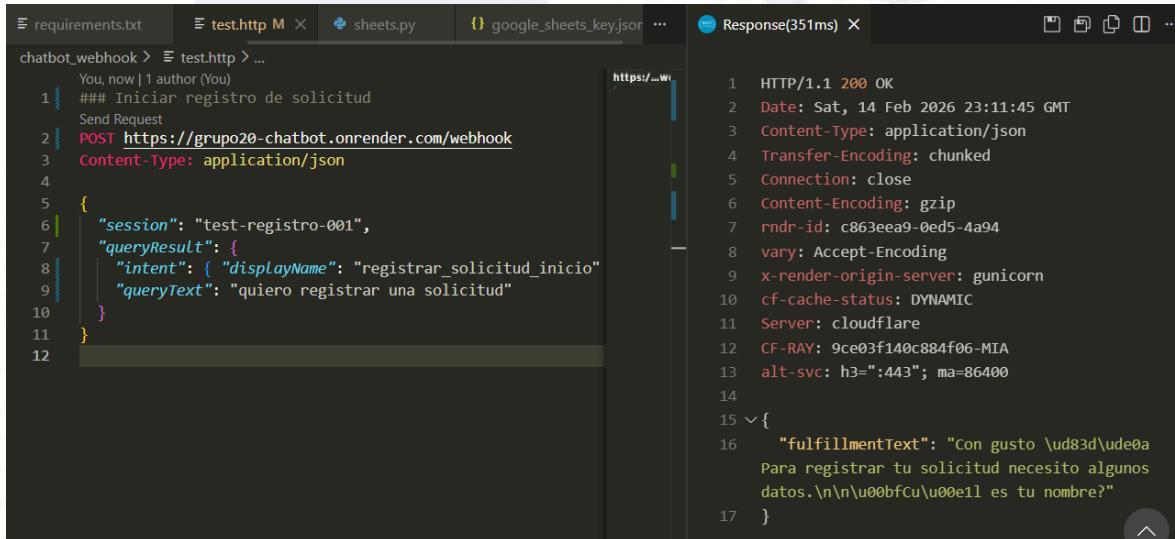
Response(569ms) X

```

1 HTTP/1.1 200 OK
2 Date: Sat, 14 Feb 2026 23:07:25 GMT
3 Content-Type: application/json
4 Transfer-Encoding: chunked
5 Connection: close
6 Content-Encoding: gzip
7 rndr-id: 6b86a3f5-952d-4469
8 vary: Accept-Encoding
9 x-render-origin-server: gunicorn
10 cf-cache-status: DYNAMIC
11 Server: cloudflare
12 CF-RAY: 9ce038b75ac2daf9-MIA
13 alt-svc: h3=":443"; ma=86400
14
15 {
16   "fulfillmentText": "Para control prenatal se solicita DUI y prueba de embarazo positivo."
17 }

```

Evidencia 14: Iniciar flujo registro_solicitud



```

chatbot_webhook > test.http > ...
You, now | 1 author (You)
1 ## Iniciar registro de solicitud
2 Send Request
3 POST https://grupo20-chatbot.onrender.com/webhook
4 Content-Type: application/json
5
6 {
7   "session": "test-registro-001",
8   "queryResult": {
9     "intent": { "displayName": "registrar_solicitud_inicio" },
10    "queryText": "quiero registrar una solicitud"
11  }
12

```

Response(351ms) X

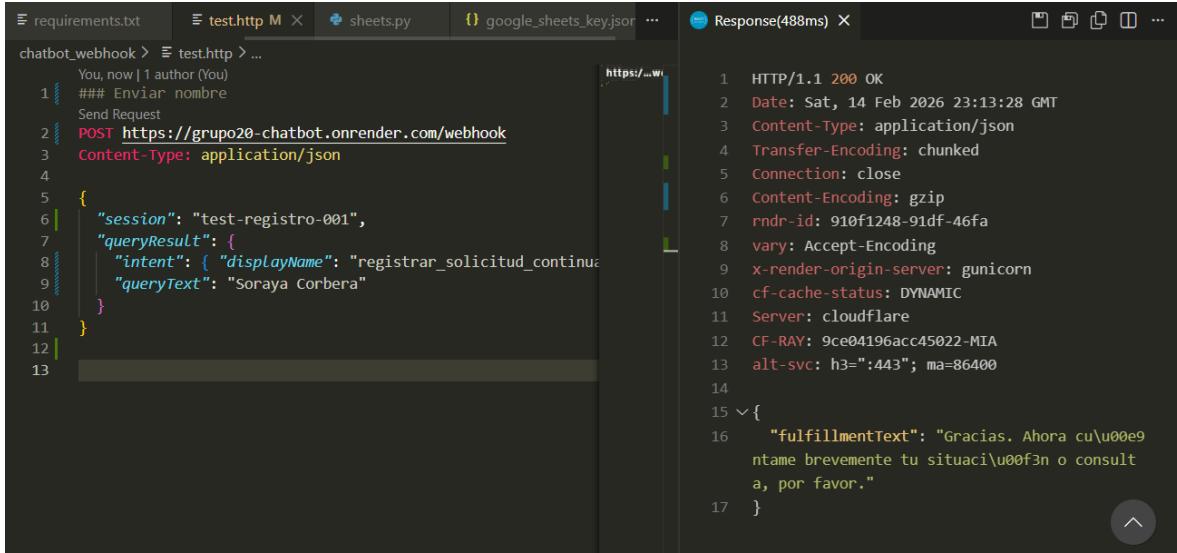
```

1 HTTP/1.1 200 OK
2 Date: Sat, 14 Feb 2026 23:11:45 GMT
3 Content-Type: application/json
4 Transfer-Encoding: chunked
5 Connection: close
6 Content-Encoding: gzip
7 rndr-id: c863eeaa-0ed5-4a94
8 vary: Accept-Encoding
9 x-render-origin-server: gunicorn
10 cf-cache-status: DYNAMIC
11 Server: cloudflare
12 CF-RAY: 9ce03f140c884f06-MIA
13 alt-svc: h3=":443"; ma=86400
14
15 {
16   "fulfillmentText": "Con gusto \ud83d\ude0a Para registrar tu solicitud necesito algunos datos.\n\n\u00bfC\u00f3mo es tu nombre?"
17 }

```



Evidencia 15: Almacenar nombre y solicitar descripción



```

chatbot_webhook > test.http ...
You, now | 1 author (You)
### Enviar nombre
Send Request
POST https://grupo20-chatbot.onrender.com/webhook
Content-Type: application/json

{
  "session": "test-registro-001",
  "queryResult": {
    "intent": { "displayName": "registrar_solicitud_continua" },
    "queryText": "Soraya Corbera"
  }
}

```

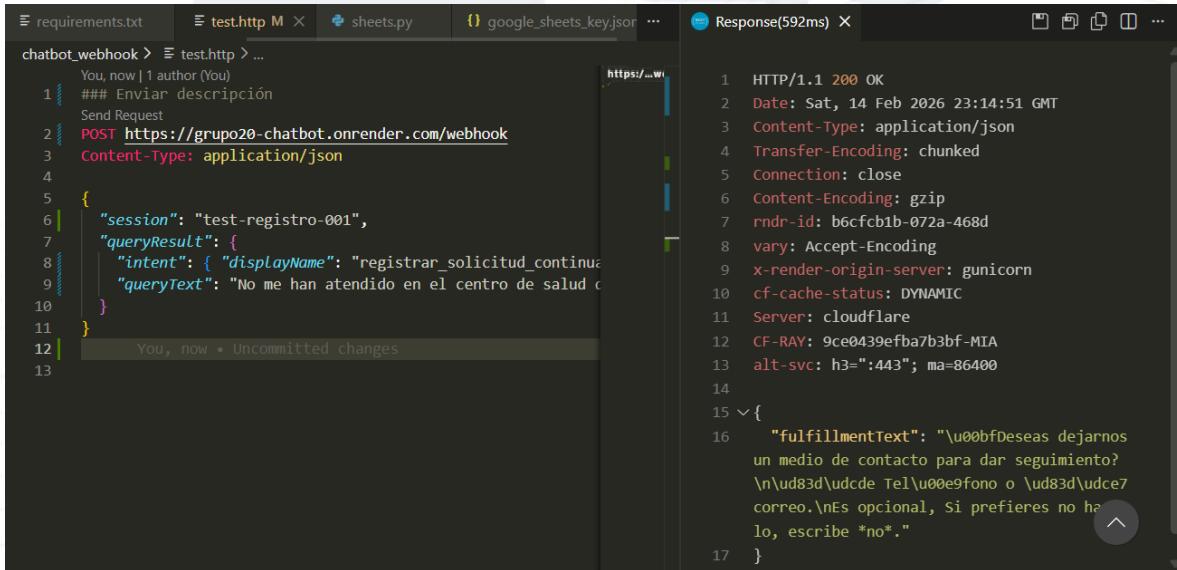
```

HTTP/1.1 200 OK
Date: Sat, 14 Feb 2026 23:13:28 GMT
Content-Type: application/json
Transfer-Encoding: chunked
Connection: close
Content-Encoding: gzip
rnrd-id: 910f1248-91df-46fa
vary: Accept-Encoding
x-render-origin-server: gunicorn
cf-cache-status: DYNAMIC
Server: cloudflare
CF-RAY: 9ce04196acc45022-MIA
alt-svc: h3=":443"; ma=86400
15 ▼ {
  "fulfillmentText": "Gracias. Ahora cu\u00f1ntame brevemente tu situaci\u00f3n o consulta, por favor."
16 }

17 }

```

Evidencia 16: Solicitar información de contacto (opcional)



```

chatbot_webhook > test.http ...
You, now | 1 author (You)
### Enviar descripción
Send Request
POST https://grupo20-chatbot.onrender.com/webhook
Content-Type: application/json

{
  "session": "test-registro-001",
  "queryResult": {
    "intent": { "displayName": "registrar_solicitud_continua" },
    "queryText": "No me han atendido en el centro de salud"
  }
}

```

```

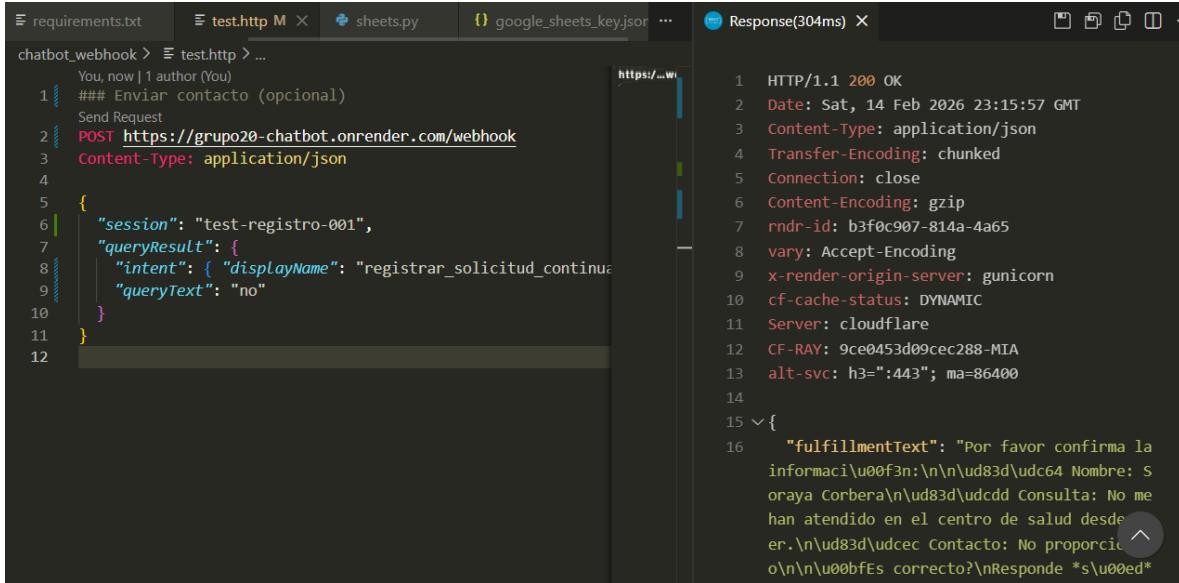
HTTP/1.1 200 OK
Date: Sat, 14 Feb 2026 23:14:51 GMT
Content-Type: application/json
Transfer-Encoding: chunked
Connection: close
Content-Encoding: gzip
rnrd-id: b6cfcbb1-072a-468d
vary: Accept-Encoding
x-render-origin-server: gunicorn
cf-cache-status: DYNAMIC
Server: cloudflare
CF-RAY: 9ce0439efba7b3bf-MIA
alt-svc: h3=":443"; ma=86400
15 ▼ {
  "fulfillmentText": "\u00bfDeseas dejarnos un medio de contacto para dar seguimiento?\n\nTel\u00e9fono o correo.\u00bf Es opcional, si prefieres no hacerlo, escribe *no*."
16 }

17 }

```



Evidencia 17: Solicitar confirmación de solicitud



```

chatbot_webhook > test.http ...
You, now | 1 author (You)
1   ### Enviar contacto (opcional)
Send Request
2   POST https://grupo20-chatbot.onrender.com/webhook
3   Content-Type: application/json
4
5   {
6     "session": "test-registro-001",
7     "queryResult": {
8       "intent": { "displayName": "registrar_solicitud_continua" },
9       "queryText": "no"
10    }
11  }
12

```

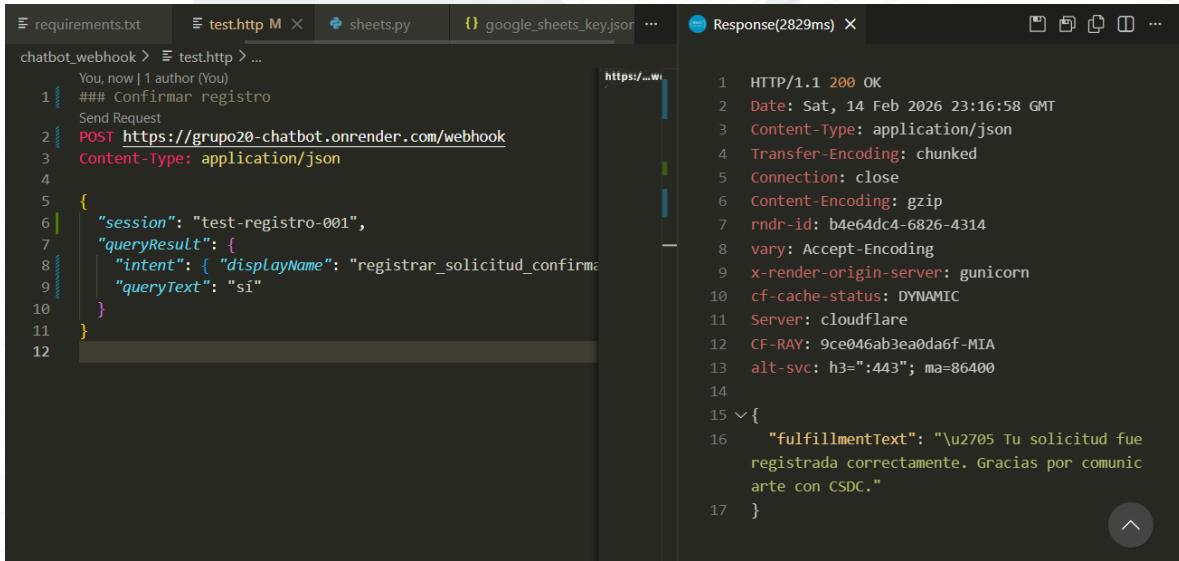
Response(304ms) X

```

1   HTTP/1.1 200 OK
2   Date: Sat, 14 Feb 2026 23:15:57 GMT
3   Content-Type: application/json
4   Transfer-Encoding: chunked
5   Connection: close
6   Content-Encoding: gzip
7   rndr-id: b3f0c907-814a-4a65
8   vary: Accept-Encoding
9   x-render-origin-server: gunicorn
10  cf-cache-status: DYNAMIC
11  Server: cloudflare
12  CF-RAY: 9ce0453d09cec288-MIA
13  alt-svc: h3=":443"; ma=86400
14
15  {
16    "fulfillmentText": "Por favor confirma la informaci\u00f3n:\nNombre: Soraya Corbera\nConsulta: No me han atendido en el centro de salud desde ayer.\nContacto: No proporcion\u00f3\no\nResponde *s\u00ed"
17  }

```

Evidencia 18: Mensaje de confirmación de registro



```

chatbot_webhook > test.http ...
You, now | 1 author (You)
1   ### Confirmar registro
Send Request
2   POST https://grupo20-chatbot.onrender.com/webhook
3   Content-Type: application/json
4
5   {
6     "session": "test-registro-001",
7     "queryResult": {
8       "intent": { "displayName": "registrar_solicitud_confirmacion" },
9       "queryText": "s\u00ed"
10    }
11  }
12

```

Response(2829ms) X

```

1   HTTP/1.1 200 OK
2   Date: Sat, 14 Feb 2026 23:16:58 GMT
3   Content-Type: application/json
4   Transfer-Encoding: chunked
5   Connection: close
6   Content-Encoding: gzip
7   rndr-id: b4e64dc4-6826-4314
8   vary: Accept-Encoding
9   x-render-origin-server: gunicorn
10  cf-cache-status: DYNAMIC
11  Server: cloudflare
12  CF-RAY: 9ce046ab3ea0da6f-MIA
13  alt-svc: h3=":443"; ma=86400
14
15  {
16    "fulfillmentText": "\u2705 Tu solicitud fue registrada correctamente. Gracias por comunicarte con CSDC."
17  }

```



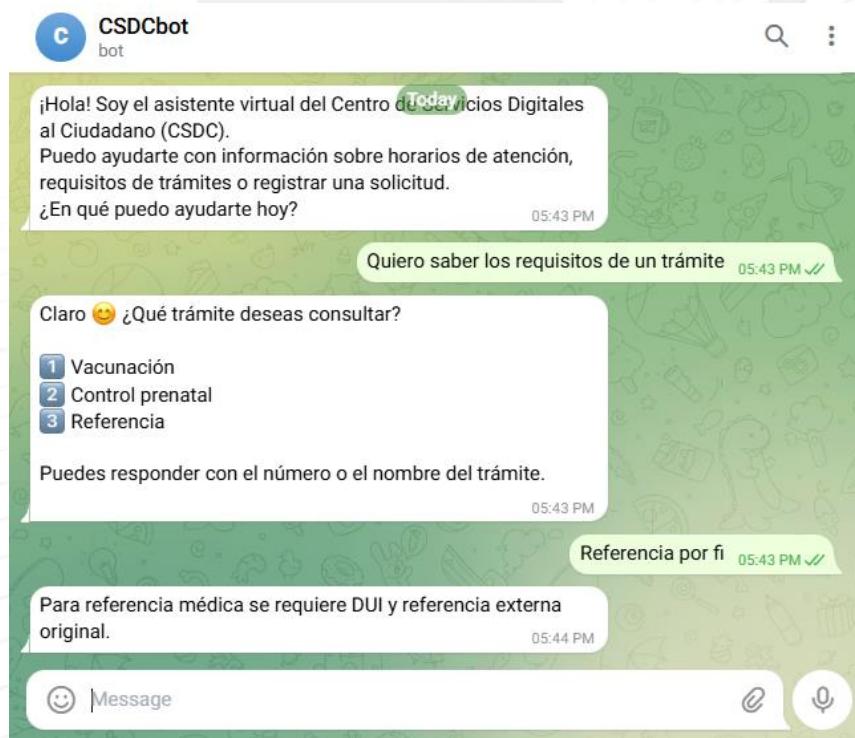
Evidencia 19: Solicitud registrada en Google Sheets

	A	B	C	D
1	Fecha	Nombre	Descripción	Contacto
2	2026-02-09 20:25:44	ana martínez	no encontré información clara sobre mi trámite	ana@email.com
3	2026-02-10 17:46:07	Karen	No encontré la información que necesitaba	698526544
4	2026-02-10 17:56:03	Daniel Carvajal	No encontré lo que buscaba	294858191
5	2026-02-10 19:18:24	Sarah Castro	No encuentro información sobre vacunación	sara@email.com
6	2026-02-14 15:33:12	Maria Pérez	No pude agendar una cita	5555-1234
7	2026-02-14 23:16:58	Soraya Corbera	No me han atendido en el centro de salud desde ayer.	No proporcionado

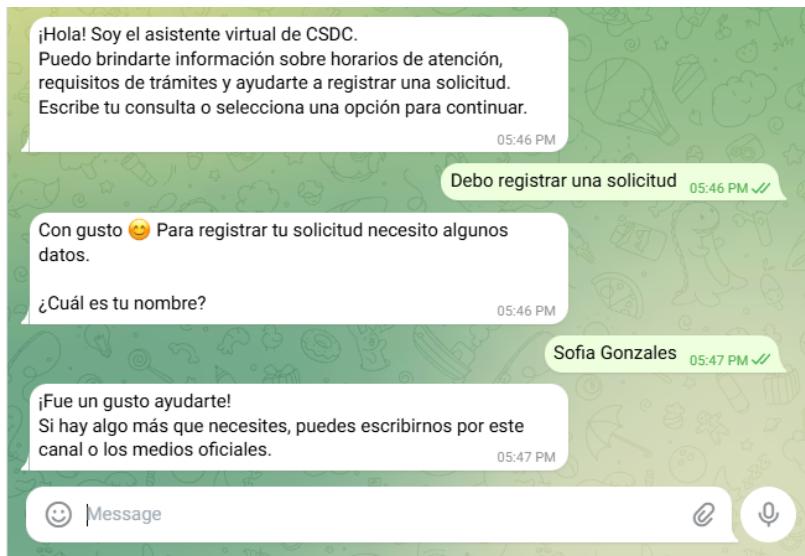
Pruebas en Telegram

Se ejecutaron pruebas del chatbot en Telegram para evaluar la experiencia conversacional desde el canal de mensajería, confirmando la continuidad de los flujos y la correcta respuesta del sistema ante diferentes entradas del usuario.

Evidencia 20: Interacción satisfactoria (requisito_tramite)



Evidencia 21: Error en flujo de conversación



Pruebas en Dialogflow

Se ejecutaron pruebas directamente en Dialogflow para identificar la causa de la interrupción de flujos por otros intents y entrenar el agente con las correcciones para lograr una conversación coherente.

Evidencia 22: Ajustes de ML para reducir el margen de error al identificar intenciones

General Languages **ML Settings** Export and Import Environments Speech Share Advanced

ML CLASSIFICATION THRESHOLD

Define the threshold value for the confidence score. If the returned value is less than the threshold value, then a fallback intent will be triggered, or if there is no fallback intents defined, no intent will be triggered.

0.5

AUTOMATIC SPELL CORRECTION

Allow ML to correct spelling of query during request processing.

AUTOMATIC TRAINING

Disable automatic re-training the agent after every agent modification.

Automatic training may slow UI responsiveness and is not recommended for large agents. You can kick off agent training manually through [API](#) or by clicking the 'TRAIN' button below.

AGENT VALIDATION

Automatically validate the agent when agent training is performed.

TRAIN



SECRETARÍA
DE INNOVACIÓN

ESIT - TSU EN SERVICIOS EN LA NUBE

Evidencia 23: Iniciar flujo registrar_solicitud

Agent

USER SAYS COPY CURL

Debo registrar una solicitud

DEFAULT RESPONSE ▾

Con gusto 😊 Para registrar tu solicitud necesito algunos datos. ¿Cuál es tu nombre?

INTENT

registrar_solicitud_inicio

Evidencia 24: Intención incorrecta identificada

Agent

USER SAYS COPY CURL

Sofia Gonzales

DEFAULT RESPONSE ▾

¡Fue un gusto ayudarte! Si hay algo más que necesites, puedes escribirnos por este canal o los medios oficiales.

INTENT

despedida

ACTION

Not available

PARAMETER VALUE

music-artist Sofia Gonzales

location



Evidencia 25: Intent “despedida” después de eliminar todos los parámetros

- despedida

SAVE



Action and parameters

Action and parameters				
Enter action name				
REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

Evidencia 26: Respuesta correcta en el flujo de registro

Agent

USER SAYS [COPY CURL](#)
Sofia Gonzales

DEFAULT RESPONSE ▾
Gracias. Ahora cuéntame brevemente tu situación o consulta, por favor.

CONTEXTS [RESET CONTEXTS](#)

INTENT
[Default Fallback Intent](#)



Evidencia 27: Solicitar información de contacto

Agent

USER SAYS [COPY CURL](#)
No me dieron mis medicinas

DEFAULT RESPONSE ▾
¿Deseas dejarnos un medio de contacto para dar seguimiento? Teléfono o correo. Es opcional. Si prefieres no hacerlo, escribe *no*.

CONTEXTS [RESET CONTEXTS](#)
`_system_counters_`

INTENT
Default Fallback Intent

Evidencia 28: Mostrar datos capturados

Agent

USER SAYS [COPY CURL](#)
sofi@mail.org

DEFAULT RESPONSE ▾
Por favor confirma la información:
Nombre: Sofia Gonzales Consulta: No me dieron mis medicinas Contacto:
sofi@mail.org ¿Es correcto? Responde *si* para confirmar o *no* para cancelar.

CONTEXTS [RESET CONTEXTS](#)
`_system_counters_`

INTENT
Default Fallback Intent



Evidencia 29: Mensaje de confirmación

Agent

USER SAYS	COPY CURL
si	
💡 DEFAULT RESPONSE ▾ <input checked="" type="checkbox"/> Tu solicitud fue registrada correctamente. Gracias por comunicarte con CSDC.	
CONTEXTS RESET CONTEXTS <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">__system_counters__</div>	
INTENT Default Fallback Intent	

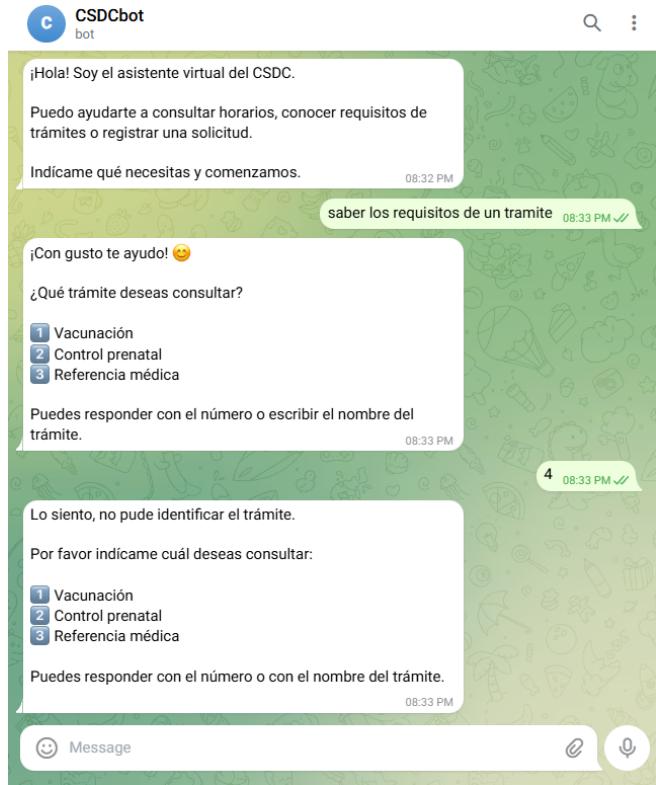
Evidencia 30: Solicitud registrada exitosamente en Google Sheets

	A	B	C	D
1	Fecha	Nombre	Descripción	Contacto
2	2026-02-09 20:25:44	ana martínez	no encontré información clara sobre mi trámite	ana@email.com
3	2026-02-10 17:46:07	Karen	No encontré la información que necesitaba	698526544
4	2026-02-10 17:56:03	Daniel Carvajal	No encontré lo que buscaba	294858191
5	2026-02-10 19:18:24	Sarah Castro	No encuentro información sobre vacunación	sara@email.com
6	2026-02-14 15:33:12	Maria Pérez	No pude agendar una cita	5555-1234
7	2026-02-14 23:16:58	Soraya Corbera	No me han atendido en el centro de salud desde ayer.	No proporcionado
8	2026-02-15 00:26:01	Sofia Gonzales	No me dieron mis medicinas	sofi@mail.org

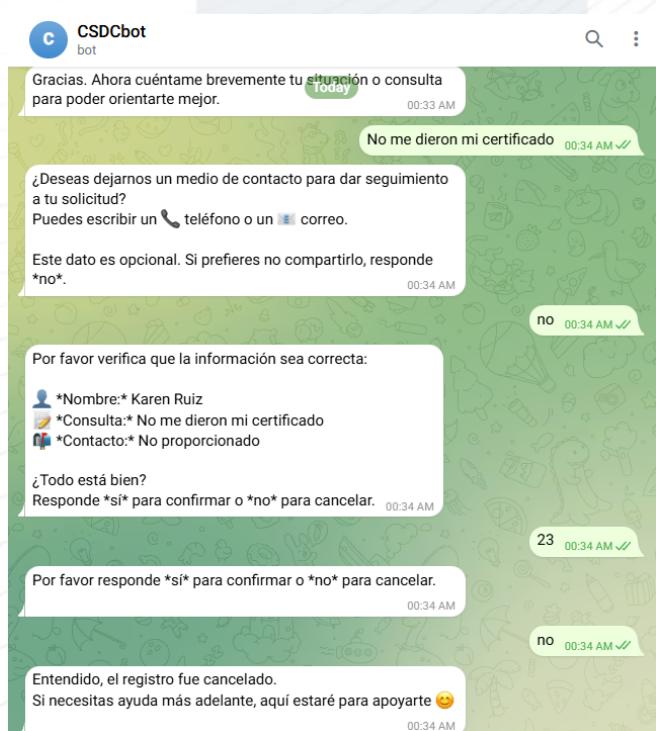
Flujos completos en Telegram

Evidencia de conversaciones con flujos coherentes con escenarios normales y de fallo, después de estabilizar los flujos completamente y mejorar los mensajes para el usuario.

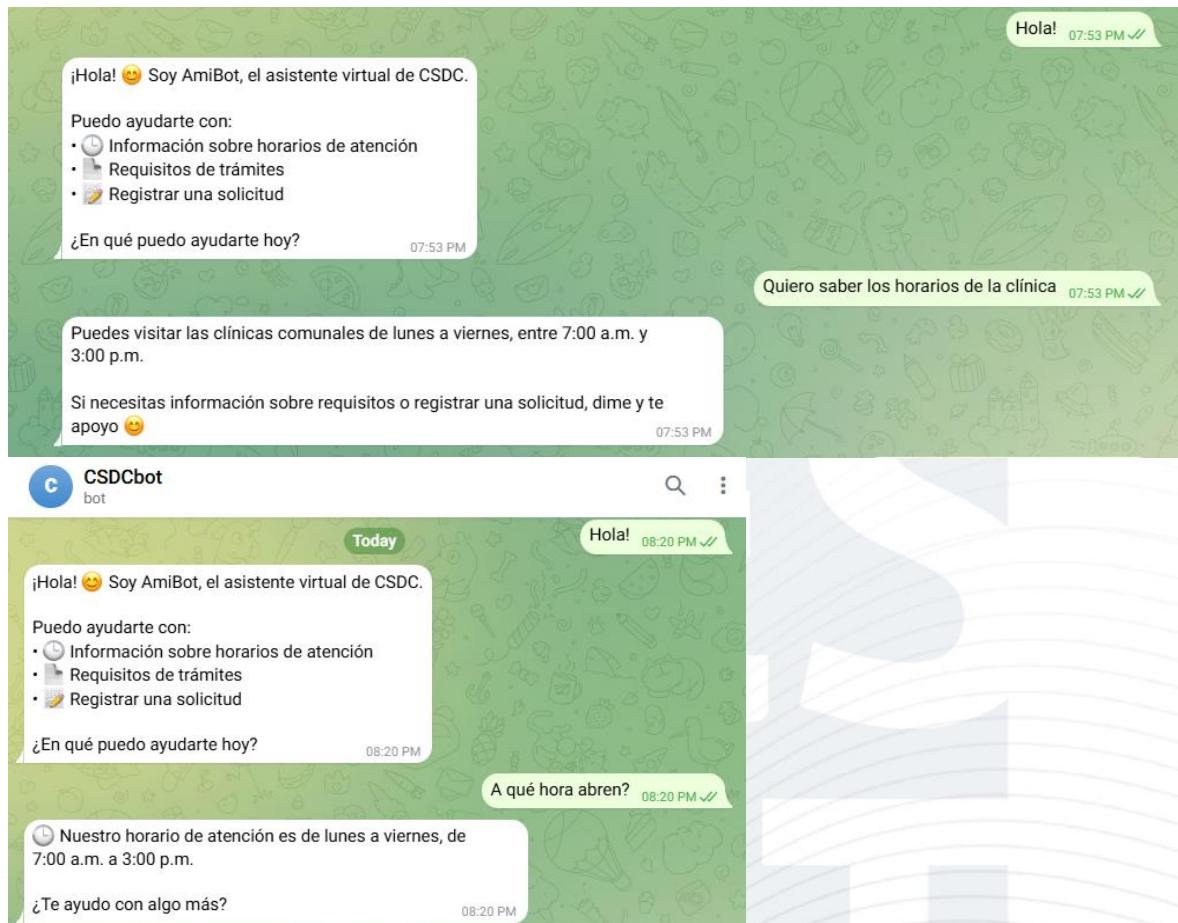
Evidencia 31: Caso de fallo en requisitos_tramite

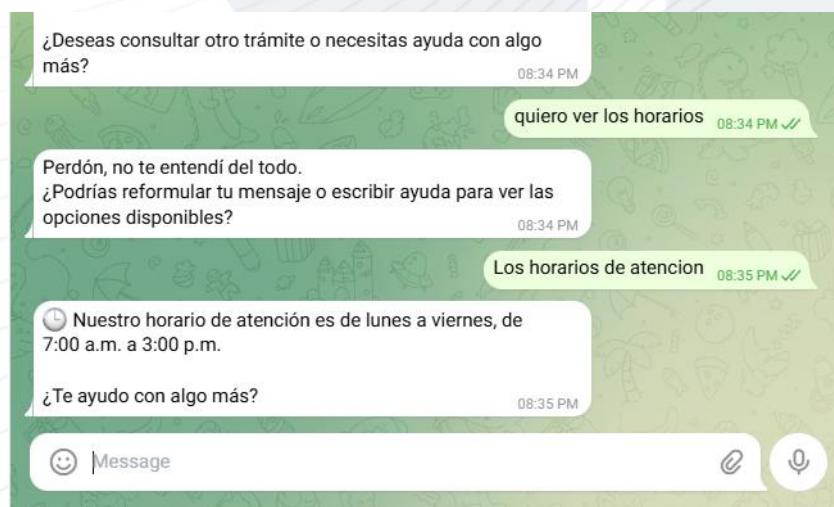
Evidencia 32: Caso de fallo en registrar_solicitud



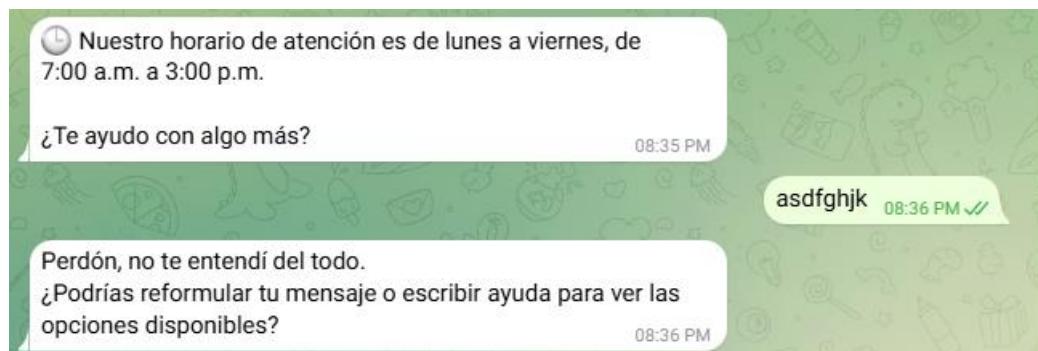
Evidencia 33: Respuestas a intención consultar_horario



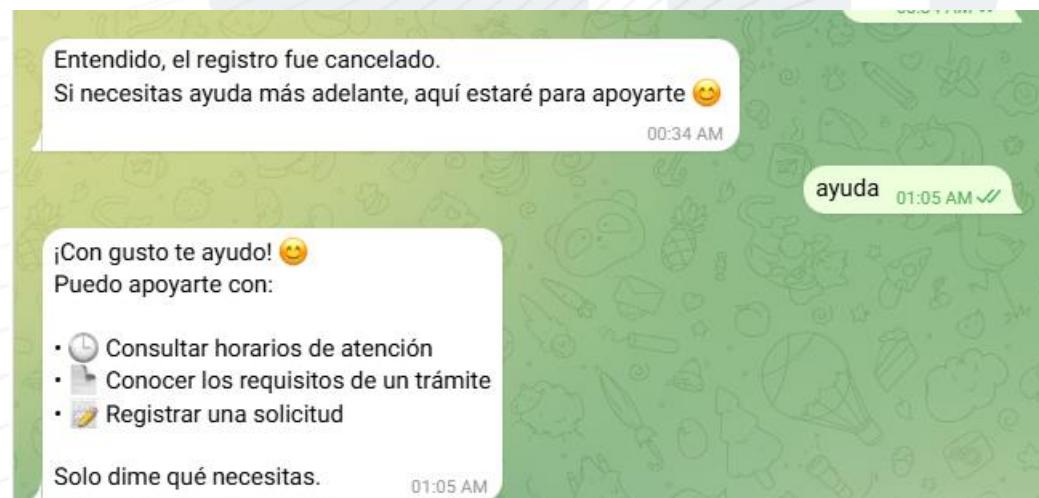
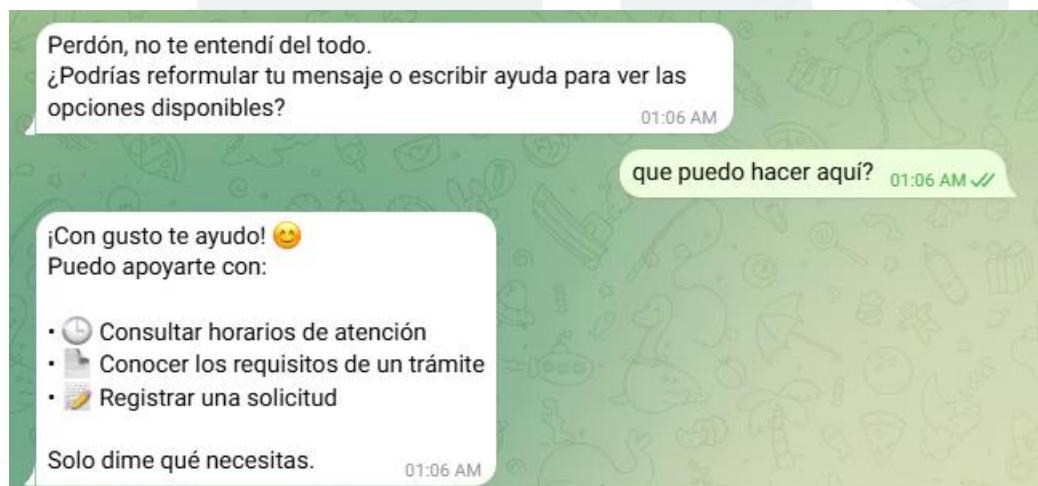
Evidencia 34: Intención Fallback y horarios



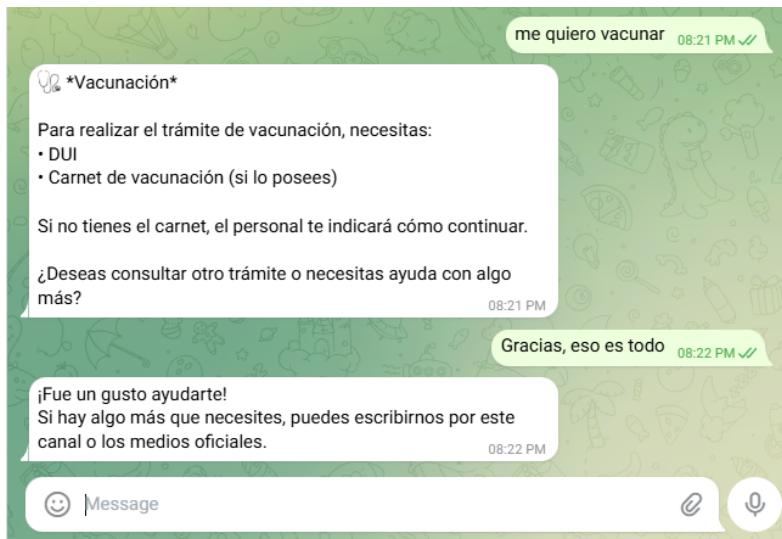
Evidencia 35: Fallback funcional



Evidencia 36: Intención ayuda activada



Evidencia 37: Despedida funcional



Conclusiones

1. En conclusión, la realización de pruebas en distintos entornos y herramientas permitió detectar inconsistencias que no habrían sido evidentes en un único escenario de validación. La comparación entre pruebas locales, en entorno cloud y en el canal de Telegram evidenció la importancia de validar la integración entre el backend, el motor de NLU y los canales de comunicación. Asimismo, esta fase reforzó la relevancia de aplicar buenas prácticas de seguridad en el manejo de credenciales y datos sensibles para un despliegue más robusto.
2. Las pruebas funcionales, acompañadas de una adecuada documentación, fueron clave para identificar fallos en la continuidad de los flujos y en la interpretación de intenciones del chatbot. El uso de matrices de prueba, evidencias y análisis de escenarios normales y de fallo permitió comprender con mayor claridad dónde se afectaba la estabilidad del sistema y por qué ocurrían dichas incidencias. Esta trazabilidad facilitó la aplicación de correcciones técnicas y la validación de mejoras, fortaleciendo el proceso de mejora continua del proyecto.

