



SECRETARÍA  
DE INNOVACIÓN

# Escuela Superior de Innovación y Tecnología



ESCUELA SUPERIOR  
de Innovación  
y Tecnología

Técnico Superior Universitario Servicios en la Nube

Estancia Profesional

## Entregable fase 3 Reporte de Resultados

Chatbot Asistente en la Nube para Atención Automatizada (Lite)

Docente: **Carlos Guillermo Rodriguez**

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

Introducción .....	2
1. Migración a entorno Cloud .....	3
1.1 Pipeline funcional .....	3
1.2 Componentes del sistema .....	3
2. Reporte de Resultados .....	4
2.1 Resultados por canal .....	5
2.2 Resultados según entorno .....	6
2.3 Resultados según flujo .....	7
2.4 Resultados Globales .....	9
Evidencias .....	10
Conclusiones .....	12

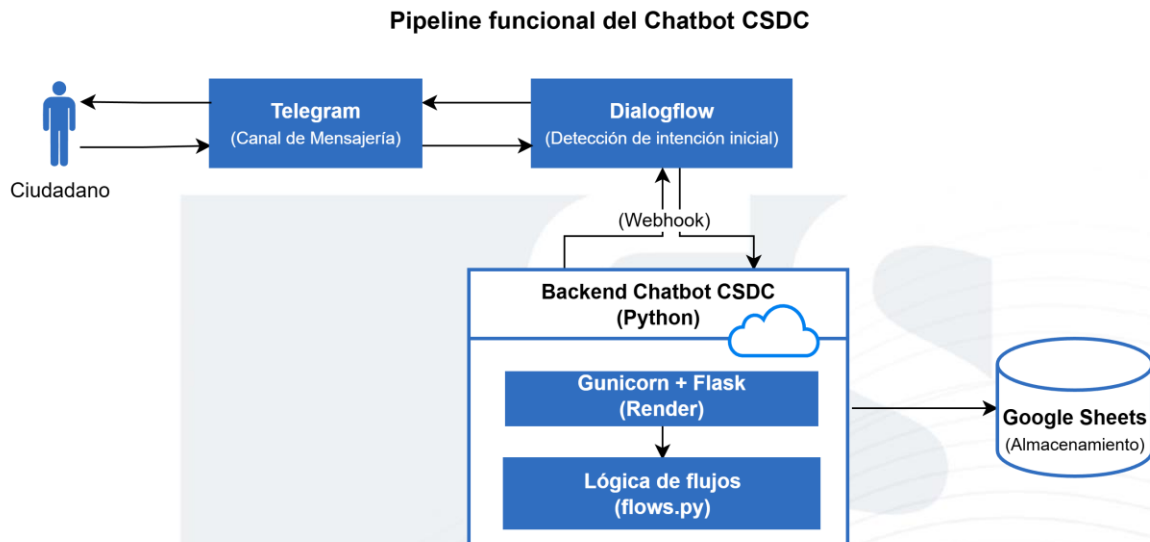
## Introducción

El presente documento tiene como objetivo principal presentar los resultados de las pruebas funcionales del chatbot CSDC, con énfasis en el análisis del desempeño del sistema a través de distintos canales, entornos y flujos conversacionales. El reporte consolida la evidencia generada durante la fase de validación, permitiendo evaluar la estabilidad alcanzada por el chatbot una vez implementadas las correcciones y ajustes técnicos correspondientes.

Como complemento al análisis de resultados, el documento incluye una descripción del pipeline funcional y de los componentes del sistema en su configuración actual, con el fin de contextualizar el funcionamiento general de la solución en el entorno Cloud. Esta información proporciona el marco técnico necesario para interpretar los resultados presentados y comprender cómo se articulan los distintos elementos del sistema en el entorno de operación definido.

# 1. Migración a entorno Cloud

## 1.1 Pipeline funcional



El pipeline funcional del chatbot integra el canal de mensajería Telegram con el motor de detección de intenciones Dialogflow y un backend desplegado en la nube mediante Flask, ejecutado sobre el servidor WSGI Gunicorn en la plataforma Render. A través del uso de webhooks, los mensajes del usuario son procesados por la lógica de flujos del sistema, permitiendo la ejecución de los flujos conversacionales definidos y el registro de solicitudes en Google Sheets como mecanismo de almacenamiento. Este pipeline valida la operación de extremo a extremo del sistema en un entorno cloud.

## 1.2 Componentes del sistema

A continuación, se presentan los principales componentes del sistema del chatbot, detallando su responsabilidad dentro de la arquitectura, los riesgos asociados a cada componente y las estrategias de mitigación consideradas durante la implementación.

Componente	Responsabilidad	Riesgo	Mitigación
<b>Telegram (Canal de mensajería)</b>	Recibir y enviar mensajes entre el ciudadano y el chatbot.	Caídas del servicio de Telegram o retrasos en entrega de mensajes.	Manejo de reintentos a nivel de plataforma y pruebas en entorno real para validar estabilidad.

<b>Dialogflow</b>	Detectar la intención del usuario y enrutar la solicitud al webhook.	Clasificación incorrecta de intenciones o interferencia de parámetros automáticos.	Ajuste de parámetros, limpieza de entidades no deseadas y pruebas de escenarios reales.
<b>Webhook Backend (Flask + Gunicorn)</b>	Procesar las solicitudes, ejecutar la lógica conversacional y responder al usuario.	Caídas del servicio en la nube o errores de ejecución en producción.	Despliegue en plataforma cloud estable, uso de Gunicorn para entorno productivo y pruebas REST en cloud.
<b>Google Sheets</b>	Almacenar las solicitudes registradas por los usuarios.	Fallos de autenticación, credenciales inválidas o indisponibilidad del servicio.	Manejo de excepciones al guardar, uso de credenciales seguras vía variables de entorno y mensajes de error controlados al usuario.
<b>Repositorio GitHub</b>	Versionar el código y documentar la evolución del proyecto.	Pérdida de cambios o conflictos de versiones.	Uso de control de versiones con commits frecuentes y repositorio centralizado.
<b>Render (Plataforma cloud)</b>	Alojar el backend del chatbot y exponer el endpoint público.	Latencia en instancias gratuitas o suspensión por inactividad.	Documentación de latencia esperada en free tier y validación funcional del pipeline en entorno cloud.

## 2. Reporte de Resultados

A continuación, se presenta una tabla consolidada de resultados orientada a evaluar el comportamiento de los flujos completos de intención del chatbot, considerando la correcta ejecución del flujo de principio a fin. El análisis detallado por paso conversacional, incluyendo la identificación del punto específico en el que se produce un fallo, se encuentra documentado en la matriz de pruebas correspondiente, contenida en un documento separado (Evidencia 1).

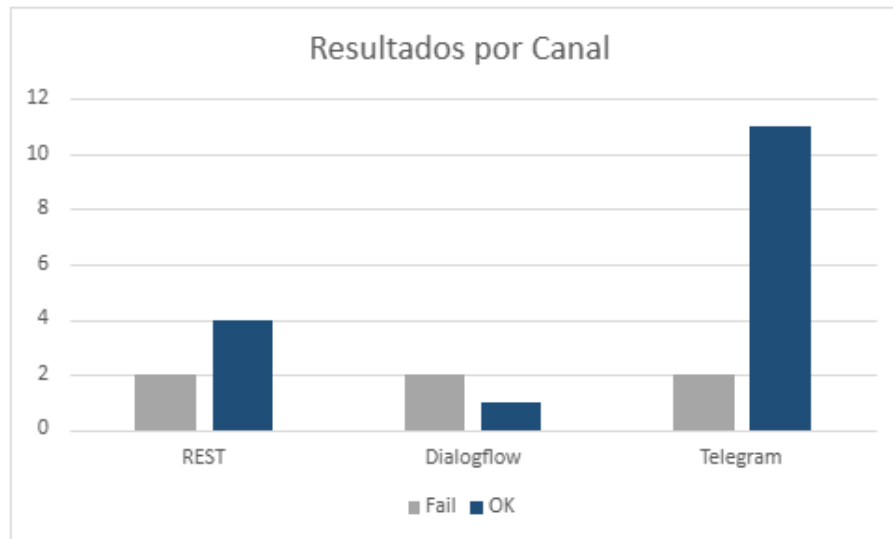
Entorno	Canal	Flujo	Resultado	Comentarios
Local	REST	requisitos_tramite	FAIL	Error en la simulación de Dialogflow
Local	REST	requisitos_tramite	OK	
Local	REST	registrar_solicitud	FAIL	Credenciales JSON no encontradas
Local	REST	registrar_solicitud	OK	Fix temporal: SETX
Cloud	REST	requisitos_tramite	OK	
Cloud	REST	registrar_solicitud	OK	Fix definitivo en Render
Cloud	Telegram	requisitos_tramite	OK	
Cloud	Telegram	registrar_solicitud	FAIL	Interrupción de flujo por intención equivocada
Cloud	Dialogflow	registrar_solicitud	FAIL	Interrupción de flujo por intención equivocada
Cloud	Dialogflow	registrar_solicitud	FAIL	Interrupción de flujo por intención equivocada
Cloud	Dialogflow	registrar_solicitud	OK	Fix: Eliminar parámetros y ajustar umbral de interpretación
Cloud	Telegram	Fallback	OK	
Cloud	Telegram	registrar_solicitud	OK	
Cloud	Telegram	Fallback	OK	
Cloud	Telegram	ayuda	OK	
Cloud	Telegram	ayuda	OK	
Cloud	Telegram	consultar_horario	OK	
Cloud	Telegram	consultar_horario	OK	
Cloud	Telegram	consultar_horario	OK	
Cloud	Telegram	consultar_horario	OK	
Cloud	Telegram	despedida	FAIL	No identificó la intención de terminar la conversación
Cloud	Telegram	despedida	OK	Fix: Aumentar frases de entrenamiento

## 2.1 Resultados por canal

El gráfico presenta la comparación de resultados de las pruebas por canal de comunicación (REST, Dialogflow y Telegram), diferenciando entre ejecuciones exitosas (OK) y fallidas (FAIL). Se observa que Telegram concentra la mayor cantidad de casos exitosos, lo que refleja un comportamiento más estable del chatbot en el entorno de uso real. En contraste, Dialogflow



presenta una mayor proporción de fallos, mientras que REST muestra resultados intermedios, propios de un entorno de pruebas y simulación manual.



### Análisis de resultados

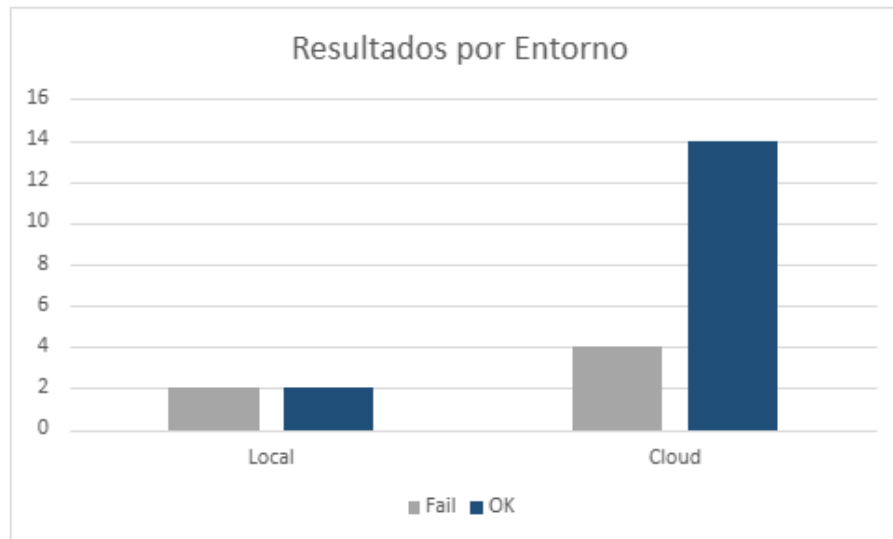
Los resultados evidencian que el desempeño del chatbot varió según el canal de interacción. Telegram presenta mayor estabilidad al integrar de forma más natural la gestión de sesiones y flujos conversacionales, especialmente porque las pruebas en Telegram, en su mayoría, se realizaron luego de aplicar todas las correcciones. Por el contrario, los fallos en Dialogflow se relacionan con ajustes necesarios en el motor de NLU y la configuración de intenciones, mientras que los resultados en REST reflejan las limitaciones propias de la simulación de conversaciones.

En conjunto, el análisis resalta la importancia de validar el sistema en múltiples canales para asegurar un comportamiento consistente antes del despliegue final.

### 2.2 Resultados según entorno

El gráfico presenta la comparación de los resultados de las pruebas realizadas en los entornos Local y Cloud, diferenciando entre ejecuciones exitosas (OK) y fallidas (FAIL). Se observa que el entorno Cloud concentra un mayor número de casos de prueba, así como una mayor cantidad de ejecuciones exitosas, mientras que el entorno Local presenta menos

registros, dado que su uso estuvo orientado principalmente a validaciones puntuales previas al proceso de despliegue.



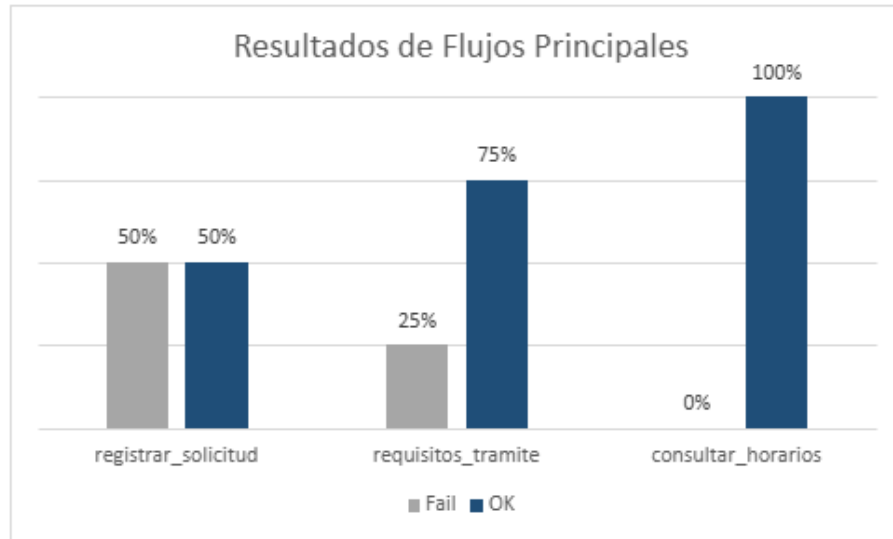
### Análisis de resultados

Los resultados evidencian una mayor estabilidad del sistema en el entorno Cloud, donde se realizaron pruebas más extensas e integrales sobre los flujos del chatbot. La mayor proporción de casos exitosos en este entorno refleja que las correcciones aplicadas durante la fase de pruebas locales permitieron un despliegue más controlado y estable en la nube. Por su parte, el entorno Local fue utilizado de manera estratégica para validar aspectos técnicos previos a la migración (configuración de dependencias, variables de entorno e integraciones), por lo que el menor volumen de pruebas en este entorno no representa una menor cobertura funcional, sino una etapa de preparación para el entorno de ejecución final. En conjunto, el gráfico respalda la decisión de concentrar la validación funcional completa en el entorno Cloud como entorno objetivo de despliegue.

### 2.3 Resultados según flujo

El gráfico presenta el porcentaje de ejecuciones exitosas (OK) y fallidas (FAIL) para los flujos principales del chatbot: *registrar\_solicitud*, *requisitos\_tramite* y *consultar\_horarios*. Se observa que *consultar\_horarios* alcanza un alto nivel de estabilidad, mientras que *requisitos\_tramite* presenta un desempeño mayoritariamente exitoso. El flujo

*registrar\_solicitud*, por su mayor complejidad y longitud conversacional, muestra una distribución más equilibrada entre casos exitosos y fallidos en las etapas iniciales de prueba.



### Análisis de resultados

Las diferencias en los porcentajes de éxito entre los flujos están directamente relacionadas con su complejidad y alcance funcional. El flujo *registrar\_solicitud* implica múltiples turnos conversacionales, validación de entradas del usuario e integración con servicios externos, por lo que requirió un proceso de ajuste más iterativo. Adicionalmente, varias de las fallas iniciales estuvieron asociadas a interrupciones del flujo por interpretaciones incorrectas del NLU de Dialogflow, que activaba intenciones no relacionadas; no obstante, tras los ajustes en parámetros, ejemplos de entrenamiento y configuración del NLU, el flujo alcanzó un comportamiento **estable y consistente**.

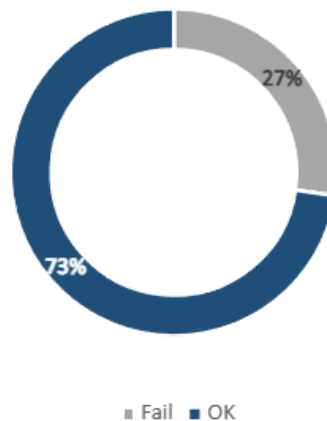
En conjunto, los resultados finales evidencian una mejora progresiva en la estabilidad de todos los flujos, destacando que los flujos más complejos demandan mayores ciclos de prueba y ajuste antes de alcanzar niveles de confiabilidad comparables a los flujos informativos más simples.



## 2.4 Resultados Globales

El gráfico de dona presenta la distribución global de los resultados de las pruebas realizadas durante la Fase 3 del proyecto, diferenciando entre ejecuciones exitosas (OK) y fallidas (FAIL). El **73% de los casos corresponde a ejecuciones exitosas**, mientras que el **27% refleja incidencias detectadas durante las primeras iteraciones de prueba**, previas a la aplicación de los ajustes y correcciones finales.

Resultados Globales de Pruebas



### Análisis de resultados

La distribución de resultados evidencia un proceso de prueba iterativo y progresivo, en el que los fallos identificados durante las etapas iniciales permitieron aplicar correcciones técnicas oportunas sobre la configuración de Dialogflow, la continuidad de los flujos conversacionales y el manejo del entorno de despliegue.

Es importante destacar que, tras la aplicación de estos ajustes, **todos los flujos principales fueron revalidados y alcanzaron un funcionamiento estable**, sin interrupciones no previstas. En este sentido, el porcentaje de fallos representado en el gráfico no corresponde al estado final del sistema, sino al historial de incidencias detectadas y corregidas durante la fase de pruebas. En conjunto, los resultados confirman la madurez funcional alcanzada por el chatbot al cierre de la Fase 3.

## Evidencias

**Evidencia 1.** Documento de Matriz de Pruebas y Documentación:

<https://drive.google.com/file/d/1tiEhspueYEJg60dLfSz9pINZoHdtqEoP/view?usp=sharing>

**Evidencia 2.** Enlace a diagrama Draw.io:

[https://app.diagrams.net/?tags=%7B%7D&lightbox=1&highlight=0000ff&edit=blank&layers=1&nav=1&dark=auto#Hcorberaks%2Fgrupo20\\_Chatbot%2Fmain%2Fflujos%2FPipeline%20Fase%203.drawio](https://app.diagrams.net/?tags=%7B%7D&lightbox=1&highlight=0000ff&edit=blank&layers=1&nav=1&dark=auto#Hcorberaks%2Fgrupo20_Chatbot%2Fmain%2Fflujos%2FPipeline%20Fase%203.drawio)

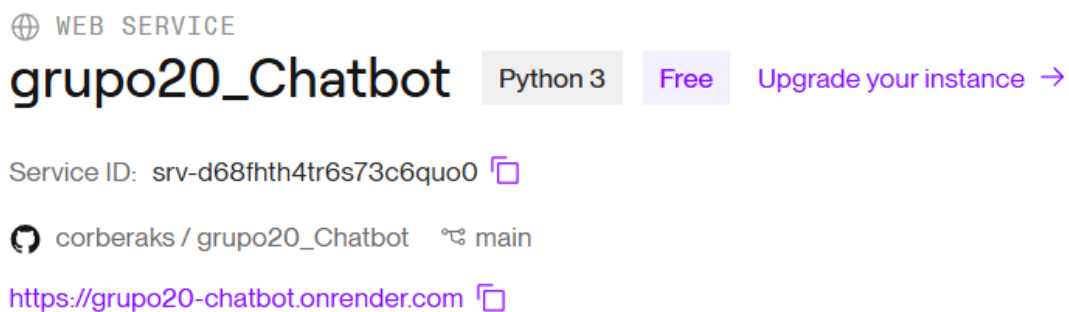
**Evidencia 3.** Conversación completa en Telegram:



#### Evidencia 4. Flujos coherentes con intenciones correctas



#### Evidencia 5. Cuenta en Render conectado a Repositorio



## Conclusiones

1. En conclusión, el despliegue del chatbot en un entorno Cloud permitió alcanzar un mayor nivel de estabilidad operativa al unificar la configuración del sistema, las dependencias y las integraciones externas en un entorno controlado. A diferencia del entorno local, donde las pruebas pueden verse afectadas por variaciones en la configuración del equipo o por dependencias no sincronizadas, la operación en la nube facilitó una validación más consistente de los flujos conversacionales y una ejecución más predecible del sistema en condiciones cercanas al uso real.
2. Finalmente, la disponibilidad del sistema en la nube permitió evaluar el chatbot de manera integral a través de distintos canales y flujos, sin depender de las limitaciones propias del entorno local (configuraciones específicas del equipo, rutas de archivos o credenciales temporales). Esto facilitó un proceso de pruebas más consistente, la aplicación de correcciones de forma iterativa y la verificación de la estabilidad final del sistema. En conjunto, el entorno cloud se consolida como el contexto más adecuado para la operación y evolución futura del chatbot.