



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



TFG del Grado en Ingeniería  
Informática

Creación de chatbots en Amazon  
Web Services y Google Cloud.  
Documentación Técnica



Presentado por Mario Lopez Matamala  
en Universidad de Burgos — 9 de julio de 2024  
Tutores: Jose Manuel Aroca Fernandez



---

# Índice general

---

Índice general	i
Índice de figuras	iii
Índice de tablas	iv
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Metodología Scrum . . . . .	2
A.3. Herramienta . . . . .	3
A.4. Planificación temporal . . . . .	3
A.5. Estudio de viabilidad . . . . .	8
<b>Apéndice B Especificación de Requisitos</b>	<b>25</b>
B.1. Introducción . . . . .	25
B.2. Objetivos generales . . . . .	26
B.3. Catálogo de requisitos . . . . .	27
B.4. Especificación de requisitos . . . . .	29
<b>Apéndice C Especificación de diseño</b>	<b>37</b>
C.1. Introducción . . . . .	37
C.2. Diseño de datos . . . . .	37
C.3. Diseño procedimental . . . . .	40
C.4. Diseño arquitectónico . . . . .	42
<b>Apéndice D Documentación técnica de programación</b>	<b>45</b>
D.1. Introducción . . . . .	45

D.2. Estructura de directorios . . . . .	45
D.3. Manual del programador . . . . .	48
<b>Apéndice E Documentación de usuario</b>	<b>55</b>
E.1. Introducción . . . . .	55
E.2. Requisitos de usuarios . . . . .	55
E.3. Instalación . . . . .	56
E.4. Manual del usuario . . . . .	56
<b>Apéndice F Acrónimos</b>	<b>61</b>
<b>Bibliografía</b>	<b>63</b>

---

# Índice de figuras

---

B.1. Casos de uso. Fuente: elaboración propia . . . . .	30
C.1. Diseño de bases de datos NoSQL para respuestas Firestore. Fuente: elaboración propia . . . . .	39
C.2. Diseño de bases de datos NoSQL para pasos Firestore. Fuente: elaboración propia . . . . .	39
C.3. Diseño de bases de datos NoSQL DynamoDB. Fuente: elaboración propia	40
C.4. Diagrama de secuencia para la extracción de texto AWS. Fuente: elaboración propia . . . . .	40
C.5. Diagrama de secuencia para la extracción de texto GCP. Fuente: elaboración propia . . . . .	41
C.6. Diagrama de secuencia para el bot AWS. Fuente: elaboración propia . .	41
C.7. Diagrama de secuencia para el bot GCP. Fuente: elaboración propia . .	42
D.1. Diagrama de paquetes general. . . . .	46
D.2. Diagrama de paquetes AWS. . . . .	46
D.3. Diagrama de paquetes GCP. . . . .	47
D.4. Diagrama de paquetes documentación. . . . .	48
D.5. Diagrama de paquetes webapp. . . . .	48
E.1. Pagina principal de la webapp . . . . .	57
E.2. Página AWS de la webapp . . . . .	58
E.3. Página GCP de la webapp . . . . .	59

---

## Índice de tablas

---

A.1. Detalles de la Fase Previa . . . . .	4
A.2. Detalles del Sprint 0 . . . . .	4
A.3. Detalles del Sprint 1 . . . . .	5
A.4. Detalles del Sprint 2 . . . . .	5
A.5. Detalles del Sprint 3 . . . . .	6
A.6. Detalles del Sprint 4 . . . . .	6
A.7. Detalles del Sprint 5 . . . . .	7
A.8. Detalles del Sprint 6 . . . . .	7
A.9. Detalles del Sprint 7 . . . . .	7
A.10. Costes por Solicitud/Unidad/Almacenamiento de los servicios AWS utilizados en el proyecto . . . . .	10
A.11. Uso Mensual Estimado de los servicios AWS utilizados en el proyecto .	11
A.12. Costes por Solicitud/Unidad/Almacenamiento de los servicios GCP utilizados en el proyecto . . . . .	13
A.13. Uso Mensual Estimado de los servicios GCP utilizados en el proyecto .	14
A.14. Coste total mensual del proyecto, combinando los servicios de AWS, GCP y la licencia de Windows 11 . . . . .	16
A.15. Conceptos de Seguridad Social . . . . .	19
A.16. Otros conceptos . . . . .	19
A.17. Resumen de costes del personal . . . . .	20
A.18. Resumen de costes del proyecto . . . . .	21
B.1. CU-1 Subida de Documentos . . . . .	31
B.2. CU-2 Extracción de Texto . . . . .	32
B.3. CU-3 Análisis y Traducción de Texto . . . . .	33
B.4. CU-4 Gestión de Tareas y Notificaciones . . . . .	34
B.5. CU-5 Interacción con el Chatbot . . . . .	35

D.1. Servicios y sus ofertas correspondientes AWS . . . . .	49
D.2. Servicios y sus ofertas correspondientes en GCP . . . . .	52





## Apéndice A

---

# Plan de Proyecto Software

---

### A.1. Introducción

La planificación temporal es esencial para el éxito de cualquier proyecto, especialmente en el desarrollo de *software*. Esta se puede definir como una etapa que implica detallar los objetivos y el alcance, identificar los requisitos y recursos necesarios, establecer un cronograma con hitos clave, asignar tareas a los miembros del equipo y prever los posibles riesgos.

En la memoria se ha indicado que se ha seguido una metodología ágil de gestión de proyectos, con clara fundamentación en Scrum[13].

En esta sección se detallará cómo se ha llevado a cabo el cronograma siguiendo esa metodología, mediante el uso de *sprints*. Se describirán los pasos necesarios para planificar y gestionar de manera eficiente el tiempo y los recursos disponibles, asegurando así el cumplimiento de los objetivos del proyecto en el plazo establecido.

Algunos de los aspectos más relevantes que han cubierto esta filosofía han sido:

- Desarrollo incremental a través de iteraciones llamadas *sprints*.
- Utilización de repositorio *Git*[10] para solicitud de mejoras y para realizar un seguimiento de la evolución, con acceso para los tutores desde las fases iniciales.
- Control temporal a través de los *sprints*, valorando al inicio de cada uno cuál sería la duración adecuada basándonos en las tareas que se iban a realizar y el producto que se esperaba al final de esa iteración.

Además de los *sprints* se diseñaron hitos relevantes denominados *Milestones* en *GitHub* que sirven como referencia de la producción realizada.

## A.2. Metodología Scrum

La metodología Scrum es un marco de trabajo ágil que se utiliza para la gestión de proyectos. Se basa en la colaboración entre el equipo de desarrollo y el cliente, así como en la entrega continua de productos funcionales en ciclos cortos de tiempo conocidos como *sprints* [12]. Scrum se centra en la flexibilidad y la adaptabilidad, facilitando los reajustes y los cambios ante situaciones inesperadas.

El uso de esta metodología permitirá tener un enfoque más dinámico y flexible, permitiéndome así el adaptarme a las necesidades a lo largo del proyecto. Además, Scrum proporcionará una mayor transparencia y comunicación con el tutor, lo que nos capacitará para tomar decisiones informadas y asegurar que el proyecto se entregue a tiempo y con éxito.

A continuación, se presentarán algunos conceptos clave que son fundamentales para comprender la metodología Scrum.

### Roles en Scrum

En Scrum, existen tres roles principales: el Scrum Master, el Product Owner y el Equipo de Desarrollo.

1. El Scrum Master es responsable de garantizar que el equipo siga las prácticas y principios de Scrum, eliminando cualquier obstáculo que pueda afectar a la productividad.
2. El Product Owner es el encargado de definir y priorizar los elementos del *product backlog*, asegurándose de que se cumplan las necesidades del cliente.
3. Por último, el Equipo de Desarrollo es responsable de la implementación de las tareas y la entrega del producto.

Dado que en este caso el equipo está conformado solamente por el tutor y alumno, es importante adaptar los roles y las prácticas de Scrum para que se ajusten a esta situación específica. Se pueden asignar los roles de Scrum Master y Product Owner al propio tutor y, el rol de Equipo de Desarrollo, correspondería al alumno.

### Medición de las tareas

Scrum utiliza la técnica de medición de tareas para estimar el esfuerzo relativo requerido para completar una tarea. Los *story points* son una medida abstracta que

no se relaciona directamente con el tiempo, sino con la complejidad y el esfuerzo necesario para completar una tarea en comparación con otras. Esta medida ayuda al equipo a planificar su capacidad de trabajo en cada *sprint* y a realizar estimaciones más precisas en función de su experiencia previa. Sin embargo, en este proyecto no se han utilizado.

## Artefactos en Scrum

Se denomina Artefacto a aquellos elementos físicos que se producen como resultado de la aplicación de Scrum. Hay tres principales: el *product backlog*, el *sprint backlog* y el incremento.

1. El *product backlog* es una lista priorizada de todas las funcionalidades, características o mejoras que se desean implementar en el producto final.
2. El *sprint backlog* es una selección de elementos del *backlog* del producto que se abordarán en un *sprint* específico.
3. Y finalmente, el incremento es el resultado tangible y potencialmente entregable al final de cada *sprint*, que incorpora nuevas funcionalidades o mejoras al producto.

## A.3. Herramienta

Para llevar a cabo este proyecto, se ha utilizado GitHub como herramienta colaborativa. GitHub proporciona un entorno seguro y eficiente para almacenar y gestionar el código fuente, permitiendo trabajar de manera transparente y controlada.

## A.4. Planificación temporal

El presente proyecto comenzó en febrero y se extendió hasta Julio. Durante este período de tiempo, ha sido importante asegurarnos de que todas las tareas e hitos hayan estado claramente definidos.

Para lograr esto, se han realizado reuniones periódicas para discutir el progreso del proyecto y asegurarnos de estar en el camino correcto. Además, se han implementado *sprints* regulares (de entre una y dos semanas de duración) para asegurarnos de que estamos cumpliendo con nuestras metas a tiempo.

Con esta planificación temporal sólida, hemos podido estar seguros de completar el proyecto a tiempo y cumplir con los objetivos establecidos. Sin embargo, es

importante ser flexibles y estar preparados para hacer ajustes según sea necesario a medida que avanzamos en el proyecto, puesto que se realiza al mismo tiempo que transcurre el curso académico.

En las siguientes tablas se recogen los distintos *sprints* junto con su duración y objetivos. La descripción detallada de las tareas y su duración puede verse en el repositorio de GitHub.

### Fase Previa

Sprint	Fecha	Objetivo
N/A	Diciembre de 2023	Elegir el proyecto

Tabla A.1: Detalles de la Fase Previa

#### Tareas:

- Idea de TFG

**Contexto:** Inicio del último curso del Grado de Ingeniería Informática. Se trata de contactar con los posibles tutores que controlen el desarrollo de un TFG. Este TFG estará basado en la creación de dos chatbot utilizando los servicios de AWS y GCP. Esta fase no ha sido establecida como *sprint* en Github pero se considera parte del proyecto.

### Fase Inicial

Sprint	Fecha	Objetivo
0	18/03/2024 - 01/04/2024	Comenzar el desarrollo del proyecto

Tabla A.2: Detalles del Sprint 0

#### Tareas:

- Elegir un modelo de referencias bibliográficas
- Definición de conceptos

**Contexto:** Se trata de la fase inicial del proyecto, donde se establecen las bases y se comienzan a definir los conceptos clave.

**Sprint 1**

Sprint	Fecha	Objetivo
1	08/04/2024 - 22/04/2024	Labor de investigación en AWS

Tabla A.3: Detalles del Sprint 1

**Tareas:**

- Búsqueda de servicios en AWS
- Comprensión del funcionamiento de estos servicios

**Contexto:** Esta fase puedo decir que fue la más complicada de todas. Consistió en investigar acerca de la plataforma AWS, todos los servicios que proporciona y que me podían ser útiles, así como búsqueda de documentación, ensayos, tutoriales, etc. Tras este sprint, no solo salí bastante capacitado para empezar a trabajar en AWS, sino también en GCP dada la similitud en la forma de trabajar en multitud de servicios.

**Sprint 2**

Sprint	Fecha	Objetivo
2	22/04/2024 - 28/04/2024	Labor de investigación en GCP

Tabla A.4: Detalles del Sprint 2

**Tareas:**

- Equivalencia de servicios de AWS en GCP
- Comprensión del funcionamiento de estos servicios

**Contexto:** Este sprint tuvo un objetivo similar al anterior. Tras tener el grupo de servicios de AWS que podían satisfacer mis necesidad, busqué su equivalencia en GCP. Necesitaba encontrar aquellos que cumplieran con las mismas necesidades, además de comprenderlos. La mayoría de ellos obtenían el mismo fin, pero con diferentes formas de usarlos.

Sprint	Fecha	Objetivo
3	29/04/2024 - 04/05/2024	Prototipo de AWS

Tabla A.5: Detalles del Sprint 3

### Sprint 3

**Tareas:**

- Comenzar a trabajar con los servicios
- Realizar un primer prototipo del entorno en AWS

**Contexto:** Este sprint consistió en la creación del primer entorno en AWS que permitiera la extracción de texto desde los PDF para formato TXT utilizando los servicios de la plataforma.

### Sprint 4

Sprint	Fecha	Objetivo
4	13/05/2024 - 19/05/2024	Prototipo de GCP

Tabla A.6: Detalles del Sprint 4

**Tareas:**

- Comenzar a trabajar con los servicios
- Realizar un primer prototipo del entorno en GCP

**Contexto:** El sprint 4 tuvo como objetivo desarrollar el mismo entorno realizado en el sprint 3 pero utilizando los servicios de GCP.

### Sprint 5

**Tareas:**

- Crear una lambda para la interacción con el bot

Sprint	Fecha	Objetivo
5	20/05/2024 - 02/06/2024	Prototipo de función lambda AWS

Tabla A.7: Detalles del Sprint 5

- Creación de intents y slots en Lex

**Contexto:** Este sprint fue uno de los mas importantes ya que empecé a crear el objetivo de este proyecto: el chatbot. Cree el primer prototipo de función lambda que interacturara con el bot para responder preguntas y explicar conceptos de creación del chatbot.

### Sprint 6

Sprint	Fecha	Objetivo
6	10/06/2024 - 24/06/2024	Prototipo de cloud function

Tabla A.8: Detalles del Sprint 6

#### Tareas:

- Crear una cloud function para la interacción con el bot
- Creación de intents y slots en dialogflow

**Contexto:** En este sprint se realizó lo equivalente de AWS del sprint anterior pero en GCP, con la novedad de la implementación de una lógica para la BBDD noSQL que permita recoger respuestas a las preguntas, así como los intents correspondientes en GCP.

### Sprint 7

Sprint	Fecha	Objetivo
7	24/06/2024 - 09/07/2024	Comenzar con la memoria

Tabla A.9: Detalles del Sprint 7

#### Tareas:

- Creación de toda la documentación del proyecto
- Implementación BBDD AWS
- Corrección de errores
- Prototipo webapp

**Contexto:** Este es el último sprint, donde ya casi todo el trabajo en ambas plataformas está hecho y toca empezar a documentar en la memoria todo el proyecto realizado. También se tiene que implementar la BBDD realizada en el sprint anterior pero en AWS, así como una webapp donde implementar los chatbot.

## A.5. Estudio de viabilidad

El estudio de viabilidad es un componente esencial en el proceso de evaluación de cualquier proyecto o iniciativa. Dado que la intención no es obtener beneficios económicos directos de este proyecto, se considerará un enfoque empresarial realista para asegurar la viabilidad y sostenibilidad a largo plazo de nuestro proyecto.

En las siguientes secciones, se analizarán en detalle los aspectos de viabilidad económica (sección A.5) y legal (sección A.5), con el objetivo de establecer un marco sólido que asegure la continuidad y éxito del proyecto.

### Viabilidad económica

La viabilidad económica se refiere a la evaluación de los recursos financieros necesarios para llevar a cabo el proyecto y asegurar su sostenibilidad a largo plazo.

Es importante considerar los costos asociados al desarrollo, mantenimiento y mejora continua de la plataforma, así como analizar posibles fuentes de financiación, como subvenciones, donaciones o colaboraciones con instituciones interesadas en respaldar este tipo de iniciativas científicas abiertas.

Vamos a considerar una empresa que enseñe a las demás empresas a crear chatbots utilizando los servicios de AWS y GCP.

Algunos aspectos relevantes para poder comenzar a hacer estos cálculos son los siguientes:

- La empresa se situaría en España, esto implica tener en cuenta las regulaciones y requisitos legales, además de fiscales del país.



- El proyecto tendría una duración estimada de 4 meses, desde marzo hasta julio. Esto equivaldría aproximadamente a 16 semanas, considerando una duración promedio de 4 semanas por mes.
- El equipo encargado del proyecto estaría formado por un desarrollador (el alumno), así como por el *Product Owner* y el *Scrum Master*, quienes ambos serían el tutor del alumno.

Para todo esto, es fundamental considerar los costes y beneficios asociados al proyecto:

- Los **costes** se refieren a los desembolsos monetarios necesarios para llevar a cabo la iniciativa, incluyendo los gastos de inversión, los costes operativos y los costes de mantenimiento [9].
- Los **beneficios** se refieren a las ganancias económicas que se esperan obtener como resultado de la implementación del proyecto. Estos beneficios pueden incluir ingresos generados por la venta de productos o servicios, ahorros en costes operativos, incremento en la productividad, entre otros [9].

## Costes

A continuación, se desarrollan los costes totales de la empresa:

### 1. Costes de software.

Comenzaremos por contemplar las licencias correspondientes a los sistemas operativos y herramientas utilizadas.

- Windows 11 Pro: El valor aproximado de la licencia de Windows 11 Pro 159 USD. Con una vida útil de 4 años:

$$\text{Coste anual de amortización} = \frac{159 \text{ USD}}{4 \text{ años}} \approx 40 \text{ USD/año}$$

La duración del proyecto ha sido de 4 meses, por lo que los costes en proporción a estos 4 meses suponen un 33,3 %

$$\text{Proporción de costes} = \frac{4 \text{ meses}}{12 \text{ meses (año)}} \approx 0,33 (33.3 \%)$$

Aplicando el 33,3 % del porcentaje anual que ha supuesto el proyecto, tenemos lo siguiente:

$$\text{Costes de licencia Windows} = 40 \text{ USD} \times 0,33 \approx 13,2 \text{ USD}$$

Se adjunta una estimación de los posibles costes en el uso de los servicios para crear el chatbot tanto en AWS como en GCP.

a) Amazon Web Services

- **AWS Lambda:** \$0.20 USD por cada millón de solicitudes.
- **Amazon S3:**
  - Solicitudes de introducir u obtener documentos: \$0.00045 USD por cada 1000 solicitudes.
  - Almacenamiento: \$0.023 USD por GB para los primeros 50 TB/mes.
- **Amazon Textract:** \$1.50 USD por cada 1000 páginas.
- **Amazon SNS:** \$0.60 USD por millón de notificaciones.
- **Amazon Comprehend:** \$0.0001 USD por unidad hasta 10 millones de unidades (1 unidad = 100 caracteres).
- **Amazon Lex:** \$0.75 USD por cada 1000 solicitudes.
- **Amazon DynamoDB:**
  - Escrituras: \$1.25 USD por millón de escrituras.
  - Lecturas: \$0.25 USD por millón de lecturas.
- **Amazon Translate:** \$15.00 USD por millón de caracteres.

Servicio AWS	Coste por Solicitud/Unidad/Almacenamiento
AWS Lambda	\$0.20 USD por millón de solicitudes
Amazon S3	<ul style="list-style-type: none"> <li>• \$0.00045 USD por 1000 solicitudes</li> <li>• \$0.023 USD por GB para los primeros 50 TB/mes</li> </ul>
Amazon Textract	\$1.50 USD por 1000 páginas
Amazon SNS	\$0.60 USD por millón de notificaciones
Amazon Comprehend	\$0.0001 USD por unidad (100 caracteres)
Amazon Lex	\$0.75 USD por 1000 solicitudes
Amazon DynamoDB	<ul style="list-style-type: none"> <li>• \$1.25 USD por millón de escrituras</li> <li>• \$0.25 USD por millón de lecturas</li> </ul>
Amazon Translate	\$15.00 USD por millón de caracteres

Tabla A.10: Costes por Solicitud/Unidad/Almacenamiento de los servicios AWS utilizados en el proyecto

Vamos a asumir las siguientes cifras de uso mensual:

- **AWS Lambda:** 5 millones de solicitudes.

- **Amazon S3:**
  - Solicitudes: 500,000 solicitudes de introducir u obtener datos.
  - Almacenamiento: 100 GB.
- **Amazon Textract:** 5,000 páginas.
- **Amazon SNS:** 2 millones de notificaciones.
- **Amazon Comprehend:** 5,000 unidades.
- **Amazon Lex:** 10,000 solicitudes de texto.
- **Amazon DynamoDB:**
  - Escrituras: 3.55 millones de escrituras.
  - Lecturas: 3.55 millones de lecturas.
- **Amazon Translate:** 1 millón de caracteres.

Servicio AWS	Uso Mensual Estimado
AWS Lambda	5 millones de solicitudes
Amazon S3	<ul style="list-style-type: none"> <li>• 500,000 solicitudes</li> <li>• 100 GB</li> </ul>
Amazon Textract	5,000 páginas
Amazon SNS	2 millones de notificaciones
Amazon Comprehend	5,000 unidades
Amazon Lex	10,000 solicitudes de texto
Amazon DynamoDB	<ul style="list-style-type: none"> <li>• 3.55 millones de escrituras</li> <li>• 3.55 millones de lecturas</li> </ul>
Amazon Translate	1 millón de caracteres

Tabla A.11: Uso Mensual Estimado de los servicios AWS utilizados en el proyecto

- **AWS Lambda:**

$$\text{Coste} = \frac{5 \text{ millones}}{1 \text{ millón}} \times 0,20 \text{ USD} = 1,00 \text{ USD}$$

- **Amazon S3:**

- Solicitudes:

$$\text{Coste} = \frac{500,000}{1000} \times 0,00045 \text{ USD} = 0,225 \text{ USD}$$

- Almacenamiento:

$$\text{Coste} = 100 \text{ GB} \times 0,023 \text{ USD/GB} = 2,30 \text{ USD}$$

- Total:

$$0,225 \text{ USD} + 2,30 \text{ USD} = 2,525 \text{ USD}$$

■ **Amazon Textract:**

$$\text{Coste} = \frac{5,000}{1000} \times 1,50 \text{ USD} = 7,50 \text{ USD}$$

■ **Amazon SNS:**

$$\text{Coste} = \frac{2 \text{ millones}}{1 \text{ millón}} \times 0,60 \text{ USD} = 1,20 \text{ USD}$$

■ **Amazon Comprehend:**

$$\text{Coste} = 5,000 \times 0,0001 \text{ USD} = 0,50 \text{ USD}$$

■ **Amazon Lex:**

$$\text{Coste} = \frac{10,000}{1000} \times 0,75 \text{ USD} = 7,50 \text{ USD}$$

■ **Amazon DynamoDB:**

- Escrituras:

$$\text{Coste} = \frac{3,55 \text{ millones}}{1 \text{ millón}} \times 1,25 \text{ USD} = 4,44 \text{ USD}$$

- Lecturas:

$$\text{Coste} = \frac{3,55 \text{ millones}}{1 \text{ millón}} \times 0,25 \text{ USD} = 0,89 \text{ USD}$$

- Total:

$$4,44 \text{ USD} + 0,89 \text{ USD} = 5,33 \text{ USD}$$

■ **Amazon Translate:**

$$\text{Coste} = \frac{1 \text{ millón}}{1 \text{ millón}} \times 15,00 \text{ USD} = 15,00 \text{ USD}$$

b) Google Cloud Platform

- **Cloud Functions:** \$0.40 USD por millón de solicitudes (primeros 2 millones gratis).
- **Document AI:** \$1.50 USD por 1000 páginas.
- **Cloud Storage:**
  - Almacenamiento: \$0.023 USD por GB al mes.
  - Operaciones: \$0.005 USD por cada 1000 operaciones.

- **Cloud Tasks:** Primer millón gratis al mes.
- **Natural Language API:** \$0.0020 USD por unidad (1000 caracteres).
- **Translation API:** \$20.00 USD por millón de caracteres.
- **Dialogflow:** \$0.002 USD por solicitud de API.
- **Firestore:**
  - Almacenamiento: \$0.18 USD por GB al mes.
  - Operaciones de lectura: \$0.02 USD por 100,000 operaciones.
  - Operaciones de escritura: \$0.18 USD por 100,000 operaciones.
  - Operaciones de eliminación: \$0.02 USD por 100,000 operaciones.

Servicio GCP	Coste por Solicitud/Unidad/Almacenamiento
Cloud Functions	\$0.40 USD por millón de solicitudes (primeros 2 millones gratis)
Document AI	\$1.50 USD por 1000 páginas
Cloud Storage	<ul style="list-style-type: none"> <li>• \$0.023 USD por GB al mes</li> <li>• \$0.005 USD por cada 1000 operaciones</li> </ul>
Cloud Tasks	Primer millón gratis al mes
Natural Language API	\$0.0020 USD por unidad (1000 caracteres)
Translation API	\$20.00 USD por millón de caracteres
Dialogflow	\$0.002 USD por solicitud de API
Firestore	<ul style="list-style-type: none"> <li>• \$0.18 USD por GB al mes</li> <li>• \$0.02 USD por 100,000 operaciones de lectura</li> <li>• \$0.18 USD por 100,000 operaciones de escritura</li> <li>• \$0.02 USD por 100,000 operaciones de eliminación</li> </ul>

Tabla A.12: Costes por Solicitud/Unidad/Almacenamiento de los servicios GCP utilizados en el proyecto

Vamos a asumir las siguientes cifras de uso mensual:

- **Cloud Functions:** 3 millones de solicitudes.
- **Document AI:** 10,000 páginas.
- **Cloud Storage:**
  - Operaciones: 200,000 operaciones.
  - Almacenamiento: 200 GB.
- **Cloud Tasks:** 1.5 millones de tareas.
- **Natural Language API:** 100,000 unidades (100 millones de caracteres).

- **Translation API:** 1 millón de caracteres.
- **Dialogflow:** 20,000 solicitudes de API.
- **Firestore:**
  - Operaciones de lectura: 500,000 operaciones.
  - Operaciones de escritura: 100,000 operaciones.
  - Operaciones de eliminación: 100,000 operaciones.

Servicio GCP	Uso Mensual Estimado
Cloud Functions	3 millones de solicitudes
Document AI	10,000 páginas
Cloud Storage	<ul style="list-style-type: none"> <li>• 200,000 operaciones</li> <li>• 200 GB</li> </ul>
Cloud Tasks	1.5 millones de tareas
Natural Language API	100,000 unidades (100 millones de caracteres)
Translation API	1 millón de caracteres
Dialogflow	20,000 solicitudes de API
Firestore	<ul style="list-style-type: none"> <li>• 500,000 operaciones de lectura</li> <li>• 100,000 operaciones de escritura</li> <li>• 100,000 operaciones de eliminación</li> </ul>

Tabla A.13: Uso Mensual Estimado de los servicios GCP utilizados en el proyecto

A continuación, se detalla el cálculo de los costes estimados para los servicios de Google Cloud Platform (GCP) utilizados en el proyecto.

■ **Cloud Functions:**

$$\text{Coste} = \frac{3 \text{ millones}}{1 \text{ millón}} \times 0,40 \text{ USD} = 1,20 \text{ USD}$$

■ **Document AI:**

$$\text{Coste} = \frac{10,000}{1000} \times 1,50 \text{ USD} = 15,00 \text{ USD}$$

■ **Cloud Storage:**

- Operaciones:

$$\text{Coste} = \frac{200,000}{1000} \times 0,005 \text{ USD} = 1,00 \text{ USD}$$

- Almacenamiento:

$$\text{Coste} = 200 \text{ GB} \times 0,023 \text{ USD/GB} = 4,60 \text{ USD}$$

- Total:

$$1,00 \text{ USD} + 4,60 \text{ USD} = 5,60 \text{ USD}$$

■ **Cloud Tasks:**

$$\text{Coste} = \frac{1,5 \text{ millones}}{1 \text{ millón}} \times 0,00 \text{ USD} = 0,00 \text{ USD}$$

■ **Natural Language API:**

$$\text{Coste} = \frac{100,000}{1000} \times 0,002 \text{ USD} = 0,20 \text{ USD}$$

■ **Translation API:**

$$\text{Coste} = \frac{1 \text{ millón}}{1 \text{ millón}} \times 20,00 \text{ USD} = 20,00 \text{ USD}$$

■ **Dialogflow:**

$$\text{Coste} = \frac{20,000}{1,000} \times 0,002 \text{ USD} = 0,04 \text{ USD}$$

■ **Firestore:**

- Operaciones de lectura:

$$\text{Coste} = \frac{500,000}{100,000} \times 0,02 \text{ USD} = 0,10 \text{ USD}$$

- Operaciones de escritura:

$$\text{Coste} = \frac{100,000}{100,000} \times 0,18 \text{ USD} = 0,18 \text{ USD}$$

- Operaciones de eliminación:

$$\text{Coste} = \frac{100,000}{100,000} \times 0,02 \text{ USD} = 0,02 \text{ USD}$$

- Almacenamiento:

$$\text{Coste} = 1 \text{ GB} \times 0,18 \text{ USD/GB} = 0,18 \text{ USD}$$

- Total:

$$0,10 \text{ USD} + 0,18 \text{ USD} + 0,02 \text{ USD} + 0,18 \text{ USD} = 0,48 \text{ USD}$$

Concepto	Descripción	Coste Mensual (USD)
<b>Licencia Windows 11</b>	Coste de la licencia de Windows 11 Pro	13.20
<b>Servicios AWS</b>		
AWS Lambda	5 millones de solicitudes	1.00
Amazon S3	Almacenamiento y solicitudes	2.53
Amazon Textract	5,000 páginas procesadas	7.50
Amazon SNS	2 millones de notificaciones	1.20
Amazon Comprehend	5,000 unidades	0.50
Amazon Lex	10,000 solicitudes de texto	7.50
Amazon DynamoDB	Escrituras y lecturas	5.33
Amazon Translate	1 millón de caracteres	15.00
<b>Servicios GCP</b>		
Cloud Functions	3 millones de solicitudes	1.20
Document AI	10,000 páginas	15.00
Cloud Storage	Almacenamiento y operaciones	5.60
Cloud Tasks	1.5 millones de tareas	0.00
Natural Language API	100,000 unidades	0.20
Translation API	1 millón de caracteres	20.00
Dialogflow	20,000 solicitudes de API	0.04
Firestore	Operaciones y almacenamiento	0.48
<b>Total (mensual)</b>		<b>96.28</b>

Tabla A.14: Coste total mensual del proyecto, combinando los servicios de AWS, GCP y la licencia de Windows 11

Para obtener los gastos totales del proyecto, simplemente se multiplica el costo mensual por el número de meses:

$$\text{Gastos totales} = 96,28 \text{ €} \times 4 \text{ meses} \approx 385,12 \text{ €}$$



**Importante**

Se considera un escenario en el que la empresa se contratan ambas plataformas. Sin embargo, en realidad esto no es así, se debería elegir entre una u otra dividiendo el coste de las licencias a la mitad.

**2. Costes de *hardware***

Para calcular los costes de *hardware*, se considerará el portátil del alumno con el que ha realizado el proyecto. Este se trata de ASUS ROG Strix G15 con un procesador AMD Ryzen 9 4.9GHz y una RAM de 32 GB. Este portátil se encuentra en el mercado a un precio de 1.500 €.

El coste de amortización se basará en su vida útil estimada. Supongamos unos 4 años, para calcular el coste anual de amortización, dividiremos el coste del portátil entre el número de años de vida útil:

$$\text{Coste anual de amortización} = \frac{1,500 \text{ €}}{4 \text{ años}} \approx 375 \text{ €/año}$$

Por tanto, como el proyecto ha durado 4 meses, los costes de *hardware* suponen 302,95 por 33,33 % (proporción de los 4 meses sobre un año).

$$\text{Costes de } \textit{hardware} = 375 \text{ €} \times 33,33 \% \approx 124,99 \text{ €}$$

**3. Costes de los empleados.**

Se procederá al cálculo del salario bruto de cada integrante del equipo de acuerdo con la normativa laboral vigente en España para un proyecto de desarrollo de *software*. El salario bruto se determina considerando factores tales como la categoría profesional y la experiencia de cada empleado.

- a) Desarrollador: El alumno es considerado un programador *junior*. En España, un programador *junior* (con menos de 3 años de experiencia laboral) puede obtener una remuneración de alrededor de 19 700 € brutos por año [8]. Además, según el artículo 34 del Estatuto de los Trabajadores, la duración fijada en convenio colectivo de la jornada laboral es un máximo de 40 horas de trabajo efectivo [3]. Ahora, para calcular el salario bruto mensual, se puede utilizar la siguiente fórmula<sup>1</sup>:

---

<sup>1</sup>Entendemos habitualmente 14 pagas (12 meses y 2 pagas extra).

$$\text{Salario Bruto Mensual} = \frac{\text{Salario Bruto Anual}}{14 \text{ pagas}}$$

Sustituyendo el valor del salario bruto anual, obtenemos:

$$\text{Salario Bruto Mensual} = \frac{19\,700 \text{ €}}{14 \text{ pagas}} \approx 1\,407,14 \text{ €}$$

- b) Product Owner: el salario bruto promedio anual para un Product Owner en España es de 41.866 € [4]. Utilizando la misma fórmula para calcular el salario bruto mensual, obtenemos:

$$\text{Salario Bruto Mensual} = \frac{41\,866 \text{ €}}{14 \text{ pagas}} \approx 2\,990,43 \text{ €}$$

- c) Scrum Master: el salario bruto promedio anual para un Scrum Master en España es de 40.626 € [5]. Utilizando la misma fórmula para calcular el salario bruto mensual, obtenemos:

$$\text{Salario Bruto Mensual} = \frac{40\,626 \text{ €}}{14 \text{ pagas}} \approx 2\,901,85 \text{ €}$$

Una vez obtenido el salario bruto, se deben aplicar los impuestos correspondientes de acuerdo con la legislación fiscal vigente. Los impuestos a tener en cuenta incluyen las contribuciones a la Seguridad Social [6], el impuesto sobre la renta<sup>2</sup> (IRPF) [1] y otros impuestos relacionados con la contratación laboral. La información detallada sobre los impuestos y las obligaciones legales se encuentra en los documentos oficiales y normativas correspondientes proporcionados por las autoridades fiscales y laborales citados anteriormente. De forma resumida, Para el año 2024, se han establecido los siguientes porcentajes<sup>3</sup>, los cuales se detallan en la Tabla A.15 y Tabla A.16.

Con base en los cálculos previos de los salarios brutos y los impuestos aplicados, se puede determinar el coste total de la empresa en cuanto a los empleados.

- a) Desarrollador: Dado que el salario bruto mensual del desarrollador es de aproximadamente 1 641,67 €, se aplicarán los porcentajes correspondientes de los conceptos de la Seguridad Social según lo establecido en

---

<sup>2</sup>En cuanto al IRPF, el porcentaje aplicado puede variar según el nivel de ingresos y la situación fiscal personal. Según la Agencia Tributaria de España, se pueden encontrar las tablas y los porcentajes correspondientes en su documentación oficial [2].

<sup>3</sup>El gobierno pone a disposición la plataforma ipyme [7], que proporciona información valiosa para facilitar el cálculo de los tramites empresariales.

Concepto	Descripción	Porc. (%)
Cuota Contingencias Comunes	Gastos comunes en el ámbito laboral	23,60
Cuota Formación Profesional	Inversión en formación y capacitación	0,60
Desempleo	Prestaciones por desempleo	5,50
Accidentes de Trabajo	Prevención y compensación por accidentes laborales	1,50
FOGASA	Fondo de Garantía Salarial	0,20

Tabla A.15: Conceptos de Seguridad Social

Concepto	Descripción	Porcentaje
IRPF	Impuesto sobre la Renta de las Personas Físicas	Variable
IVA	Impuesto sobre el Valor Añadido (tipo general)	21,00 %

Tabla A.16: Otros conceptos

la Tabla A.15. El total de los conceptos de Seguridad Social se calcula sumando los porcentajes aplicados al salario bruto mensual:

$$\text{Total SS} = 0,236 + 0,006 + 0,055 + 0,015 + 0,002 \approx 0,314$$

El IRPF es un impuesto variable y su porcentaje de retención en nómina depende del nivel de ingresos y la situación fiscal personal. Por lo tanto, no se tendrá en cuenta para el resultado de este cálculo.

Por último, el gasto de la empresa se calcula dividiendo el salario bruto mensual entre 1 menos el total de la Seguridad Social. Sustituyendo los valores correspondientes, obtenemos:

$$\text{Gasto mensual} = \frac{1\,407,14\,€}{1 - 0,314} \approx 2\,051,22\,€$$

- b) Product Owner: Como el salario bruto anual para el Product Owner es de 3 488,83 €, aplicando los porcentajes de la Tabla A.15, el gasto total de la empresa para el Product Owner sería:

$$\text{Salario mensual} = \frac{2\,990,43\text{€}}{1 - 0,314} \approx 4\,359,22\text{€}$$

De las 154 horas de la jornada mensual (22 días laborables por 7 horas al día), hemos calculado una dedicación del 7 % de su tiempo. Por tanto, se va aplicar este 7 % al gasto mensual del Product Owner y del Scrum Master. Se ha estimado una dedicación de 2 horas de reuniones semanales más 3 horas repartidas a lo largo del mes.

$$\text{Gasto mensual} = 4\,359,22\text{€} \times 0,07 \approx 305,13\text{€}$$

- c) Scrum Master: De manera similar, para el Scrum Master, partimos de un salario bruto mensual aproximado de 3 385,50 €. Aplicando los porcentajes de la Tabla A.15, el gasto total de la empresa para el Scrum Master sería:

$$\text{Salario mensual} = \frac{2\,901,85\text{€}}{1 - 0,314} \approx 4\,230,10\text{€}$$

Igualmente, al Scrum Master se le aplica un coeficiente reductor del 7 % de su salario, dado que ha invertido el mismo tiempo que el Product Owner.

$$\text{Gasto mensual} = 4\,230,10\text{€} \times 0,07 \approx 296,11\text{€}$$

A modo de resumen, se agrupa el resultado de todos los cálculos en la Tabla A.17.

Empleado	Gasto Mensual (€)
Desarrollador	2 051,22
Product Owner	305,13
Scrum Master	296,11
<b>Total Proyecto</b>	<b>2 652,46</b>

Tabla A.17: Resumen de costes del personal

Para visualizar los costes de forma clara, se ha elaborado una última tabla resumen con la suma de los costes finales de software, hardware y personal: (ver Tabla A.18).

**Importante**

Se ha hecho una conversión de dólares a euros, considerando que un dólar son 0,92 euros, para las licencias software. El precio total de las licencias era de 385 dólares, que son 355 euros.

Concepto	Descripción	Coste Total (€)
Personal	Costes totales de empleados	10,609.84
Hardware	Costes por el portátil	124.99
Software	Costes totales por licencias y servicios	355
<b>Coste Total del Proyecto</b>		<b>11,089,83</b>

Tabla A.18: Resumen de costes del proyecto

**Beneficios**

A pesar de que el fin de la aplicación no sea lucrativos- estará disponible de forma gratuita para la comunidad científica -es importante considerar una forma hipotética de generar ingresos para sostener el proyecto. Valoraremos diferentes opciones.

**1. Donaciones y Financiación Pública:**

- **Crowdfunding:** Se puede lanzar una campaña de financiación colectiva a través de plataformas como Kickstarter, Indiegogo, o GoFundMe. Este método permite a los usuarios interesados en la aplicación contribuir económicamente para su sostenimiento y futuras mejoras.
- **Subvenciones y Becas:** Existen diversas organizaciones y entidades gubernamentales que ofrecen subvenciones y becas para proyectos de investigación y desarrollo tecnológico. Aplicar a estas ayudas puede proporcionar los fondos necesarios para mantener el proyecto.

- **Donaciones Directas:** Habilitar una opción de donaciones directas en la página web del proyecto puede permitir a los usuarios hacer contribuciones voluntarias mediante plataformas como PayPal, Patreon, o BuyMeACoffee puede facilitar este proceso.

## 2. Servicios de Consultoría y Personalización:

- **Consultoría Técnica:** Ofrecer servicios de consultoría técnica a organizaciones que necesiten asesoramiento especializado en la implementación y uso de chatbots.

## 3. Publicidad y Patrocinios:

- **Publicidad en la Plataforma:** Implementar espacios publicitarios dentro de la aplicación y en su página web. Esto puede generar ingresos a través de la venta de estos espacios a empresas interesadas en llegar a la comunidad científica.
- **Patrocinios:** Establecer acuerdos de patrocinio con empresas y organizaciones relacionadas con el campo científico y tecnológico. Los patrocinadores pueden aportar fondos a cambio de la visibilidad de su marca en la plataforma.

## 4. Modelos de Suscripción:

- **Freemium:** Ofrecer una versión básica gratuita de la aplicación con funcionalidades limitadas y una versión premium con características avanzadas mediante una suscripción mensual o anual.
- **Suscripción a Contenidos Exclusivos:** Proporcionar acceso a contenidos exclusivos, como informes detallados, análisis avanzados, y herramientas especializadas, a través de una suscripción.

## 5. Colaboraciones y Alianzas Estratégicas:

- **Alianzas con Universidades y Centros de Investigación:** Establecer colaboraciones con universidades y centros de investigación para el uso y promoción de la aplicación en sus proyectos y programas educativos.
- **Proyectos Conjuntos:** Colaborar con otras organizaciones y empresas en proyectos conjuntos que utilicen la aplicación, compartiendo tanto los costos como los beneficios generados.

## Viabilidad legal

La viabilidad legal implica evaluar las cuestiones jurídicas y normativas que puedan afectar la operación y distribución de la aplicación *open source*. Es importante asegurarse de cumplir con las regulaciones y requisitos legales relacionados con la privacidad, protección de datos y derechos de propiedad intelectual.

### Descargo de responsabilidades

El *software* que se proporciona en este trabajo es solo para fines informativos y debe tenerse cuidado con los servicios que se utilizan dado que pueden generar gastos excesivos.

### Licencias *software*

La única licencia que se ha utilizado para el desarrollo del proyecto ha sido python, la cual es OSI-open source.





## *Apéndice B*

---

# Especificación de Requisitos

---

## B.1. Introducción

Para llevar a cabo el proyecto adecuadamente, se realizará un análisis exhaustivo de los requisitos funcionales y no funcionales de la aplicación. En la sección de Objetivos generales (Sección B.2) se establecerán los resultados esperados y los criterios de éxito del proyecto. Seguidamente, se presentará el catálogo de requisitos (sección B.3), detallando las funcionalidades y características que la aplicación debe cumplir. Finalmente, se procederá a la especificación de requisitos (sección B.4), describiendo cada uno de los identificados en el catálogo y estableciendo su prioridad, complejidad y dependencias.

Se han seguido las recomendaciones del estándar IEEE 830-1998[11] como guía de buenas prácticas. Se ha tratado de crear una especificación de requisitos que cumpla con las siguientes condiciones:

- Modificabilidad y jerarquización
  - Ser fácil de modificar.
  - Establecer una jerarquía de prioridades basada en la relevancia para el negocio o propósito.
- Verificabilidad y rastreabilidad
  - Deberá existir un método finito y sin costo para verificar los requisitos.
  - Emplear términos y definiciones claras.
- Integridad y coherencia
  - Esencial incluir todos los requisitos y referencias pertinentes.

- Mantener la coherencia con los propios requisitos y otros documentos de especificación.
- Correctitud y accesibilidad
  - Asegurar que el software cumpla con los requisitos especificados.
  - Facilitar la accesibilidad y la comprensión tanto para los usuarios como para los desarrolladores.
- Claridad y precisión
  - Redactar de manera clara para evitar malentendidos.
  - Utilizar términos y definiciones precisas.

El presente informe tiene como objetivo establecer las bases y directrices esenciales para el desarrollo exitoso de una aplicación web, ofreciendo a los usuarios información relevante sobre el impacto de las revistas científicas, con el fin de facilitar decisiones informadas en el ámbito de la publicación académica.

## B.2. Objetivos generales

Este trabajo persigue los siguientes objetivos generales:

- Desarrollar dos chatbots funcionales utilizando los servicios en la nube de AWS y GCP.
- Emplear técnicas de procesamiento de lenguaje natural.
- Utilizar servicios para extraer y analizar texto.
- Implementar y evaluar el uso del paradigma serverless utilizando AWS Lambda y Google Cloud Functions.

El objetivo final de este proyecto es desarrollar y comparar dos chatbots utilizando los servicios de AWS y GCP para identificar la plataforma más adecuada en términos de funcionalidad, rendimiento, escalabilidad y experiencia del usuario, proporcionando una guía detallada para futuros desarrollos de chatbots en entornos empresariales.

## B.3. Catálogo de requisitos

### Requisitos funcionales

Los requisitos funcionales describen las funciones y capacidades específicas que debe cumplir el sistema:

#### ■ RF1: Extracción de Texto

- **RF1.1:** El sistema debe permitir la subida de documentos a procesar en los servicios de almacenamiento (Amazon S3 en AWS y Cloud Storage en GCP).
- **RF1.2:** El sistema debe ser capaz de extraer texto de los documentos utilizando Amazon Textract (AWS) y Document AI (GCP).
- **RF1.3:** El sistema debe ser capaz de aceptar texto subido en diferentes disposición, como en columnas, y ser ordenado correctamente en AWS.

#### ■ RF2: Análisis y Traducción de Texto

- **RF2.1:** El sistema debe analizar el texto extraído para detectar el idioma utilizando Amazon Comprehend (AWS) y Cloud Language (GCP).
- **RF2.2:** El sistema debe traducir el texto analizado a un idioma especificado utilizando Amazon Translate (AWS) y Cloud Translate (GCP).

#### ■ RF3: Gestión de Tareas y Notificaciones

- **RF3.1:** El sistema debe gestionar y coordinar las tareas de extracción, análisis y traducción utilizando AWS SNS y Cloud Tasks.

#### ■ RF4: Almacenamiento de Resultados

- **RF4.1:** El sistema debe almacenar los resultados del análisis y la traducción en Amazon S3 (AWS) y Google Cloud Storage (GCP).
- **RF4.2:** Los resultados deben ser accesibles por el chatbot en Lex (AWS) y DialogFlow (GCP).

#### ■ RF5: Interacción con Chatbot

- **RF5.1:** El sistema debe permitir la interacción con los usuarios a través de chatbots desarrollados con Amazon Lex (AWS) y Dialogflow (GCP).
- **RF5.2:** El chatbot debe poder responder preguntas relacionadas con el contenido extraído y analizado.

- **RF5.3:** El sistema debe definir intents específicos en Amazon Lex y Dialogflow para manejar diferentes tipos de consultas, como:
  - **RF5.3.1:** Intents para preguntas sobre los pasos de creación del chatbot.
  - **RF5.3.2:** Intents para preguntas generales.

## Requisitos no funcionales

Los requisitos no funcionales describen los atributos de calidad y restricciones del sistema:

### ■ RNF1: Mantenibilidad

- **RNF1.1:** El código fuente de la aplicación debe estar bien estructurado, modularizado y documentado.
- **RNF1.2:** El código debe seguir las mejores prácticas de desarrollo de *software* y ser fácilmente mantenible y escalable.
- **RNF1.3:** El sistema debe permitir actualizaciones y mantenimiento sin causar interrupciones significativas en el servicio.

### ■ RNF2: Seguridad y Escalabilidad

- **RNF2.1:** El sistema debe garantizar la seguridad y privacidad de los datos personales que introduzca el usuario, implementando medidas de protección adecuadas.
- **RNF2.2:** La aplicación debe protegerse contra ataques de fuerza bruta sobreutilizando los servicios.
- **RNF2.3:** El sistema debe ser escalable para manejar un número creciente de documentos y usuarios sin degradar el rendimiento.
- **RNF2.4:** Las comunicaciones entre los componentes del sistema deben estar restringidas para evitar el acceso no autorizado a los servicios.

### ■ RNF3: Disponibilidad y Rendimiento

- **RNF3.1:** El sistema debe estar disponible al menos el 99.9 % del tiempo para asegurar la fiabilidad del servicio.
- **RNF3.2:** El sistema debe procesar los documentos y responder a las consultas del chatbot en un tiempo razonable, minimizando la latencia.
- **RNF3.3:** La aplicación debe ser capaz de manejar de manera eficiente la carga de usuarios concurrentes utilizando el chatbot.

- **RNF3.4:** Las consultas a la base de datos deben optimizarse para proporcionar respuestas rápidas a los usuarios.
- **RNF4: Usabilidad**
  - **RNF4.1:** La interfaz de usuario debe ser intuitiva y fácil de usar para los usuarios.
  - **RNF4.2:** Los mensajes de error deben ser claros y descriptivos, brindando orientación sobre cómo solucionar los problemas.
- **RNF5: Compatibilidad**
  - **RNF5.1:** El sistema debe ser compatible con diferentes tipos de documentos y formatos para asegurar su versatilidad.
  - **RNF5.2:** La aplicación debe ser compatible con múltiples navegadores y dispositivos para garantizar una experiencia de usuario coherente.

## B.4. Especificación de requisitos

### Actores

Antes de comenzar con los casos de uso, se identifican los diferentes actores que pueden interactuar con la aplicación. Estos son:

- **Usuario:** Referido a una persona que desde la página web donde está desplegado el chatbot, hace uso de este.
- **Administrador:** Usuario con privilegios de acceso a la cuenta de AWS y/o GCP donde puede acceder a los servicios y trabajar con ellos.

### Casos de uso

Se han identificado los siguientes casos de uso:

- **CU-1 Subida de Documentos**
- **CU-2 Extracción de Texto**
- **CU-3 Análisis y Traducción de Texto**
- **CU-4 Gestión de Tareas y Notificaciones**
- **CU-5 Interacción con el Chatbot**

Para poder visualizar los casos de uso se ha incluido un diagrama de casos de uso (ver Figura B.1).

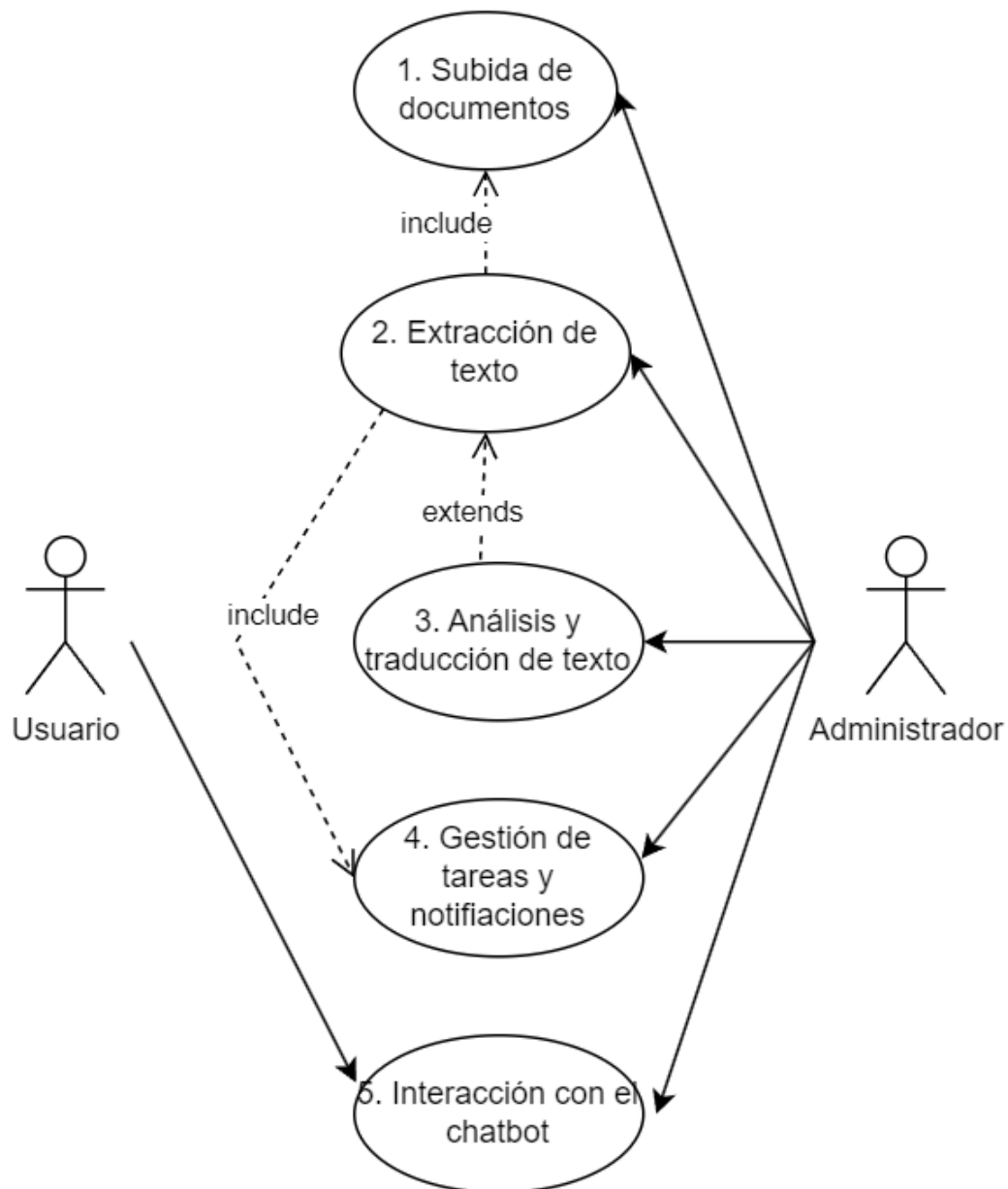


Figura B.1: Casos de uso. Fuente: elaboración propia

CU-1	Subida de Documentos
<b>Versión</b>	1.0
<b>Autor</b>	Mario Lopez Matamala
<b>Requisitos asociados</b>	RF1, RF1.1
<b>Descripción</b>	Este caso de uso describe el proceso de subir documentos al sistema para su posterior procesamiento.
<b>Precondición</b>	El administrador debe tener acceso al sistema y el documento debe estar en un formato PDF.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El administrador accede al servicio en la nube de AWS y GCP.</li> <li>2. El administrador selecciona el documento a subir.</li> <li>3. El administrador confirma la subida del documento.</li> <li>4. El sistema sube el documento a Amazon S3 (AWS) o Cloud Storage (GCP).</li> </ol>
<b>Postcondición</b>	El documento está almacenado en el servicio de almacenamiento correspondiente.
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ Si hay problemas de conexión, se muestra un mensaje de error y se pide al usuario que intente nuevamente.</li> </ul>
<b>Importancia</b>	Alta

Tabla B.1: CU-1 Subida de Documentos

CU-2	Extracción de Texto
<b>Versión</b>	1.0
<b>Autor</b>	Mario Lopez Matamala
<b>Requisitos asociados</b>	RF1, RF1.2, RF1.3
<b>Descripción</b>	Este caso de uso describe el proceso de extracción de texto de documentos subidos.
<b>Precondición</b>	El documento debe estar subido al sistema.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El sistema detecta un nuevo documento subido en el bucket</li> <li>2. El bucket desencadena la función lambda.</li> <li>3. El sistema utiliza Amazon Textract (AWS) o Document AI (GCP) para extraer el texto del documento mediante la función lambda.</li> <li>4. El sistema trata correctamente el texto extraído.</li> </ol>
<b>Postcondición</b>	El texto ha sido extraído y se encuentra listo para ser tratado.
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ Si el texto no puede ser extraído, se notifica al administrador.</li> </ul>
<b>Importancia</b>	Alta

Tabla B.2: CU-2 Extracción de Texto



CU-3	Análisis y Traducción de Texto
<b>Versión</b>	1.0
<b>Autor</b>	Mario Lopez Matamala
<b>Requisitos asociados</b>	RF2, RF2.1, RF2.2
<b>Descripción</b>	Este caso de uso describe el proceso de análisis y traducción del texto extraído de documentos.
<b>Precondición</b>	El texto debe haber sido extraído del documento.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El sistema analiza el texto extraído para detectar el idioma utilizando Amazon Comprehend (AWS) o Cloud Language (GCP).</li> <li>2. El sistema traduce el texto al idioma especificado utilizando Amazon Translate (AWS) o Cloud Translate (GCP).</li> </ol>
<b>Postcondición</b>	El texto ha sido analizado y traducido correctamente.
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ Si el idioma no puede ser detectado, se notifica al administrador del fallo.</li> <li>■ Si la traducción falla, se notifica al administrador.</li> </ul>
<b>Importancia</b>	Alta

Tabla B.3: CU-3 Análisis y Traducción de Texto

CU-4	Gestión de Tareas y Notificaciones
<b>Versión</b>	1.0
<b>Autor</b>	Mario Lopez Matamala
<b>Requisitos asociados</b>	RF3, RF3.1
<b>Descripción</b>	Este caso de uso describe cómo el sistema gestiona y coordina las tareas de extracción, análisis y traducción de texto, y envía notificaciones sobre el estado de estas tareas.
<b>Precondición</b>	Las tareas de extracción, análisis y traducción deben estar en curso.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El sistema utiliza AWS SNS o Cloud Tasks para gestionar y coordinar las tareas.</li> </ol>
<b>Postcondición</b>	La notificación ha sido enviada.
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ Si alguna tarea falla, se envía una notificación de error al administrador.</li> </ul>
<b>Importancia</b>	Alta

Tabla B.4: CU-4 Gestión de Tareas y Notificaciones

CU-5	Interacción con el Chatbot
<b>Versión</b>	1.0
<b>Autor</b>	Mario Lopez Matamala
<b>Requisitos asociados</b>	RF5, RF5.1, RF5.2, RF5.3
<b>Descripción</b>	Este caso de uso describe cómo los usuarios interactúan con el chatbot para obtener información sobre el contenido extraído y analizado.
<b>Precondición</b>	El usuario debe haber establecido una sesión con el chatbot.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario hace preguntas sobre el tema o pide al chatbot los pasos de creación.</li> <li>2. El chatbot responde utilizando la información almacenada en Amazon S3 (AWS) o Google Cloud Storage (GCP) y las bases de datos.</li> <li>3. El chatbot maneja diferentes tipos de consultas utilizando intents definidos en Amazon Lex (AWS) o Dialogflow (GCP).</li> </ol>
<b>Postcondición</b>	El usuario ha recibido la información solicitada a través del chatbot.
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>■ Si el chatbot no puede responder la pregunta, se notifica al usuario que no se ha encontrado una respuesta.</li> </ul>
<b>Importancia</b>	Alta

Tabla B.5: CU-5 Interacción con el Chatbot



## *Apéndice C*

---

# Especificación de diseño

---

## C.1. Introducción

El objetivo principal de la especificación de diseño es proporcionar una guía clara y completa para el desarrollo del chatbot, estableciendo una base sólida sobre la cual los desarrolladores puedan implementar el sistema. Este informe servirá como referencia y documentación para futuras etapas del proyecto y posibles mejoras o expansiones.

En esta sección se describirán y analizarán en detalle los diferentes aspectos del diseño, desde la estructura y organización de los datos hasta los procedimientos y algoritmos que permiten su funcionamiento adecuado. Por último, se profundizará en el diseño arquitectónico, que define la distribución de los componentes y la interacción entre ellos

## C.2. Diseño de datos

Para empezar, se detallarán como se organizan los datos en la aplicación.

### Base de datos

En el proyecto se utiliza una base de datos NoSQL para almacenar y gestionar la multitud de respuestas a las preguntas de los usuarios que interactúan con los chatbots. . En el entorno de AWS, se emplea Amazon DynamoDB, mientras que en Google Cloud se utiliza Firestore.

El término NoSQL se refiere a un conjunto de tecnologías de bases de datos diseñadas para gestionar datos de forma distintas a los modelos tradicionales. A

diferencia de las bases de datos SQL, que utilizan un esquema rígido y un lenguaje de consulta estructurado, estas bases de datos NoSQL ofrecen una mayor flexibilidad al permitir la definición dinámica de esquemas. Para ello, cada uno de los servicios proporcionan métodos con los que poder recuperar o introducir datos.

El funcionamiento de firestore es algo complejo, organizando las bases de datos mediante colecciones dentro de una única tabla. Cada colección sería lo correspondiente a una o varias tablas del estilo tradicional. A su vez, dentro de cada colección se encuentran los documentos, que equivale a las filas de la tabla donde dentro de cada documento se encuentran los campos. Los documentos por defecto obtienen un ID aleatorio, y cada documento puede tener distintos campos a pesar de estar dentro de una misma colección. Esta es la peculiaridad de las bases de datos noSQL: La flexibilidad.

DynamoDB permite definir tablas con claves de partición para poder recuperar grupos de datos, así como de ordenación para poder ordenar los datos dentro de su grupo. Además, permite organizar los datos en tablas, lo que se asemeja a un estilo un poco más tradicional. A diferencia de GCP, no permite tanta flexibilidad de datos dentro de una misma tabla, ya que te obliga a definir como mínimo el mismo tipo de clave de partición y ordenación para todos los datos.

## Firestore

En firestore son dos las tablas:

1. ChatbotResponses. Maneja las respuestas a las preguntas.
  - **IntentName:** campo de tipo **string** que guarda el nombre del intent que detectaría el bot para ser invocado.
  - **Question:** campo de tipo **string** que almacena la posible pregunta.
  - **email:** campo de tipo **string** que almacena la respuesta a la posible pregunta.
2. ChatbotSteps. Guarda información del nombre del documento con la información de los pasos a seguir.
  - **IntentName:** campo de tipo **string** que guarda el nombre del intent que detectaría el bot para ser invocado.
  - **Question:** campo de tipo **string** que almacena el paso.
  - **email:** campo de tipo **string** que almacena el nombre del documento correspondiente al paso donde se encuentra toda la información.

ChatbotResponses
Documento: IntentName_Question
IntentName: Cadena Question: Cadena Response: Cadena

Figura C.1: Diseño de bases de datos NoSQL para respuestas Firestore. Fuente: elaboración propia

ChatbotSteps
Documento: IntentName_Step
IntentName: Cadena Step: Cadena Response: Cadena

Figura C.2: Diseño de bases de datos NoSQL para pasos Firestore. Fuente: elaboración propia

## DynamoDB

En DynamoDB se ha utilizado una única tabla donde se guarda la información de las preguntas y de los pasos:

- **IntentName:** campo de tipo **string** que guarda el nombre del intent que detectaría el bot para ser invocado. Además, es clave de partición.
- **Question/steps:** campo de tipo **string** que almacena la posible pregunta o el paso. Además, es clave de ordenación.
- **email:** campo de tipo **string** que almacena la respuesta a la posible pregunta o el nombre del documento.

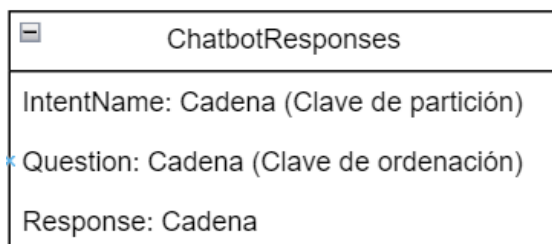


Figura C.3: Diseño de bases de datos NoSQL DynamoDB. Fuente: elaboración propia

### C.3. Diseño procedimental

Todo el diseño procedimental se divide en dos partes, tanto para AWS como GCP:

- Secuencia desde que se sube un archivo PDF hasta que se ha extraído el texto. [C.4](#), [C.5](#)
- Secuencia donde el usuario interactúa con el chatbot. [C.6](#), [C.7](#)

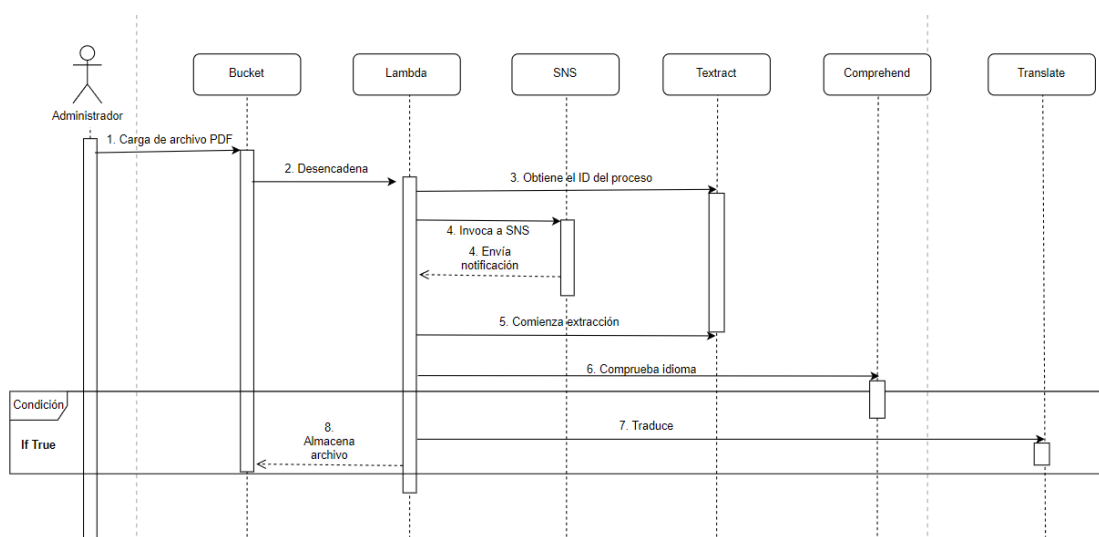


Figura C.4: Diagrama de secuencia para la extracción de texto AWS. Fuente: elaboración propia



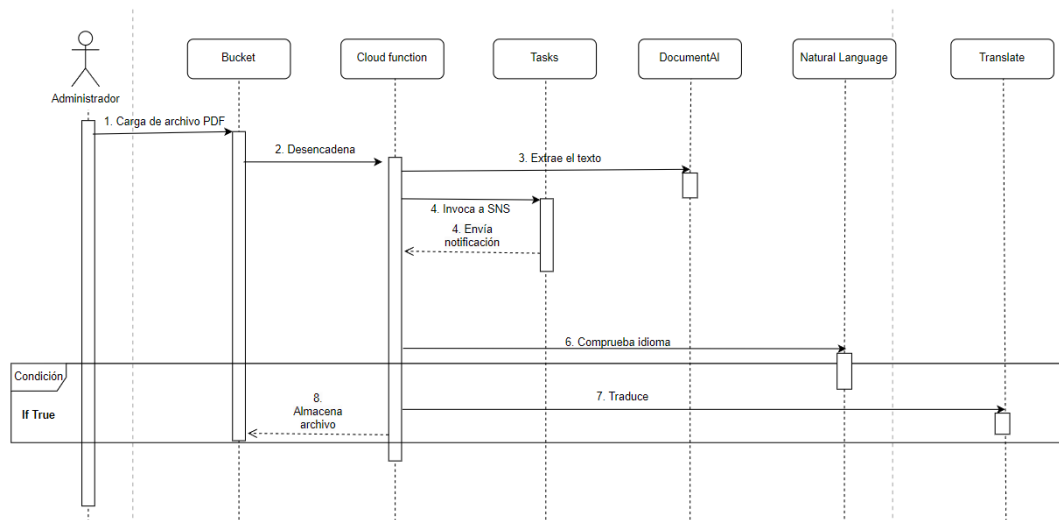


Figura C.5: Diagrama de secuencia para la extracción de texto GCP. Fuente: elaboración propia

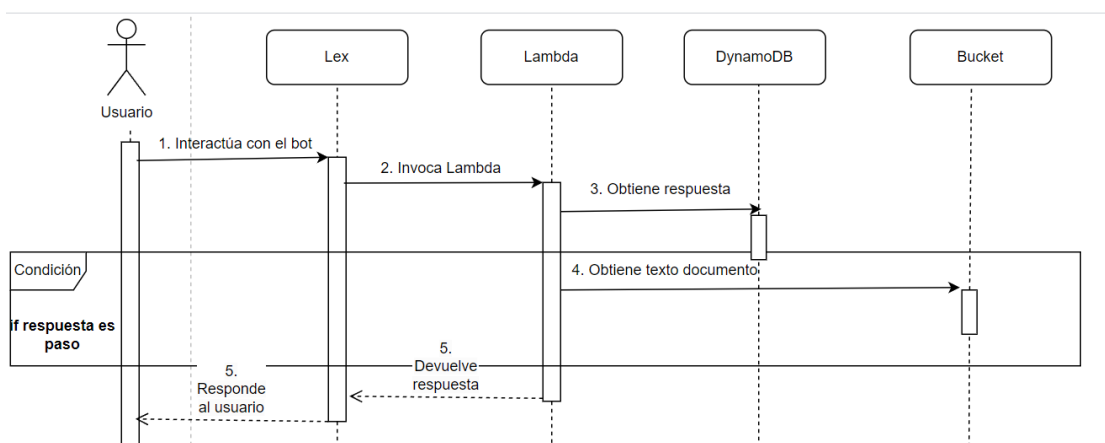


Figura C.6: Diagrama de secuencia para el bot AWS. Fuente: elaboración propia

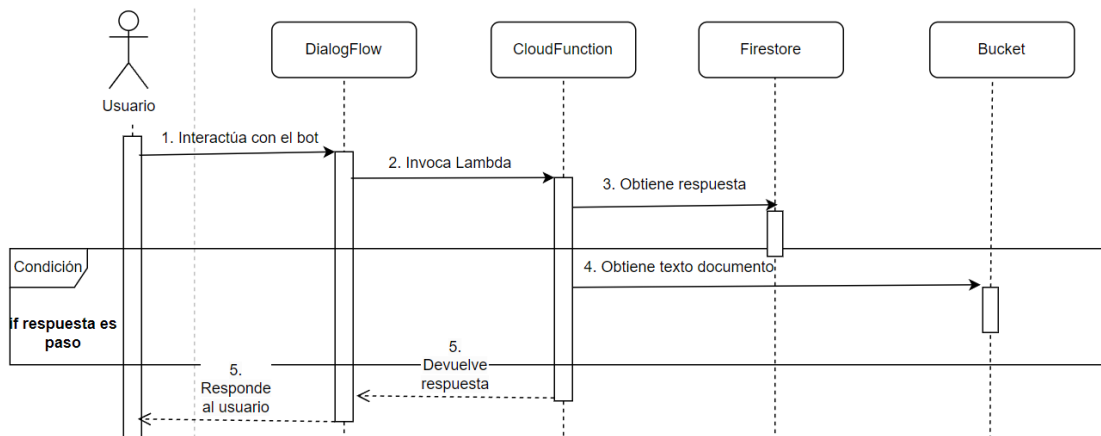


Figura C.7: Diagrama de secuencia para el bot GCP. Fuente: elaboración propia

## C.4. Diseño arquitectónico

El diseño arquitectónico puede describirse mediante la arquitectura de micro-servicios basada en la nube. Este tipo de arquitectura permite descomponer el sistema en pequeños servicios independientes que pueden desarrollarse, desplegarse y escalarse de manera autónoma.

La arquitectura general se puede dividir en las siguientes capas y componentes, los cuales se comunican entre sí para proporcionar la funcionalidad completa del chatbot:

### 1. Capa de Almacenamiento de Datos:

- **AWS:** Utiliza Amazon S3 y DynamoDB.
- **Google Cloud:** Utiliza Google Cloud Storage y Firestore.

### 2. Capa de Procesamiento y Análisis de Datos:

- **AWS:** Utiliza AWS Lambda, Amazon Textract, Amazon Comprehend y Amazon Translate.
- **Google Cloud:** Utiliza Cloud Functions, Document AI, Cloud Language y Cloud Translate.

### 3. Capa de Notificación y Orquestación:

- **AWS:** Utiliza Amazon SNS para la notificación de eventos.

- **Google Cloud:** Utiliza Cloud Tasks para la notificación de eventos.

4. **Capa de Interacción con el Usuario:**

- **AWS:** Utiliza Amazon Lex para la implementación del chatbot.
- **Google Cloud:** Utiliza Dialogflow para la implementación del chatbot.



## *Apéndice D*

---

# Documentación técnica de programación

---

## D.1. Introducción

. En este anexo se recoge la documentación técnica de programación, que incluye recomendaciones para el entorno de desarrollo, la estructura de directorios, el diseño, implementación y funcionamiento de la aplicación y del resto de módulos de código.

## D.2. Estructura de directorios

El repositorio (disponible en <https://github.com/mariolopezmatamala/cloud-duo-aws-gcp>) se organiza en un total de 4 directorios principales (ver Figura D.1).

### Diagramas de paquetes y directorios

Directorio general de todo el proyecto. Distribución de paquetes

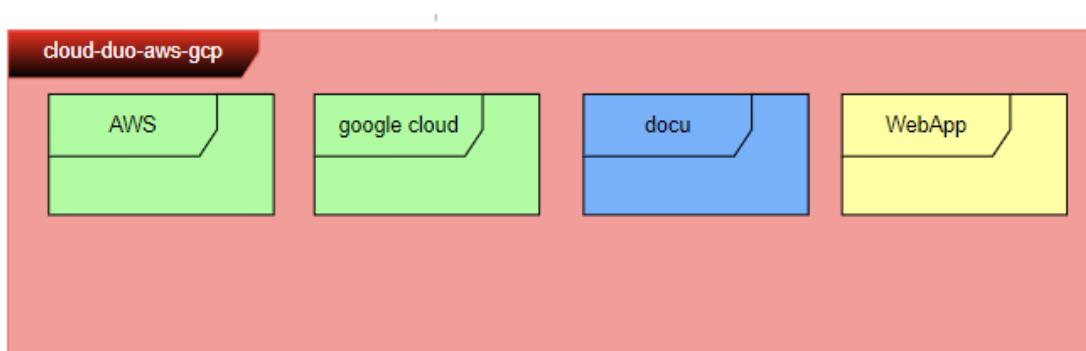


Figura D.1: Diagrama de paquetes general.

### Paquete AWS

En este paquete se almacenan los archivos correspondientes a las funciones lambda, así como los archivos en formato pdf que deben ser cargados para su extracción de texto.

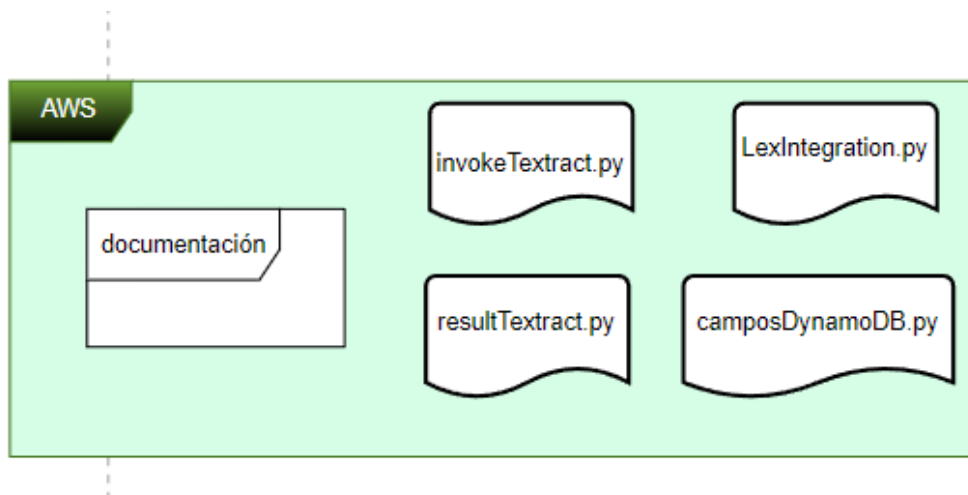


Figura D.2: Diagrama de paquetes AWS.

### Paquete google cloud

En este paquete se almacenan los archivos correspondientes a las cloud function de GCP, así como los archivos en formato pdf que deben ser cargados para su extracción de texto.

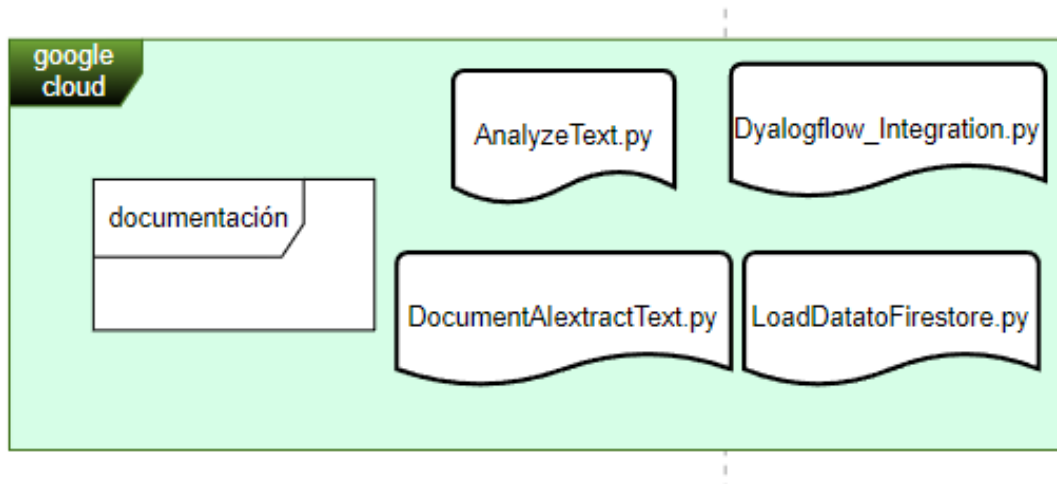


Figura D.3: Diagrama de paquetes GCP.

### Paquete docu

La documentación del proyecto puede ser consultada en este directorio. Los archivos están distribuidos de la siguiente manera:

- */tex*: recoge los capítulos y apéndices de la memoria y anexos.
- */img*: recoge todas las imágenes.

Además, se encuentran los archivos en formato .tex y .pdf de la memoria y los anexos, así como su bibliografía.

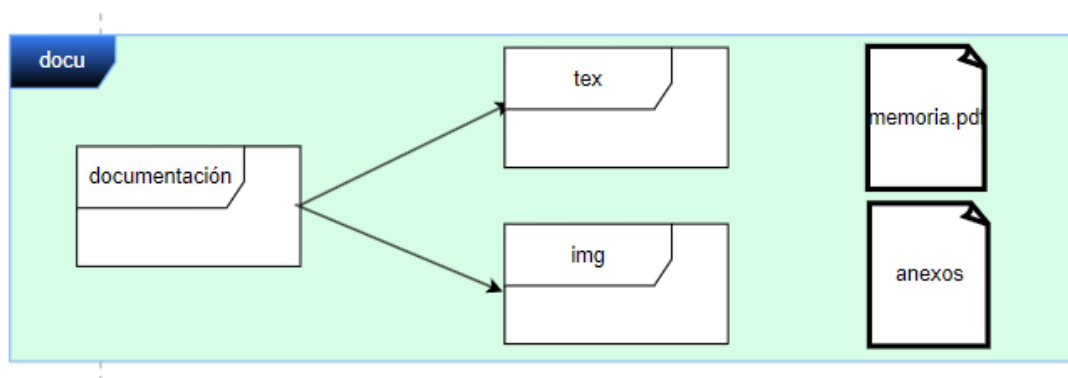


Figura D.4: Diagrama de paquetes documentación.

### Paquete webapp

Los archivos correspondientes a la webapp se encuentran en este paquete.

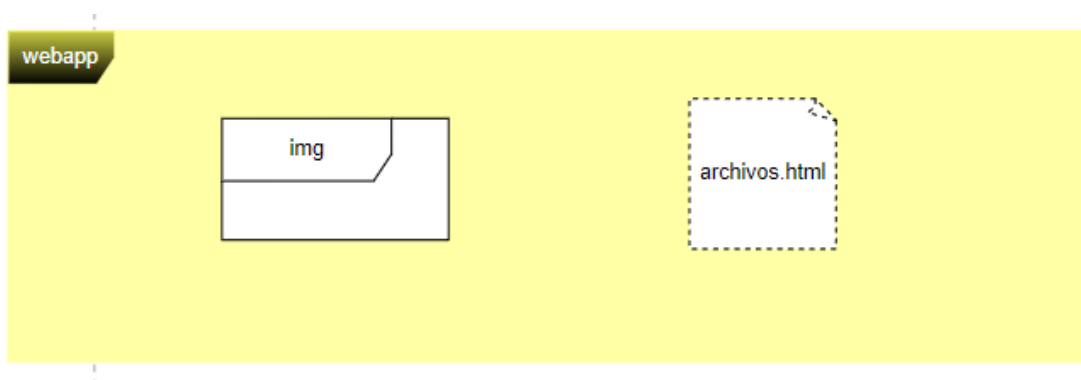


Figura D.5: Diagrama de paquetes webapp.

## D.3. Manual del programador

Esta sección está destinada a proporcionar información detallada sobre cómo utilizar el programa desarrollado en el proyecto. En él se detallan los pasos más relevantes que se han seguido para la creación y que pueden servir como referencia para futuros desarrolladores tanto para AWS como GCP:



## Amazon Web Services

Para la realización de todos los pasos estaremos utilizando la consola de AWS, se puede acceder desde este enlace: <https://aws.amazon.com/es/>

### 1. Creación de la cuenta

Para ello hay que proporcionar un email y un nombre de cuenta. Además, será necesario introducir una forma de pago que será utilizada en caso de que debamos pagar por el uso de los servicios al sobrepasar el límite de su capa gratuita.

AWS proporciona diferentes capas gratuitas, algunas están disponibles de forma indefinida, otras tienen una duración de meses o algunas son como pruebas. A continuación, proporciono una tabla de los servicios y su tipo de tarifa para nuevos usuarios (ver tabla D.1). Es interesante recalcar que todos

Servicio	Oferta
S3	12 meses gratis, 5 GB gratis, 20,000 solicitudes gratis
Lambda	1 millón de solicitudes gratis
SNS	1 millón de publicaciones gratis
Textract	1,000 páginas al mes gratis
Comprehend	12 meses gratis, 5 millones de caracteres al mes gratis
Translate	12 meses gratis, 2 millones de caracteres al mes gratis
DynamoDB	25 GB gratis de almacenamiento
Lex	12 meses gratis, 10,000 solicitudes al mes gratis

Tabla D.1: Servicios y sus ofertas correspondientes AWS

los servicios de AWS utilizados en este proyecto han tenido coste cero debido a estas capas gratuitas.

### 2. Creación del Rol IAM

Comenzando a trabajar con los servicios, es necesario configurar un Rol IAM en AWS. Este rol define los permisos que las funciones Lambda necesitarán para interactuar con otros servicios de AWS.

Primero, busca el servicio Identity and Access Management (IAM) y crea un nuevo rol en la sección Roles. Yo le he puesto el nombre “AWS\_PDF\_Textract\_Lex\_Role”. En el Paso 1, configura el tipo de entidad de confianza como Servicio de AWS, con el caso de uso Lambda. En el Paso 2, asigna los siguientes permisos: *AmazonS3FullAccess*, *AmazonTextractFullAccess*, *AWSLambdaExecute*, *AWSLambdaSNSFullAccess*, *ComprehendFullAccess* y *TranslateFullAccess*. En el Paso 3, elige un nombre para el rol y revisa el resumen de configuración. Finalmente, crea el rol.

Repite estos pasos para crear otro rol con los siguientes permisos, en mi caso lo llamé “AWS\_Lambda\_LexIntegration”: *AmazonDynamoDBFullAccess*, *AmazonS3FullAccess*, *AWSLambdaBasicExecutionRole* y *AmazonLexFullAccess*.

### 3. Creación de un Bucket S3 con dos Carpetas

El siguiente paso es crear un bucket en Amazon S3 para almacenar los archivos que se procesarán y los resultados. Busca el servicio Amazon S3 y crea un nuevo bucket, eligiendo un nombre y dejando todas las configuraciones por defecto para mayor seguridad. Dentro de este bucket, crea dos carpetas: una para cargar los PDF y otra para almacenar el texto extraído de los PDF en formato TXT.

### 4. Crear un SNS

Para la mensajería y las notificaciones, utiliza Amazon Simple Notification Service (SNS). Busca el servicio SNS y crea un nuevo tópico, asignándole un nombre y seleccionando el tipo estándar. Deja todas las configuraciones por defecto y crea el tópico.

### 5. Creación de las Funciones Lambda

En este paso, se crearán tres funciones Lambda que interactuarán con S3, Textract y Lex.

Primero, busca el servicio AWS Lambda y crea una nueva función desde cero. Asigna un nombre a la función y selecciona Python como el lenguaje de tiempo de ejecución, con arquitectura x86\_64. En la sección de permisos, añade el rol IAM “AWS\_PDF\_Textract\_Lex\_Role”. Después de crear la función, ve a la configuración básica y ajusta la memoria a 1GiB, el almacenamiento efímero y el tiempo de espera a 15 minutos. Configura un desencadenador

(trigger) para el bucket S3 que hemos creado, especificando el prefijo de la carpeta donde se cargarán los PDF y el sufijo `.pdf`.

La segunda función Lambda será similar a la primera, pero el desencadenador será el servicio SNS configurado anteriormente. Utiliza el mismo rol IAM que la primera función.

La tercera función Lambda no tendrá un desencadenador y puedes usar la misma configuración básica de las funciones anteriores, exceptuando el rol, que será el `"AWS_Lambda_LexIntegration"`.

## 6. Creación de la Base de Datos

Ahora vamos a configurar la base de datos donde el chatbot buscará respuestas a las preguntas. Busca el servicio Amazon DynamoDB y crea una nueva tabla. Introduce el nombre de la tabla y establece la clave de partición como `IntentName`. Esto permitirá recuperar el conjunto de preguntas asociadas a cada intent y proporcionar respuestas precisas. Deja el resto de las configuraciones por defecto y crea la tabla.

Después, crea una función Lambda para cargar los datos en la tabla DynamoDB. El código necesario para esta función se puede encontrar en un enlace de GitHub especificado. Una vez creada la función, realiza una prueba para asegurarte de que los datos se cargan correctamente.

## 7. Creación del Bot

Para crear el chatbot, accede al servicio AWS Lex y crea un nuevo bot desde cero. Asigna un nombre al bot y deja la configuración de permisos predeterminada, ya que las funciones Lambda previamente creadas serán las que interactúen con los servicios necesarios.

Crea una lista de intents y añade ejemplos de enunciados para cada uno. Configura los hooks de código utilizando las funciones Lambda creadas anteriormente. Los intents iniciales que debes crear incluyen `StartTutorial`, `RepeatStep`, `NextStep` y `GoToStep`. Añade frases de entrenamiento como "crear un chatbot" o "siguiente paso".

A continuación, configura los intents correspondientes al banco de preguntas. Asegúrate de utilizar los nombres de intents que coincidan con los utilizados en la función Lambda que carga los datos en DynamoDB, para garantizar la correcta detección y respuesta. Añade también las frases de entrenamiento similares a las preguntas que has introducido en la base de datos.

## 8. Implementación del Chatbot

Utilizaremos el servicio de `kommunicate` para implementarlo. Necesitarás crear un rol de usuario con permisos de Lex,

copia el ID y la credencial. Pega estos datos en Kommunicate y crea el bot, cuando te pida el Alias selecciona el borrador. Para implementarlo en tu página web, simplemente copia el código script de la instalación.

## Google Cloud Platform

Al igual que en AWS, estaremos utilizando la consola que proporciona GCP, sin embargo, también se puede configurar un entorno local para trabajar con los servicios de GCP. <https://cloud.google.com/?hl=es>

Los pasos son bastante similares a los de AWS, guardan bastante relación.

### 1. Creación de la cuenta

Lo primero será empezar creando una cuenta desde la consola. Se deberá proporcionar un email y un nombre de cuenta. Además, será necesario introducir una forma de pago que será utilizada en caso de que debamos pagar por el uso de los servicios al sobrepasar su capa gratuita.

GCP también proporciona una capa gratuita de 300USD durante 3 meses. Dentro de esta capa gratuita, también se incluyen varios servicios gratuitos dentro de los cuales se encuentran algunos de los que se usarán (ver tabla (ver tabla D.2)).

Servicio	Oferta
Cloud Storage	■ 5 GB de almacenamiento al mes
Firestore	■ 1 GB de almacenamiento
Cloud Functions	■ 2 millones de llamadas

Tabla D.2: Servicios y sus ofertas correspondientes en GCP

Para el resto de servicios, el precio por su uso se descontará desde los 300USD que proporciona GCP.

**Importante**

Es importante activar la cuenta antes de que terminen los tres meses de prueba, de lo contrario se perderá todo el trabajo realizado

**2. Creación del Proyecto**

Para poder empezar a trabajar se debe crear un proyecto nuevo en la consola de Google Cloud. Al acceder a la consola, se presenta la opción de crear un nuevo proyecto. Es importante guardar el ID del proyecto, ya que puede ser necesario más adelante para configurar el entorno virtual.

**3. Creación del Storage**

El primer paso es crear un bucket en el servicio Cloud Storage. Se debe configurar el nombre del bucket y la región deseada, dejando el resto de las configuraciones en su estado predeterminado. Una vez creado el bucket, se deben crear dos carpetas dentro de él: una para los archivos PDF que se desean procesar y otra para almacenar los archivos TXT con el texto extraído.

**4. Creación de Document AI**

A continuación, se configura el servicio Document AI. Al acceder a este servicio, se solicita habilitar la API correspondiente. Después, se procede a crear un procesador, eligiendo el tipo Document OCR para realizar la extracción de texto. Se le asigna un nombre y una región al procesador. Es recomendable mantener la misma región para todos los servicios relacionados con el proyecto. Una vez configurado, se puede probar subiendo un documento de prueba para verificar su correcto funcionamiento.

**5. Creación de las Cloud Functions**

En este paso, se crean tres funciones Cloud Functions, cada una con un propósito específico: extraer texto, procesar el texto y gestionar la interacción con DialogFlow.

La primera función se encarga de la extracción de texto. Se selecciona el entorno de 1ª generación por su estabilidad. Se configura el nombre y la región, y se establece como activador Cloud Storage con tipo de evento "finalized". Se ajustan los recursos de memoria, CPU y tiempo de espera para asegurar la finalización de la tarea. También se asigna la cuenta de servicio del proyecto.

Antes de configurar las otras funciones, se crea una cola en el servicio Cloud Tasks para gestionar las notificaciones. Se habilita la API de Cloud Tasks y se configura el nombre y la región de la cola.

La segunda función se activa mediante una notificación HTTP enviada por la primera función. Esta función realiza la tarea de procesar el texto, para lo cual necesita habilitar las API de Cloud Natural Language y Cloud Translate.

La tercera función se encarga de la integración con el bot de DialogFlow. Esta función se configura de manera similar a las anteriores, pero con el propósito específico de gestionar la interacción con DialogFlow.

## 6. Creación de la Base de Datos

Para almacenar el banco de preguntas y respuestas del chatbot, se utiliza Firestore. Se selecciona el modo Nativo y se asigna un nombre, dejando el resto de configuraciones por defecto. Una vez creada la base de datos, se añaden dos colecciones: una para el banco de preguntas y respuestas (`chatbotresponses`) y otra para los pasos de creación del chatbot (`chatbotsteps`).

## 7. Configuración de DialogFlow ES

Se utiliza DialogFlow ES (Essentials) para configurar el bot. En la consola de DialogFlow, se crean los intents necesarios, añadiendo frases de entrenamiento y habilitando la opción `.Enable webhook call for this intent.`<sup>en</sup> el fulfilment.

Los intents iniciales incluyen `StartTutorial`, `RepeatStep`, `NextStep` y `GoToStep`, con frases de entrenamiento como “crear un chatbot” o “siguiente paso”. También se configuran los intents correspondientes al banco de preguntas, utilizando los nombres de intents que recibe la Cloud Function y añadiendo frases de entrenamiento similares a las preguntas almacenadas en la base de datos.

## 8. Implementación del Chatbot

Finalmente, se implementa el chatbot en una página web. En la consola de DialogFlow, se navega a la configuración del bot y se añade una descripción. En el apartado de “Integrations” -> “Web demo”, se copia el código `<iframe>` y se inserta en la página web.

## *Apéndice E*

---

# Documentación de usuario

---

## E.1. Introducción

En esta sección se presenta un breve manual de usuario que permita interactuar con la herramienta de manera satisfactoria, aprovechando al máximo sus características y funcionalidades.

## E.2. Requisitos de usuarios

En este apartado se enumerarán los requisitos que el usuario debe cumplir para poder interactuar con la aplicación satisfactoriamente.

- Sistema Operativo: Indiferente, ya que el usuario únicamente interactúa con la aplicación web donde se encuentra alojado chatbot.
- Navegador web: Se recomienda a los usuarios utilizar un navegador web actualizado, como Google Chrome o Mozilla Firefox. Esto asegurará una mejor compatibilidad con las tecnologías utilizadas en la aplicación y ofrecerá una experiencia de usuario óptima.
- Conexión a Internet: Para utilizar la aplicación web, los usuarios deben tener acceso a una conexión estable a Internet. Esto permitirá la comunicación con el servidor donde se encuentra alojada la aplicación web y las aplicaciones en la nube.

## E.3. Instalación

Dado que se trata de una aplicación web, el usuario no necesita instalar ningún *software* en su dispositivo. Bastará con acceder (asegurándose siempre de tener conexión a Internet) a la dirección donde se aloja la aplicación: <https://aws-gcp-chatbot-webapp.s3.amazonaws.com/index.html>.

## E.4. Manual del usuario

### Página principal

La página principal es la página del comienzo por defecto, donde accederá en primer lugar el usuario. Aquí podrá leer una breve descripción del funcionamiento de la aplicación, así como de las dos plataformas implementadas. En la barra superior, podrá acceder a cualquiera de las dos.



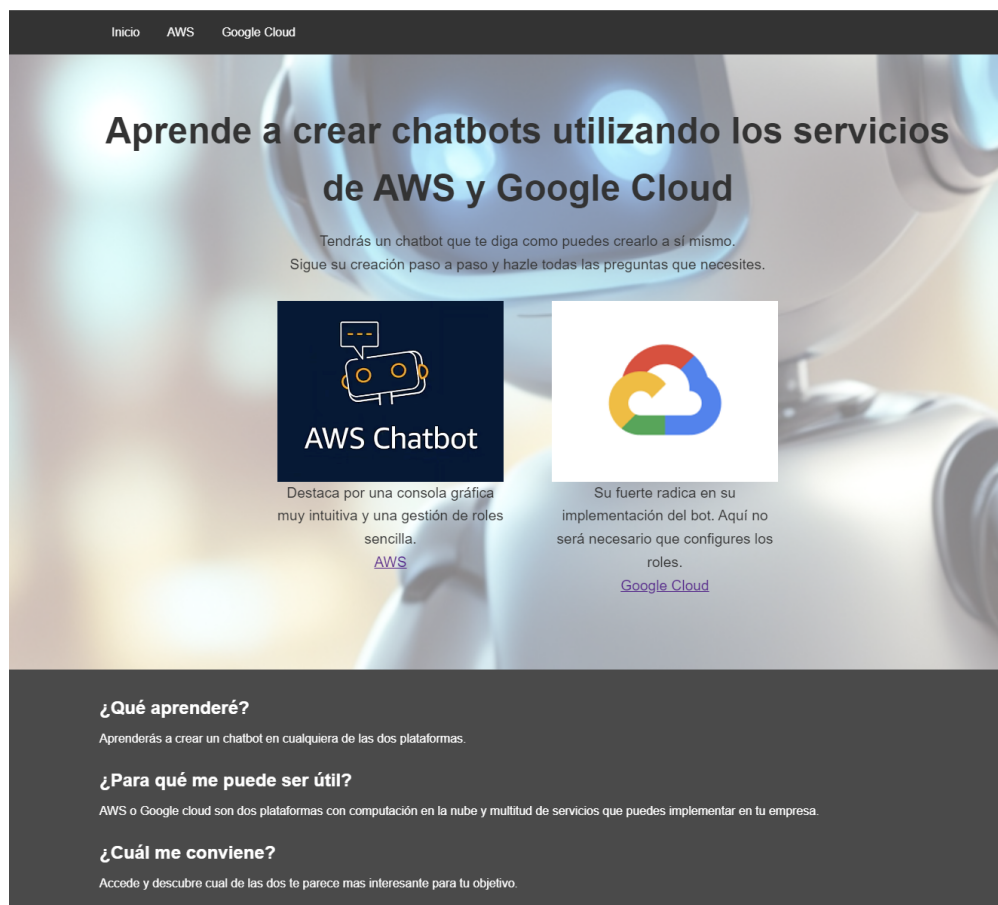


Figura E.1: Pagina principal de la webapp

## Página AWS

En esta página el usuario accede para interactuar con el bot de AWS. En la parte inferior derecha de la pantalla, le saldrá un desplegable donde podrá interactuar con el bot.

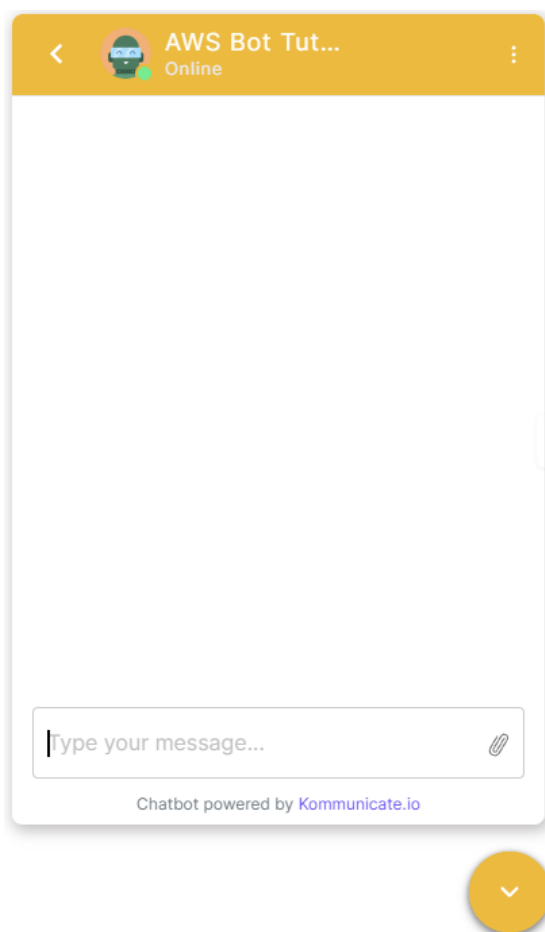


Figura E.2: Página AWS de la webapp

## Página GCP

En esta página el usuario accede para interactuar con el bot de GCP. En la parte inferior derecha de la pantalla, le saldrá un desplegable donde podrá interactuar con el bot.

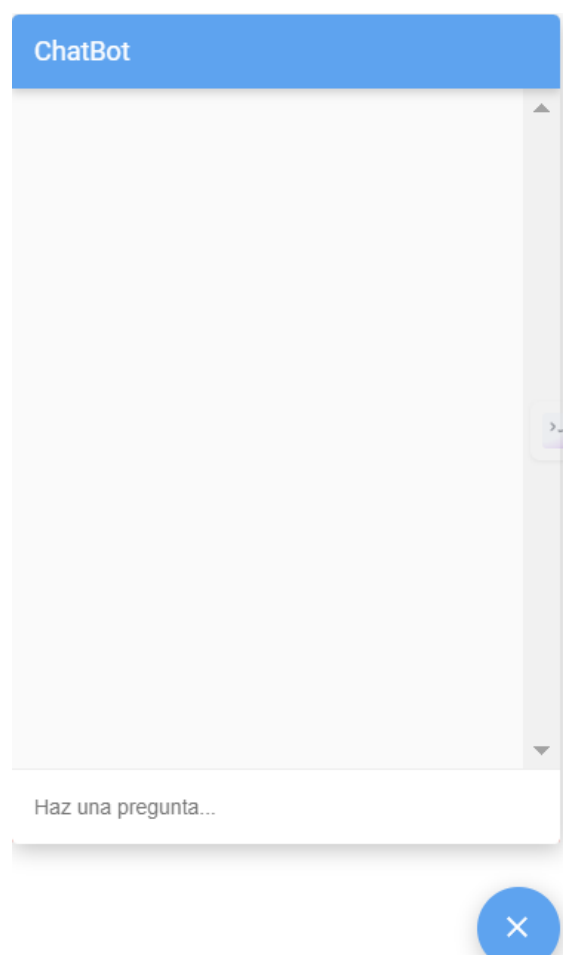


Figura E.3: Página GCP de la webapp



## *Apéndice F*

---

# Acrónimos

---

**AWS** Amazon Web Services

**GCP** Google Cloud Platform

**NLP** Natural Language Processing (Procesamiento del Lenguaje Natural)

**IAM** Identity and Access Management

**SNS** Simple Notification Service

**S3** Simple Storage Service

**IDE** Integrated Development Environment (Entorno de Desarrollo Integrado)

**POS** Part of Speech (Etiqueta de Partes del Discurso)

**NER** Named Entity Recognition (Reconocimiento de Entidades Nombradas)

**COV** Coverage

**Pytest-cov** Pytest Coverage

**HTML** HyperText Markup Language

**CSS** Cascading Style Sheets

**NLU** Natural Language Understanding (Comprensión del Lenguaje Natural)

**ML** Machine Learning (Aprendizaje Automático)

**AI** Artificial Intelligence (Inteligencia Artificial)

**GIT** Sistema de control de versiones

**SQL** Structured Query Language (Lenguaje de Consulta Estructurado)

**NoSQL** Not Only SQL (No solo SQL)

**OCR** Optical Character Recognition (Reconocimiento Óptico de Caracteres)

**URL** Uniform Resource Locator (Localizador Uniforme de Recursos)

**API** Application Programming Interface (Interfaz de Programación de Aplicaciones)

**GUI** Graphical User Interface (Interfaz Gráfica de Usuario)

**SDK** Software Development Kit (Kit de Desarrollo de Software)

**TFG** Trabajo Fin de Grado

---

## Bibliografía

---

- [1] Agencia tributaria: Irpf. <https://sede.agenciatributaria.gob.es/Sede/irpf.html>.
- [2] Agencia tributaria: Retenciones. <https://sede.agenciatributaria.gob.es/Sede/Retenciones.shtml>.
- [3] Jornada laboral, permisos y vacaciones - condiciones de trabajo, incluyendo trabajadores desplazados - trabajo y jubilación - ciudadanos - tus derechos y obligaciones en la ue - tu espacio europeo - punto de acceso general. [https://administracion.gob.es/pag\\_Home/Tu-espacio-europeo/derechos-obligaciones/ciudadanos/trabajo-jubilacion/condiciones-trabajo/jornada-permisos.html#-488e2841e18d](https://administracion.gob.es/pag_Home/Tu-espacio-europeo/derechos-obligaciones/ciudadanos/trabajo-jubilacion/condiciones-trabajo/jornada-permisos.html#-488e2841e18d).
- [4] Payscale. product owner salary. [https://www.payscale.com/research/ES/Job=Product\\_Owner/Salary](https://www.payscale.com/research/ES/Job=Product_Owner/Salary).
- [5] Payscale. scrum manager salary. <https://www.payscale.com/research/ES/Job=ScrumMaster/Salary>.
- [6] Seguridad social: Cotización. <https://www.seg-social.es/wps/portal/wss/internet/Trabajadores/CotizacionRecaudacionTrabajadores/10721/10957/583>.
- [7] Trámites para la puesta en marcha. <http://www.ipyme.org/es-ES/creaciondelaempresa/Paginas/Tramites.aspx>.
- [8] Jobted. slario programador, 2023. <https://www.jobted.es/salario/programador>.
- [9] Anthony E. Boardman, David H. Greenberg, Aidan R. Vining, and David L. Weimar. *Cost-benefit analysis: Concepts and practice*. Prentice Hall, 2006.

- [10] Git. *Git*. <https://git-scm.com/>, 2024. Online; Accedido el 20-Abr-2024.
- [11] Institute of Electrical and Electronics Engineers. IEEE Recommended Practice for Software Requirements Specifications. *IEEE Std 830-1998*, pages 1–40, 1998.
- [12] Marta Palacio. *Scrum Master. Temario troncal 1*. 06 2021.
- [13] Wikipedia. *Scrum (desarrollo de software)*. [https://es.wikipedia.org/wiki/Scrum\\_\(desarrollo\\_de\\_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)), 2024. Online; Accedido el 20-Abr-2024.





Esta obra está bajo una licencia Creative Commons  
Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional  
(CC BY-NC-SA 4.0 DEED).