

DESARROLLO DE UN PROTOTIPO PARA UN AGENTE  
CONVERSACIONAL INTELIGENTE AUTÓNOMO QUE PERMITA ESTUDIAR  
LA SOLEDAD EN LAS PERSONAS MAYORES APLICANDO TÉCNICAS DE  
INTELIGENCIA ARTIFICIAL

TRABALLO FIN DE GRAO  
PARA A TITULACIÓN DO  
GRAO EN ROBÓTICA

AUTOR:

BRUNO RECALDE FREIRE

TITORES:

SONIA MARÍA VALLADARES  
RODRÍGUEZ

DPTO ELECTRÓNICA E  
COMPUTACIÓN - USC

**UNIVERSIDADE DE SANTIAGO DE COMPOSTELA**

**TRABALLO FIN DE GRAO**

---

**DESARROLLO DE UN PROTOTIPO PARA UN AGENTE  
CONVERSACIONAL INTELIGENTE AUTÓNOMO QUE  
PERMITA ESTUDIAR LA SOLEDAD EN LAS PERSONAS  
MAYORES APLICANDO TÉCNICAS DE INTELIGENCIA  
ARTIFICIAL**

---

**AUTOR:**

**BRUNO  
RECALDE  
FREIRE**

**TITORES:**

**SONIA MARÍA  
VALLADARES  
RODRÍGUEZ**

*Este traballo preséntase para cumprir cos requirimentos normativos  
esixidos para acadar o Grao en Robótica  
na*

**Escola Politécnica Superior de Enxeñaría.**

Data da Convocatoria: Xaneiro de 2025

# Resumo

## **DESENVOLVEMENTO DUN PROTOTIPO PARA UN AXENTE CONVERSACIONAL INTELIXENTE AUTÓNOMO QUE PERMITA ESTUDAR A SOIDADE NAS PERSOAS MAIORES APLICANDO TÉCNICAS DE INTELIXENCIA ARTIFICIAL**

por  
Bruno Recalde Freire

**Palabras Chave:** detección de soidade social, axente conversacional, soidade social, intelixencia artificial, chatbot.

A soidade social é a sensación xerada polo illamento ou a falta de pertenza a un grupo, o que é independente da cantidade de interaccións sociais que unha persoa poida ter. Investigacións demostraron que a soidade presenta unha distribución en forma de U, onde os niveis máis altos acádanse na adolescencia e pasados os 80 anos [\[1\]](#). Unha serie de estudos expuxeron que a pandemia situou a existencia da soidade en persoas maiores no 28,6% e o illamento social no 31,2%, sendo uns valores cando menos alarmantes. Unha das razóns do crecemento da soidade social son os escasos servizos de saúde mental e terapeutas experimentados cos que contamos hoxe en día; tampouco axuda que sexan tratamentos que poden ser custosos ou as restricións xeográficas dalgunhas zonas illadas, que dificultan en gran medida o acceso a estes servizos.

En contraposición, o uso de intelixencias artificiais como os asistentes virtuais para o estudo de técnicas que permitan detectar estes casos está en aumento e é un campo en constante crecemento, empregando procesamento de texto, imaxes ou audio. O incremento do uso e investigación deste tipo de ferramentas neste sector foi en gran medida posible grazas á expansión da tecnoloxía na actualidade, o que favorece a posibilidade de chegar a un maior número de posibles afectados, aínda que a día de hoxe o número de proxectos que abordan este ámbito segue sendo bastante limitado [\[2\]](#).

Este proxecto propón o desenvolvemento dunha solución que permita realizar cribados masivos para detectar síntomas de soidade social. O proxecto consiste nun prototipo dun axente conversacional intelixente que sexa capaz de aplicar a escala de soidade social Este II interactuando co usuario mediante unha conversa fluída, aceptando variedade de respostas e xerando unha valoración final sobre os resultados do test. O proxecto desenvolveuse e probouse na ferramenta do entorno de AWS chamada Amazon Lex, ademais de contar cunha implementación en Facebook Messenger e Slack. Os resultados demostran que é unha ferramenta capaz de ofrecer apoio aos profesionais da saúde á hora de realizar un cribado sobre o estado de soidade nos usuarios.

# *Resumen*

## **DESARROLLO DE UN PROTOTIPO PARA UN AGENTE CONVERSACIONAL INTELIGENTE AUTÓNOMO QUE PERMITA ESTUDIAR LA SOLEDAD EN LAS PERSONAS MAYORES APLICANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL**

por  
Bruno Recalde Freire

**Palabras Clave:** detección de soledad social, agente conversacional, soledad social, inteligencia artificial, chatbot.

La soledad social es la sensación generada por el aislamiento o la falta de pertenencia a un grupo lo que es independiente de la cantidad de interacciones sociales que una persona pueda tener. Investigaciones han demostrado que la soledad presenta una distribución en forma de U, en donde los niveles más altos los alcanzamos en la adolescencia y pasados los 80 años. Una serie de investigaciones han expuesto que la pandemia ha situado la existencia de la soledad en adultos mayores en el 28,6% y el aislamiento social en el 31,2%, siendo unos valores cuanto menos alarmantes. Una de las razones del crecimiento de la soledad social son los escasos servicios de salud mental y terapeutas experimentados con los que contamos hoy en día, no ayuda además que son tratamientos que pueden llegar a ser costosos o las restricciones geográficas de algunas zonas aisladas que dificultan en gran medida el acceder a estos servicios.

En contraposición el uso de inteligencias artificiales como lo son los asistentes virtuales para el estudio de técnicas para detectar estos casos está en aumento y es un campo en constante crecimiento empleando procesamiento de texto, imágenes o audio. El incremento del uso e investigación de este tipo de herramientas en este sector ha sido en gran medida posible debido a la expansión de la tecnología en la actualidad, lo que favorece la posibilidad de llegar a un mayor número de posibles afectados, aunque a día de hoy el número de proyectos que abordan este ámbito sigue siendo bastante limitado.

Este proyecto propone el desarrollo de una solución que permita realizar cribados masivos que permitan detectar síntomas de soledad social. El proyecto consiste en un prototipo de un agente conversacional inteligente que sea capaz de aplicar la escala de soledad social Este II interactuando con el usuario mediante una conversación fluida, aceptando variedad de respuestas y generando una valoración final sobre los resultados del test. El proyecto se ha desarrollado y probado en la herramienta del entorno de AWS llamada Amazon Lex además de tener una implementación en Facebook Messenger y Slack. Los resultados demuestran que es una herramienta capaz de realizar apoyo para los profesionales de la salud a la hora de hacer un cribado sobre el estado de soledad en los usuarios.

# *Abstract*

## **DEVELOPMENT OF A PROTOTYPE FOR AN AUTONOMOUS INTELLIGENT CONVERSATIONAL AGENT THAT ALLOWS STUDYING LONELINESS IN OLDER PEOPLE APPLYING ARTIFICIAL INTELLIGENCE TECHNIQUES**

by  
Bruno Recalde Freire

**Keywords: detection of social loneliness, conversational agent, social loneliness, artificial intelligence, chatbot.**

Social loneliness is the feeling generated by isolation or the lack of belonging to a group, which is independent of the number of social interactions a person may have. Research has shown that loneliness follows a U-shaped distribution, with the highest levels occurring during adolescence and after the age of 80. Several studies have revealed that the pandemic has placed the prevalence of loneliness in older adults at 28.6% and social isolation at 31.2%, figures that are at least alarming. One of the reasons for the increase in social loneliness is the lack of mental health services and experienced therapists available today. Furthermore, these treatments can be costly, and geographic restrictions in some isolated areas significantly hinder access to these services.

In contrast, the use of artificial intelligence, such as virtual assistants, to study techniques for detecting such cases is on the rise and represents a constantly growing field. This growth employs text, image, or audio processing. The increased use and research of such tools in this sector have been largely driven by the expansion of technology today, facilitating the ability to reach a greater number of potential individuals affected. However, the number of projects addressing this field remains quite limited.

This project proposes the development of a solution to perform mass screenings to detect symptoms of social loneliness. The project consists of a prototype of an intelligent conversational agent capable of applying the Social Loneliness Scale Este II by interacting with users through a fluid conversation, accepting a variety of responses, and generating a final assessment based on the test results. The project was developed and tested using the AWS environment tool called Amazon Lex, with additional implementations on Facebook Messenger and Slack. The results demonstrate that it is a tool capable of supporting healthcare professionals in screening the state of loneliness in users.

# Agradecimientos

Quiero agradecer a mi tutora Sonia María Valladares Rodríguez, por su guía y apoyo a lo largo del desarrollo del proyecto.

También a mi familia por todo el apoyo que me dan siempre y la paciencia que tienen conmigo.

# Índice General

1 Introducción	1
2. Estado del conocimiento	3
3. Materiales	5
3.1. Amazon Lex	5
3.2. Medición de soledad social: Escala Este II	8
3.3. Modelos de Lenguaje	9
3.4. Paradise	10
3.5. Entrenamiento y Test	12
3.6. Cuestionario de Evaluación	13
4. Metodología	14
4.1. Plataforma de Desarrollo	14
4.2. Escala de Soledad	15
4.3. Diseño base del Agente	16
4.4. Implementación del Agente	17
4.4.1. Diseño del Agente en Amazon Lex	18
4.4.2. Función Lambda	20
4.4.3. Creación de las Frases de Entrenamiento	21
4.4.4. Implementación en aplicaciones	22
5. Pruebas	24
5.1. Plan de Pruebas	24
5.2. Resultados de las Pruebas en Amazon Lex	25
5.3. Resultado Pruebas con Usuarios	26
6. Análisis de resultados	29
7. Conclusiones	31
Apéndice	33
Bibliografía	35

# Índice de figuras

Figura 2.1 : Ejemplo de conversación con el chatbot SERENA	4
Figura 2.2 : Ejemplo de conversación con el asistente virtual Celia	4
Figura 3.1 : Estructura de un chatbot en Amazon Lex	7
Figura 3.3 : Matriz de confusión para cálculo del coeficiente Kappa	11
Figura 4.1 : Comparación entre escala Este II y la Escala de la UCLA	16
Figura 4.2 : Esquema del flujo de conversación diseñado	17
Figura 4.3 : Intents del agente en Amazon Lex	18
Figura 4.4 : Ejemplo de un Slot de un Intent de Pregunta	19
Figura 4.6 : Función Lambda para el control del flujo	21
Figura 4.7 : Ejemplos de frases de entrenamiento utilizadas	22
Figura 4.8 : Conversación del agente mediante la plataforma de Slack	23
Figura 4.9 : Canales para implementación en Messenger o Slack	23
Figura 4.10 : Conversación del agente mediante la plataforma de Messenger	24
Figura 5.1 : Matriz de confusión de las pruebas de texto en Lex	26
Figura 5.2 : Gráfica de tiempos en Lex	26
Figura 5.3 : Gráfica de tiempos en Slack	27
Figura Apéndice 1: Bot en Lex	33
Figura Apéndice 2: Importar el bot en Lex	33
Figura Apéndice 3: Importar el bot en Lex	34
Figura Apéndice 4: Comunicación Lex - Lambda	34

# Índice de Táboas

Tabla 3.1 : Escala de Soledad Social Este II	9
Tabla 3.2 : Niveles de puntuación en la Escala ESTE II	9
Tabla 3.3 : Cuestionario de Evaluación	13
Tabla 5.1 : Resultado pruebas en usuarios	28
Tabla 5.2 : Resultado cuestionario de evaluación de los usuarios	29



# 1 INTRODUCCIÓN

Un **agente conversacional** es una tecnología que tiene la capacidad de interactuar con los usuarios mediante el lenguaje natural pudiendo ser a través de conversaciones habladas, basadas en texto o multimodales. Estos agentes pueden recibir información a través de distintos sensores, procesarla e interactuar en función de esta, proporcionando retroalimentación, lo que lo convierte en una herramienta versátil y potente. Algunas características de los agentes conversacionales son:

- **Procesamiento de Lenguaje Natural (PLN):** tienen la capacidad para comprender y generar lenguaje humano de manera que la interacción sea lo más natural posible.
- **Aprendizaje Automático:** empleo de técnicas de aprendizaje automático (machine learning) para mejorar sus respuestas basándose en interacciones previas.
- **Interfaz Multimodal:** algunos agentes pueden interactuar a través de texto, voz e incluso señales visuales, adaptándose al contexto y preferencias del usuario. [\[3\]](#)

Para el desarrollo de este trabajo desarrollaremos un agente conversacional de tipo chatbot, utilizados en plataformas de mensajería como puede ser Facebook Messenger, aunque además se planteará su integración en un asistente virtual en este caso el Alexa de Amazon. Algo interesante en este trabajo es conseguir integrar el agente para que esté disponible en cualquier tipo de entorno y usuario con el fin de abarcar la mayor cantidad de estos.

Los agentes conversacionales cuentan con una gran ventaja y es que generan un alto grado de comodidad y confidencialidad en el usuario. En España, la salud mental se ha convertido en uno de los principales problemas de salud en la población [\[4\]](#). Es muy común que una persona decida no expresar abiertamente sus sentimientos o maquille la realidad para evitar la posibilidad de ser juzgados de alguna forma. Otra de las ventajas es su capacidad de accesibilidad para personas que tengan algún tipo de impedimento para desplazarse o que no cuenten con los recursos necesarios.

Los objetivos de este proyecto son, crear un prototipo de un agente conversacional que sea capaz de administrar en los usuarios una escala que identifique signos de soledad social mediante una conversación fluida y sea capaz de interpretar distintas respuestas, implementar el agente en gran variedad de dispositivos empleando tanto plataformas de mensajería como asistentes virtuales y realizar algunas pruebas con usuarios reales para recabar información y analizar resultados con el fin de optimizar el agente e intentar proveer unos resultados lo más óptimos posible.

En cualquier caso, este proyecto pretende ser una herramienta accesible y cómoda para los usuarios rebajando el nivel de incomodidad que se pudiese llegar a producir en caso de impartirlo

una persona real, pero en ningún caso se pretende que el agente sustituya el trabajo de ningún profesional del sector de la salud mental, la finalidad de este agente es ser una herramienta accesible y cómoda que permita realizar cribados y diagnósticos lo más tempranos posibles en los usuarios.

## 2. ESTADO DEL CONOCIMIENTO

El uso de la inteligencia artificial (**IA**) para la detección o tratamiento de la soledad social en las personas es un campo de investigación que está en un continuo crecimiento hoy en día, muchos proyectos e investigaciones aprovechan técnicas avanzadas en aprendizaje automático y procesamiento del lenguaje natural para tratar de identificar síntomas o patrones relacionados con la soledad social en las personas.

Uno de los enfoques más utilizados es el análisis del comportamiento en el hogar. Para ello se emplean sensores de movimiento además de otros sensores instalados en las puertas de sus hogares para conseguir monitorear completamente los movimientos de los usuarios. A parte se monitorean el número de llamadas y uso de teléfonos y el comportamiento informático con software. Todo esto para investigar la relación entre el tiempo fuera del hogar, el número de tiempo de uso de computadora o de llamadas telefónicas y la soledad de las personas. [5]

Otro enfoque destacado es el procesamiento del lenguaje natural para evaluar la soledad a partir del análisis de conversaciones con personas mayores. Para ello se realizan entrevistas sobre su experiencia de soledad o la aplicación de diversas escalas de soledad como por ejemplo, podría ser la Escala Este II. Con esto, se busca encontrar indicios de tristeza o respuestas más largas al responder preguntas relacionadas con la soledad [6]. Además se realizan análisis de las expresiones faciales empleando procesamiento de imágenes y videos además del uso de distintos algoritmos de extracción de características faciales ya que estas pueden ser un indicativo de un estado emocional concreto. [7]

El empleo de agentes conversacionales inteligentes, a través de chatbots o asistentes virtuales, enfoque similar al de este proyecto, es otra de las áreas más utilizadas hoy en día. Empleando estos agentes, se analizan entradas de audio, texto o imágenes para investigar el estado emocional de los usuarios. Por ejemplo, en España nos podemos encontrar una serie de proyectos que abordan esta área de investigación como lo son el chatbot SERENA o el asistente virtual CELIA.

**SERENA** es un proyecto desarrollado con el objetivo de detectar y prevenir la soledad no deseada en personas mayores mediante un chatbot conversacional que interactúa con los usuarios para evaluar sus sentimientos de soledad. Su funcionamiento se basa en una conversación natural mediante dispositivos móviles, para analizar preguntas basadas en escalas diseñadas para la detección de la soledad, como puede ser por ejemplo, la Escala UCLA. Además, es capaz de analizar respuestas abiertas, que permiten a los usuarios expresar libremente sus sentimientos, lo que proporciona información muy valiosa para evaluar el estado emocional [8]. Utilizando algoritmos de aprendizaje automático, SERENA analiza las respuestas para identificar patrones lingüísticos que estén relacionados con la soledad. Esto facilita la detección temprana de situaciones de soledad social o no deseada, permitiendo intervenciones rápidas y eficientes de profesionales de la salud. [9]

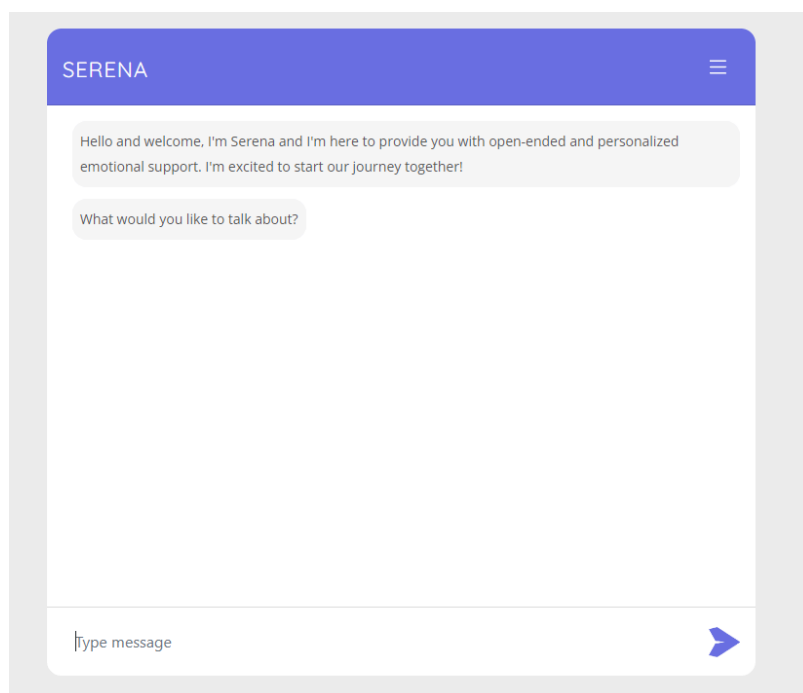


Figura 2.1 : Ejemplo de conversación con el chatbot SERENA

Por otro lado, **Celia** es un asistente conversacional desarrollado con el objetivo de combatir y monitorear la soledad social no deseada en personas mayores. Esta, utiliza inteligencia artificial y procesamiento de lenguaje natural para interactuar de forma proactiva con los usuarios a través de plataformas accesibles como WhatsApp. Además de analizar patrones en el lenguaje, es capaz de proporcionar actividades como juegos para estimular la mente, mientras detecta señales de aislamiento o de deterioro cognitivo que podrían ser indicativos de alguna enfermedad y también tiene la capacidad de aplicar escalas de detección de la soledad al igual que SERENA. Celia, se diseñó empleando algoritmos de aprendizaje automático para entrenar al asistente utilizando datos obtenidos de entrevistas con personas mayores para ayudar a mejorar su calidad de vida y estado emocional. El proyecto usa como plataforma principal WhatsApp, debido a la cantidad de usuarios que utilizan la herramienta y su facilidad de uso, esta se agrega como un contacto en la aplicación convirtiéndose en un acceso directo y amigable. [\[10\]](#)

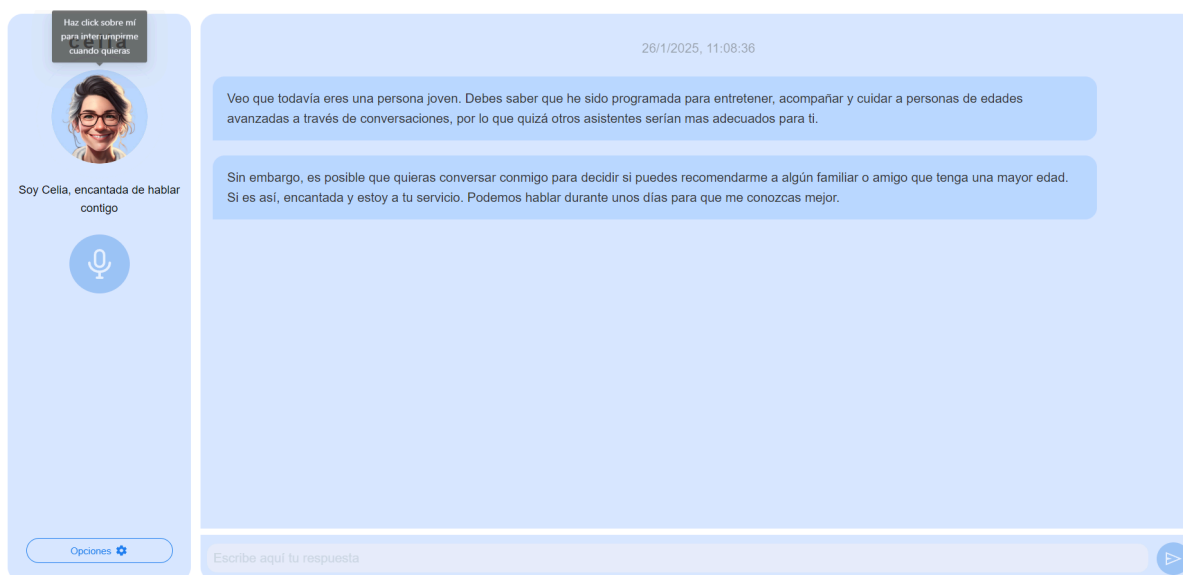


Figura 2.2 : Ejemplo de conversación con el asistente virtual Celia

## 3. MATERIALES

En esta sección se describen y explican los materiales que se emplearon para realizar el trabajo, en la siguiente sección se tratarán las razones por la cuales se escogieron estos por encima de otras posibles alternativas.

### 3.1. Amazon Lex

Para el desarrollo del agente conversacional en este proyecto, decidí utilizar la herramienta de Amazon Lex, un servicio de Amazon Web Services (AWS).

Amazon Lex utiliza las mismas tecnologías de aprendizaje profundo que Alexa, además de ser una plataforma que simplifica en gran medida la creación de agentes conversacionales. Es una herramienta capaz de captar entradas producidas por el usuario de tipo audio o texto y producir además respuestas de estos mismos tipos. [\[11\]](#)

Una de las principales ventajas de Amazon Lex es su capacidad para integrarse en otros servicios de AWS, como por ejemplo AWS Lambda el cual empleamos para manejar el flujo de conversación y eventos que tendrá el chatbot a la hora de interactuar con el usuario.

Además, Amazon Lex ofrece soporte para múltiples plataformas de mensajería, lo que permite implementar el agente conversacional en diversos canales, como aplicaciones móviles, sitios web y servicios de chat como Facebook Messenger o Slack. Esta flexibilidad facilita poder llegar a una mayor cantidad de usuarios además de aportar una gran comodidad a la hora de realizar las pruebas con estos. [\[12\]](#)

Otra de las ventajas de la herramienta es su interfaz de usuario en la cual podemos ver pequeñas explicaciones de cada uno de los componentes además de dar soporte a una gran cantidad de idiomas lo que nos podría permitir internacionalizar el bot.

Para la construcción del chatbot de este proyecto se empleó la versión de Amazon Lex v2, versión actual de la herramienta. Para comprender el desarrollo del agente conversacional en Amazon Lex, es fundamental familiarizarse con varios conceptos clave de la herramienta:

- **Intent:** representan las acciones e intenciones que el agente debe resolver para satisfacer las solicitudes de un usuario. Los intents contienen ejemplos de frases de usuarios las cuales se emplean para invocar a ese intent en particular. A parte tenemos el intent llamado “Default Fallback Intent” que se activa en caso de recibir una entrada que no coincide con ningún intent de los predefinidos en el bot. En este proyecto se creó un intent asociado a cada una

de las preguntas del cuestionario y después uno inicial para gestionar la bienvenida y el comienzo de este.

- **Slots:** los slots son los encargados de verificar si se proporcionó la información necesaria para que se cumpla la intención o en caso contrario solicitar esa información al usuario, son una parte fundamental ya que será la herramienta que emplearemos para hacer las preguntas del cuestionario e ir así cumpliendo los requisitos de los intents asociados a estos.
- **Slot Types:** mediante estos podremos generar y entrenar las respuestas que va a recibir el agente y que así a la hora de realizar el cuestionario, sea capaz de asociar las entradas a cada una de las respuestas posibles que tiene la escala que escogimos. Por ejemplo, en caso de que el bot reciba una entrada como “Si, me siento triste”, este la interpretará como si la respuesta recibida a la hora de cumplir el slot fuese en este caso “Siempre”.
- **Función Lambda:** Amazon Lex permite realizar una integración con AWS Lambda para ejecutar lógica personalizada. Las funciones Lambda pueden utilizarse para validar las entradas del usuario o realizar acciones específicas en función de la conversación, esto es algo que emplearemos para tratar las entradas recibidas y que el bot trabaje en función de estas y para crear el flujo de la conversación del usuario y el agente a lo largo del cuestionario.
- **CloudWatch:** en esta sección se recopilan todas las llamadas a la función lambda, con lo que nos será útil a la hora de detectar cualquier tipo de error que se pueda estar produciendo en ésta, además se puede realizar una monitorización de las entradas de los usuarios que estén utilizando el agente, por lo que es una herramienta a tener en cuenta.

Comprender y configurar correctamente estos conceptos es esencial para desarrollar un agente conversacional en Amazon Lex, capaz de interactuar de manera natural y eficiente con los usuarios.

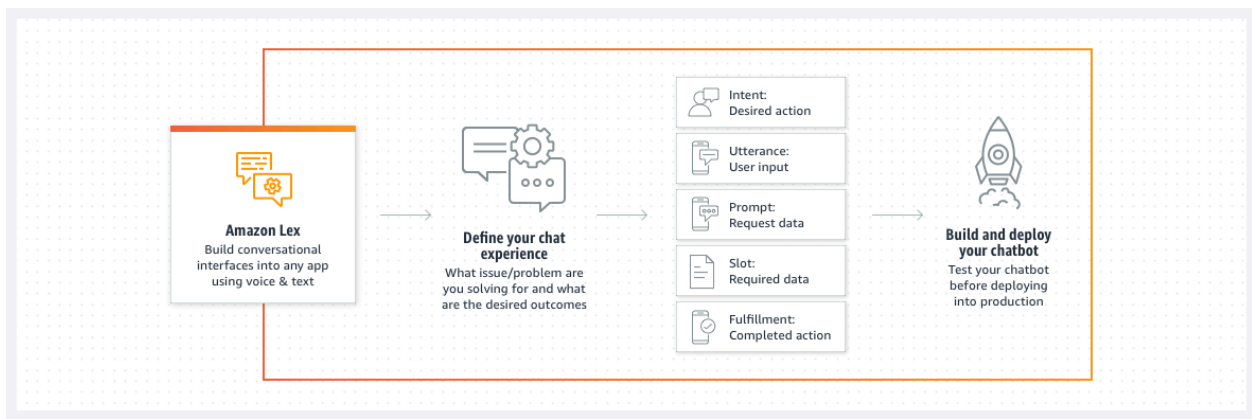


Figura 3.1 : Estructura de un chatbot en Amazon Lex

El funcionamiento de esta plataforma, a grandes rasgos, es el siguiente:

- **Entrada usuario:** el usuario envía una entrada de texto o de voz al agente para empezar con la conversación. Si el mensaje se recibe por voz, Amazon Lex cuenta con un módulo de NLP que transcribe automáticamente la entrada, lo que nos permite trabajar directamente con la entrada en formato texto.
- **Chequeo del intent:** una vez que se recibe la entrada del usuario, se analiza el texto para identificar que intent es con el que mejor coincide comparando además con las frases de ejemplo que forman los intents.
- **Solicitud de Información:** en caso de que un intent requiera de información adicional para completar la solicitud, se pedirá al usuario la información necesaria para procesar el intent la cual es almacenada en los slots.
- **Generación de la Respuesta:** una vez que se procesa la solicitud, Lex devuelve una respuesta al usuario que puede ser tanto en formato texto como mediante una voz sintética.
- **Seguimiento del Contexto:** lex es capaz de gestionar diálogos manteniendo el contexto de la conversación mediante los “Session Attributes” los cuales guardan información durante la interacción con el usuario permitiendo al bot entender la relación entre distintas partes de una misma conversación.
- **Gestión de errores:** en caso de que el bot no sea capaz de comprender la solicitud del usuario se activará el ya nombrado “Default Fallback Intent”, este será el encargado de indicarle al usuario que su solicitud no pudo ser procesada a través de un mensaje personalizado.

### 3.2. Medición de soledad social: Escala Este II

Para empezar a tratar los materiales empleados en el trabajo no hay mejor forma que con la escala o test empleado para la detección de la soledad a través de un agente conversacional. La elegida fue la Escala de Medición de Soledad Social: Escala Este II [13]. Esta es la sucesora de la previamente usada Escala Este I que fue creada en 1999 por Rubio Aleixandre [14], pero a diferencia de su predecesora la cual mide factores como la soledad conyugal, la soledad familiar, soledad existencial y soledad social, ésta estará enteramente centrada en esta última, la soledad social.

Para la construcción de esta se hizo una revisión bibliográfica de las principales escalas de medición de la soledad social tales como: la escala de Soledad de Russell (1980) [15], la escala de Soledad Social y emocional de Jong Gierveld (2006) [16], la adaptación española de la Escala de Evaluación de Soledad Social y Emocional en Adultos SESLA-S de Yárnoz (2008) [17]. A partir de ahí se generó una primera versión la cuál fue evolucionando a lo largo de una serie de fases y reuniones con expertos en distintas áreas del conocimiento psicológico y social para conseguir así llegar a una versión que consiguiera captar y analizar la soledad social entre la población de la tercera edad.

Es una de las más utilizadas hoy en día para medir y evaluar la soledad social, se trata de un cuestionario de 15 preguntas separadas en 3 factores, la percepción de apoyo social, el uso de nuevas tecnologías y la participación social subjetiva.

Factor 1: Percepción de Apoyo Social			
	SIEMPRE	A VECES	NUNCA
1. ¿Vd. tiene a alguien con quien puede hablar de sus problemas cotidianos?	0	1	2
2. ¿Cree que hay personas que se preocupan por usted?	0	1	2
3. ¿Tiene amigos o familiares cuando le hacen falta?	0	1	2
4. ¿Siente que no le hacen caso?	2	1	2
5. ¿Se siente triste?	2	1	0
6. ¿Se siente usted solo?	2	1	0
7. ¿Y por la noche, se siente solo?	2	1	0
8. ¿Se siente querido?	2	1	0
Factor 2: Uso de Nuevas tecnologías			
	SIEMPRE	A VECES	NUNCA



9. ¿Utiliza Ud. el teléfono móvil?	0	1	2
10. ¿Utiliza Ud. el ordenador (consola, juegos de la memoria)?	0	1	2
11. ¿Utiliza Ud. Internet?	0	1	2
Factor 3: Índice de Participación social subjetiva			
	SIEMPRE	A VECES	NUNCA
12. Durante la semana y los fines de semana ¿le llaman otras personas para salir a la calle?	0	1	2
13. ¿Le resulta fácil hacer amigos?	0	1	2
14. ¿Va a algún parque, asociación, hogar del pensionista donde se relacione con otros mayores?	0	1	2
15. ¿Le gusta participar en las actividades de ocio que se organizan en su barrio/pueblo?	0	1	2
PUNTUACIÓN EN SOLEDAD SOCIAL _____ Pts.			

Tabla 3.1 : Escala de Soledad Social Este II

La encuesta en cuestión acepta 3 tipos de respuestas (Siempre, A veces, Nunca) las cuales como podemos observar tienen asociada una puntuación. Según los resultados finales de la encuesta, se diferenciaron tres niveles de Soledad Social en función de la puntuación obtenida:

Soledad Social	
NIVELES	PUNTUACIÓN
Bajo	0 a 10 puntos
Medio	11 a 20 puntos
Alto	21 a 30 puntos

Tabla 3.2 : Niveles de puntuación en la Escala ESTE II

En el contexto de este proyecto, la Escala de Soledad Social ESTE II se pretende implementar a través de un agente conversacional, permitiendo una conversación fluida y accesible para los usuarios por lo que además de las respuestas predefinidas de la escala se entrenará el agente conversacional para que acepte una mayor cantidad de respuestas y las asocie a las de la escala en cuestión. Con esta integración busca realizar y facilitar el cribado de detección de soledad social en personas mayores y conseguir así una actuación rápida y eficiente.

### 3.3. Modelos de Lenguaje

Para conseguir que el agente tenga la capacidad de asociar adecuadamente las respuestas de la escala de soledad social, tendremos que generar una batería de frases de entrenamiento con el fin de entrenar nuestro agente para que cumpla esa función. Las frases deberán de estar adaptadas a las preguntas de nuestro cuestionario.

Para obtener las frases de entrenamiento se emplearon respuestas generadas artificial y manualmente. Las herramientas empleadas para generar artificialmente las respuestas fueron las siguientes:

- **ChatGPT-4o:** modelo de lenguaje desarrollado por OpenAI que emplea inteligencia artificial para generar respuestas en lenguaje natural. Este modelo fue entrenado con una gran cantidad de información y datos de muchas fuentes que hace que pueda producir y comprender respuestas humanas. [\[18\]](#)
- **Aria:** asistente de inteligencia artificial que utiliza el navegador Opera capaz de generar texto e imágenes basadas en los requisitos del usuario. [\[19\]](#)
- **Gemini:** similar al de Opera pero desarrollado por Google puede manejar una variedad de tipos de datos, incluyendo texto e imágenes. [\[20\]](#)

### 3.4. Paradise

El Paradigm for Dialogue System Evaluation (PARADISE) es un framework diseñado para evaluar sistemas de diálogo multimodales y orientados a tareas. Propuesto por Marilyn A. Walker en 1997, busca maximizar y evaluar los agentes conversacionales tanto en cuestiones de eficacia como en la calidad de las interacciones de este con el usuario. [\[21\]](#)

Para ello se evalúa el coeficiente Kappa (K) el cual nos indica lo bien que el sistema realizó una tarea concreta dentro de un escenario determinado. Para calcularlo tendremos que utilizar una matriz de confusión. En la matriz de confusión se muestra el número de veces que se obtuvo el valor correcto y, en el caso de los valores erróneos, qué valor se sustituyó.

	Key													
	Depart-City				Arrival-City				Depart-Range		Depart-Time			
Data	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14
v1	<b>22</b>		1		3									
v2		<b>29</b>												
v3	4		<b>16</b>	4			1							
v4	1	1	5	<b>11</b>			1							
v5	3				<b>20</b>									
v6						<b>22</b>								
v7			2		1	1	<b>20</b>	5						
v8			1		1	2	8	<b>15</b>						
v9									<b>45</b>	10				
v10									5	<b>40</b>				
v11											<b>20</b>		2	
v12											1	<b>19</b>	2	4
v13											2		<b>18</b>	
v14											2	6	3	<b>21</b>
Sum	30	30	25	15	25	25	30	20	50	50	25	25	25	25

Figura 3.3 : Matriz de confusión para cálculo del coeficiente Kappa

Para entender la matriz debemos tener en cuenta que:

- Las columnas las conforman los valores correctos en cada una de las preguntas y las filas las posibles respuestas entre las que se podría clasificar. En nuestro caso serían “Siempre”, “A veces” y “Nunca”.
- Los valores de la diagonal que están en negrita son las respuestas que se clasificaron correctamente.
- Los valores que aparecen fuera de esta se tratan de errores que cometió el agente a la hora de realizar la clasificación y en qué valor erróneo fueron añadidas.

Con la información de la matriz de confusión calcularemos el Kappa:

$$K = P(A) - P(E) / 1 - P(E)$$

P(A) indica cuán eficiente que fue el sistema al reconocer cada uno de los valores. Para el cálculo de P(A) únicamente tendremos que dividir el sumatorio de los valores que clasificaron correctamente (los de la diagonal), entre el total.

$P(E)$  representa el número de veces que se espera que los valores coincidan al azar. Para el cálculo de  $P(E)$  tendremos que:

- Hacer el producto de la suma de los valores de cada columna y fila:
  - Suma de los valores de la columna  $n1$  por los de la fila  $n1$  así hasta completar todas las filas y columnas.
- Se divide cada uno de los valores anteriormente calculados entre el cuadrado del total.
- Por último, sumaremos todos los valores y ya tendríamos calculado  $P(E)$

Al hacer el cálculo del coeficiente Kappa( $k$ ), cuanto más cerca esté el valor resultante de 1, con mayor éxito habrá sido capaz de resolver las tareas el agente.

### 3.5. Entrenamiento y Test

Si queremos que nuestro agente sea capaz tanto de administrar la escala de forma fluida como de comprender y seleccionar de forma correcta las entradas generadas por los usuarios, tendremos que entrenar a nuestro agente con las frases de entrenamiento previamente nombradas. El conjunto de entrenamiento consta de una serie de oraciones que representan cada una de las tres respuestas posibles para las preguntas de la Escala Este II.

Para cada una de las respuestas posibles se generó una batería de setenta y cinco posibles respuestas que se emplearon para entrenar al agente, veinticinco para cada una de las respuestas del test. Por ejemplo, para una pregunta al azar del test:

4. ¿Siente que no le hacen caso?

Nos podremos encontrar las siguientes frases de entrenamiento para cada respuesta:

- Respuesta "Siempre":
  - "Si, me pasa a menudo"
  - "La verdad es que sí"
  - "Constantemente "

- Respuesta" A veces":
  - "De vez en cuando"
  - "No siempre, pero sí"
  - "En ciertas ocasiones "
  
- Respuesta" Siempre":
  - "En ningún momento"
  - "Nunca me pasa"
  - "No suele pasar "

### 3.6. Cuestionario de Evaluación

Se creó el siguiente cuestionario para entregar a los usuarios que participan en las pruebas con la herramienta, con el fin de recabar datos sobre el grado de satisfacción en una escala del 1 al 5 de distintos aspectos relacionados con las pruebas y el agente y así poder revisar los cambios en función de los resultados que se generen.

PREGUNTAS	1	2	3	4	5
ÍNDICE DE SATISFACCIÓN	MUY BAJO	BAJO	MEDIO	ALTO	MUY ALTO
1. ¿Estás satisfecho/a con la dificultad de utilizar la herramienta?					
2. ¿Te sientes cómodo/a compartiendo tus sentimientos con el agente?					
3. ¿Te sientes más cómodo/a respondiendo las preguntas por voz que a través de mensajes de texto?					
4. ¿Te sientes cómodo/a empleando el asistente en una plataforma como Slack?					
5. ¿Consideras que puede ser una herramienta útil?					
6. ¿Recomendarías el uso de la herramienta?					

Tabla 3.3 : Cuestionario de Evaluación

La primera pregunta está destinada a evaluar la facilidad del uso de la herramienta, las tres siguientes, se diseñaron para evaluar la satisfacción de los usuarios a la hora de emplear el agente en el contexto de la prueba. Las últimas dos preguntas tratan sobre la relevancia social que los usuarios creen que tiene la herramienta.

## 4. METODOLOGÍA

Ahora pasaré a comentar la metodología de trabajo, explicando los procedimientos y elecciones que se hicieron a lo largo del desarrollo de la implementación del agente.

Para comenzar con el desarrollo se decidió primero la plataforma en la que se iba a desarrollar el agente para así tener una base sobre la que trabajar y barajar posibilidades a la hora de decidir la posterior escala. A partir de ahí, se estudiaron diversas escalas de soledad social que se podrían implementar en el agente. A partir de ahí con el agente y la escala ya en claro se pasó a desarrollar el y modelar el agente para que pudiese aplicar esta escala en los usuarios de forma autónoma.

### 4.1. Plataforma de Desarrollo

Tras haber presentado ya la plataforma en la que se decidió realizar el desarrollo del agente para el proyecto, Amazon Lex, ahora se pasará a comentar las razones que llevaron a la decisión de utilizarla frente a otras opciones, aunque la razón que más peso generó en la elección final fue la capacidad de integración del agente en un dispositivo Alexa:

- **Rasa:** Es una plataforma de código abierto que permite construir chatbots y asistentes virtuales con un nivel de personalización muy alto. A diferencia de las otras opciones, Rasa ofrece control total sobre los datos y la implementación, pero a cambio requiere de un nivel de programación y conocimientos en aprendizaje automático bastante altos por lo que la curva de aprendizaje se sale de cuestión para el proyecto. [\[22\]](#)
- **Dialogflow:** Plataforma de Google diseñada para desarrollar agentes conversacionales que pueden integrarse en múltiples plataformas, como aplicaciones móviles o sitios web. Los datos se procesan a través de los servicios de Google y tiene una escalabilidad a través de la infraestructura de Google Cloud. Aunque la interfaz sea más intuitiva que la de Lex la capacidad para integrarlo en un dispositivo Alexa hace que finalmente se descarte. [\[23\]](#)

Finalmente, Amazon Lex fue la opción que se consideró más factible para crear el agente gracias a su capacidad de integración en distintos dispositivos en especial en Alexa. Otra característica importante son las tecnologías automáticas de reconocimiento del habla que nos proporciona la herramienta.

Amazon Lex permite publicar el bot con facilidad en diversos servicios directamente desde la consola de Amazon Lex, lo que reduce esfuerzo en cuestión de desarrollo multiplataforma como las siguientes plataformas de chat: Facebook Messenger, Slack y Twilio SMS.

## 4.2. Escala de Soledad

A la hora de detectar los síntomas de soledad social en los usuarios mediante un agente conversacional, necesitamos emplear una herramienta que nos permita realizar ese diagnóstico para poder realizar un cribado general y ayudar así estos a obtener de forma prematura posible ayuda profesional.

A parte de este cuestionario, se barajaron otras posibles opciones para implementar en el agente y se llegó a las siguientes conclusiones:

- **Escala de la soledad de UCLA:**

- Herramienta psicológica utilizada para medir los sentimientos subjetivos de soledad y aislamiento social en individuos.
- La versión que se barajó constaba de diez ítems que evalúan sentimientos asociados a la soledad de los usuarios.
- Ha sido criticada por su enfoque unidimensional ya que argumentan que, para comprender y abordar adecuadamente la soledad, es necesario tratar componentes más específicos, como la soledad social y emocional. [\[24\]](#)

- **Escala de Valoración Sociofamiliar TSO:**

- Herramienta diseñada para evaluar de forma objetiva y fiable el grado de soledad social en personas mayores permitiendo identificar síntomas que requieran intervención y mejorar así el bienestar de los usuarios.
- Se evalúan cinco dimensiones fundamentales entre las que se encuentran: estructura y función familiar, situación económica, vivienda, apoyos de la red y relaciones sociales.
- Su aplicación es rápida, sobre un par de minutos, proporcionando información sobre en qué área se necesita intervención.
- El problema que presenta es que las respuestas a las preguntas del cuestionario son cerradas y no dejan que el usuario se exprese como a él le gustaría pudiendo llegar a no estar de acuerdo con la respuesta ofrecida por la prueba. [\[25\]](#)

Tras esta información se descartó la escala TSO por su problema a la hora de generar respuestas personalizadas por el usuario y se pasó a comparar la escala de soledad de UCLA y la escala Este II:



NOMBRE	NÚMERO DE PREGUNTAS	ALTERNATIVAS RESPUESTAS	PUNTUACIÓN
<b>Escala Este II de Soledad Social</b>	<b>15</b>	<b>3</b>	<b>BAJO = 0 a 10 puntos</b>
1. ¿Vd. tiene a alguien con quien puede hablar de sus problemas cotidianos?		SIEMPRE = 0	MEDIO = 11 a 20 puntos
2. ¿Cree que hay personas que se preocupan por usted?		AVECES = 1	ALTO = 21 a 30 puntos
3. ¿Tiene amigos o familiares cuando le hacen falta?		NUNCA = 2	
4. ¿Siente que no le hacen caso?			
5. ¿Se siente triste?			
6. ¿Se siente usted solo?			
7. ¿Y por la noche, se siente solo?			
8. ¿Se siente querido?			
9. ¿Utiliza Ud. el teléfono móvil?			
10. ¿Utiliza Ud. el ordenador (consola, juegos de la memoria)?			
11. ¿Utiliza Ud. Internet?			
12. Durante la semana y los fines de semana ¿le llaman otras personas para salir a la calle?			
13. ¿Le resulta fácil hacer amigos?			
14. ¿Va a algún parque, asociación, hogar del pensionista donde se relacione con otros mayores?			
15. ¿Le gusta participar en las actividades de ocio que se organizan en su barrio/pueblo?			
<b>Escala de la soledad de UCLA</b>	<b>10</b>	<b>4</b>	<b>GRADO BAJO DE SOLEDAD = 20-34 puntos</b>
1. Con qué frecuencia se siente infeliz haciendo tantas cosas solo		ME SIENTO ASÍ A MENUDO = 1	GRADO MODERADO DE SOLEDAD = 35-49 puntos
2. Con qué frecuencia siente que no tiene a nadie con quien hablar		ME SIENTO ASÍ CON FRECUENCIA = 2	GRADO MODERADAMENTE DE SOLEDAD = 50-64 puntos
3. Con qué frecuencia siente que no puede tolerar sentirse solo		RARAMENTE ME SIENTO ASÍ = 3	GRADO ALTO DE SOLEDAD = 65-80 puntos
4. Con qué frecuencia siente que nadie le entiende		NUNCA ME SIENTO DE ESE MODO = 4	
5. Con qué frecuencia se encuentra a sí mismo esperando que alguien le llame o le escriba			
6. Con qué frecuencia se siente completamente solo			
7. Con qué frecuencia se siente incapaz de llegar a los que le rodean y comunicarse con ellos			
8. Con qué frecuencia se siente hambriento de compañía			
9. Con qué frecuencia siente que es difícil para usted hacer amigos			
10. Con qué frecuencia se siente silenciado y excluido por los demás			

Figura 4.1 : Comparación entre escala Este II y la Escala de la UCLA

Analizando los ítems de ambas escalas, salta a la vista el porqué de las críticas a la Escala UCLA si la comparamos en la Escala Este II, esta no profundiza en aspectos como el apoyo o participación social de los usuarios.

Además, la Escala Este II la realizada y validada en España, por lo que está adaptada a nuestro entorno, nuestra cultura, y a los diferentes niveles sociales. Permite detectar si se padece o no soledad y conocer el grado de ésta, en tres niveles: bajo, medio y alto, en función de la puntuación obtenida.

Diferencia tres factores: el primero, más social; el segundo el uso de las nuevas tecnologías, y, el tercero, el sentimiento de soledad no deseada. Otra de las razones principales es su actual introducción de su uso en nuevas tecnologías, teléfonos móviles, ordenador e internet, que son los medios de comunicación más empleados por la gente hoy en día.

### 4.3. Diseño base del Agente

Antes de ponernos a construir nuestro agente es fundamental saber qué queremos que haga, como se comporte en cada situación o el flujo que queremos que siga. Para poder implementar de la mejor forma y natural posible nuestra escala de soledad, decidí poner los siguientes puntos en el diseño:

- Mantener el orden de las preguntas tal cual como se describen para así conseguir mantener separadas las distintas temáticas de los grupos de preguntas y sea más cómodo de responder para el usuario.
- Permitirle al usuario realizar respuestas personalizadas a las preguntas sin mantenerse

cerrado a las respuestas que acepta el test, “Siempre”, “A veces”, “Nunca”.

- Añadir una cuarta opción de respuesta, “Paso”, que permita al usuario no responder preguntas en caso de que no quiera o no se sienta cómodo, aunque eso sí, afectando al resultado final del test por lo que no será lo óptimo.

El flujo de conversación que se plantea entonces es el siguiente:

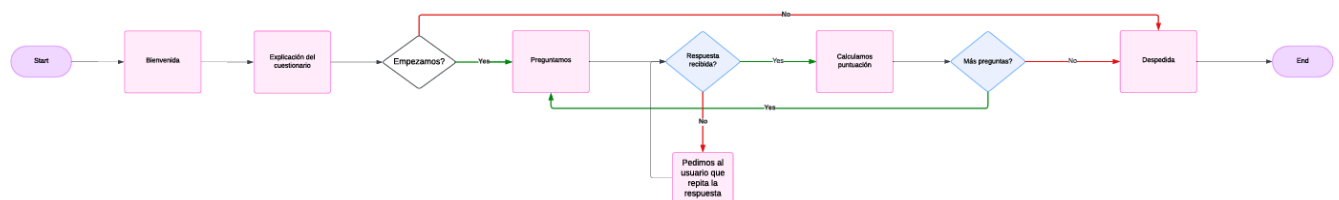


Figura 4.2 : Esquema del flujo de conversación diseñado

1. El agente empezará dándonos la bienvenida y ofreciéndonos una pequeña explicación sobre lo que trata la prueba.
2. A continuación, nos preguntará si deseamos empezar con el cuestionario, en caso afirmativo empezará con el test y en caso negativo, se despedirá de nosotros a la espera de que le indiquemos que queremos comenzar.
3. Al iniciar el cuestionario procederá a realizar las preguntas de la escala y a recopilar la información que le proporciona el usuario. En caso de que reciba una entrada y no sea capaz de entenderla pedirá que se vuelva a proporcionar.
4. Al completar todas las preguntas el agente se despedirá no sin antes proporcionarnos la puntuación final a la que hemos llegado
5. El usuario podrá en cualquier momento pasar una pregunta lo cual no afectará en nada al funcionamiento ya que será tratada como una respuesta normal y no afectará al cálculo de la puntuación, aunque podrá ser detectada como veremos más adelante.

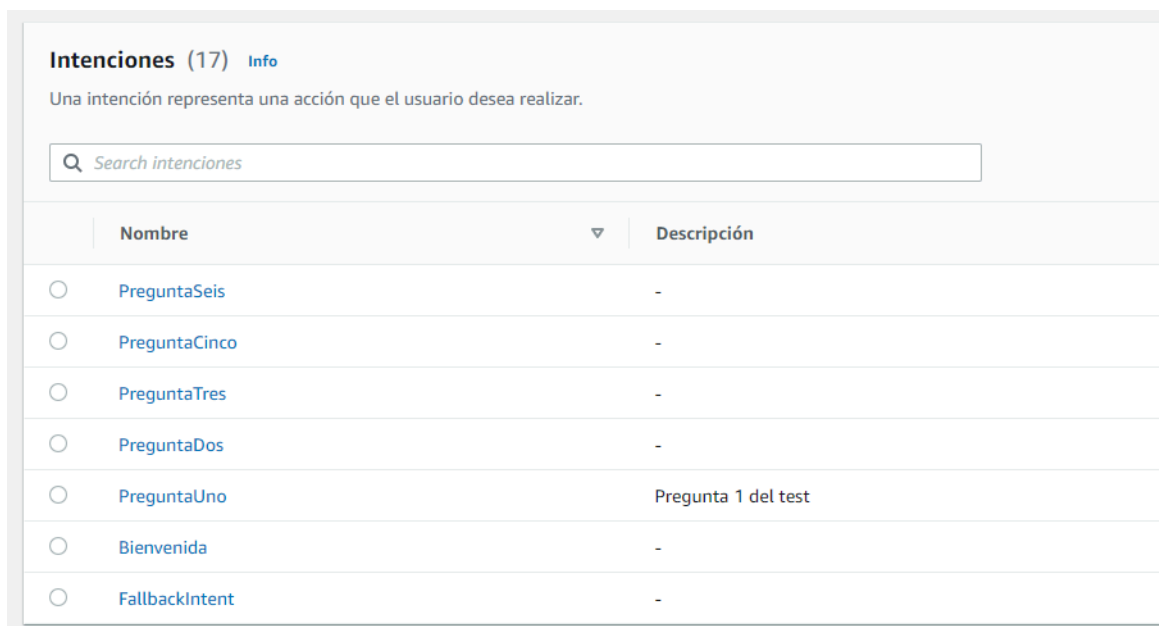
#### 4.4. Implementación del Agente

Tras tener el diseño definido y la escala lista empezaremos con la implementación del agente. Lo primero que haremos será realizar la implementación base del agente en Amazon Lex, posteriormente utilizaremos AWS lambda para gestionar el flujo y eventos de nuestro chatbot en Lex, generaremos y entrenaremos al bot con nuestras frases de entrenamiento para que sea capaz de aceptar y diferenciarlas y finalmente integraremos el agente tanto en Alexa como en las plataformas de Slack y Facebook Messenger.

#### 4.4.1. Diseño del Agente en Amazon Lex

Como ya expliqué en la sección de Amazon Lex, la herramienta cuenta con una serie de elementos fundamentales con los que vamos a construir el bot. Primero, tendremos que definir los intents, los cuales identificarán las intenciones del usuario a lo largo de la conversación.

- **Intent de Bienvenida:** este intent será el encargado de iniciar la conversación con el usuario, le ofrecerá una pequeña explicación sobre en qué consiste la prueba que se va a realizar y le pregunta al usuario si desea iniciar con esta.
- **Intents de Preguntas:** se construyó un intent específico para cada una de las preguntas de la escala con el fin de tener todo mejor organizado y separado a la hora de, más adelante, trabajar con estos intents en la función lambda ya que emplearemos los nombres de los intents como índices a la hora de manejar el flujo de la conversación.
- **Default Fallback Intent:** este intent existe por defecto en el agente y es el encargado de informar al usuario cuando se recibe una entrada que no entiende.



	Nombre	Descripción
<input type="radio"/>	PreguntaSeis	-
<input type="radio"/>	PreguntaCinco	-
<input type="radio"/>	PreguntaTres	-
<input type="radio"/>	PreguntaDos	-
<input checked="" type="radio"/>	PreguntaUno	Pregunta 1 del test
<input type="radio"/>	Bienvenida	-
<input type="radio"/>	FallbackIntent	-

Figura 4.3 : Intents del agente en Amazon Lex

Después de crear los intents tendremos que ir añadiendo a cada uno de estos slots que serán los encargados de pedirle al usuario la información necesaria para que se completen. El slot del intent “Bienvenida” será el encargado de preguntarle al usuario si desea empezar con la encuesta. A partir de aquí los slots de los intents de “Preguntas ”estarán formados por cada una de las preguntas de nuestra escala.

Las respuestas a los slots se guardan en una variable llamada slot type, por defecto lex sólo

reconoce respuestas que ya están predefinidas de base por lo que tendremos que crear nuestro slot type personalizados ajustados a nuestros requisitos. Para ello crearemos uno que entienda y recopile la información del slot del intent “Bienvenida”, que era el encargado de preguntar si se quería empezar con la encuesta, y otro que será el encargado de recopilar las respuestas a las preguntas del test. Estos slots types serán los que entrenaremos para que puedan recoger de forma efectiva las respuestas de los usuarios.

The screenshot shows the AWS Lex console interface for configuring a slot. At the top, there's a section titled "Ranuras (1) - opcional" with an "Info" link and a description: "Información que un bot necesita para cumplir la intención. El bot solicita las ranuras necesarias para el cumplimiento de intenciones, en el siguiente orden de prioridad." A button "Agregar ranura" is in the top right. Below this is a search bar labeled "Filter". The main content area shows a slot configuration for "Solicitar ranura: Respuesta1". It includes a message preview: "Mensaje: ¿Usted tiene a alguien con quien puede habla...". The "Tipo de ranura" is set to "Respuestas". A checkbox "Requerido para esta intención" is checked, with a note: "El bot solicitará esta ranura durante la conversación si el usuario no proporciona un valor." Below this, the "Nombre" is "Respuesta1" and the "Tipo de ranura" is "Respuestas". A text box for "Solicitudes" contains the question: "¿Usted tiene a alguien con quien puede hablar de sus problemas cotidianos?". At the bottom, there's a note about advanced options and a button "Opciones avanzadas".

Figura 4.4 : Ejemplo de un Slot de un Intent de Pregunta

Ahora que ya tenemos construida la base de nuestro chatbot tendremos que marcar en cada intent y en cada slot la opción de “lambda function” gracias a esto cada vez que se necesite un slot se hará una llama a la función lambda que crearemos a continuación y que será la encargada de gestionar el flujo entre los intents, comprobará que las respuestas son aceptadas por el agente y calculará los puntajes además de que cada vez que se llame a esta función, toda la información generada por esta, tanto las llamadas que se realicen como de las variables que queramos, podrán ser vistas desde los “CloudWatch Logs”.

▶	2025-01-27T10:36:28.730+01:00	Procesando intent: Bienvenida
▶	2025-01-27T10:36:28.730+01:00	Puntaje acumulado antes de procesar el intent: 0
▶	2025-01-27T10:36:28.730+01:00	Reiniciando el puntaje al inicio de la encuesta.
▶	2025-01-27T10:36:28.730+01:00	Validación del slot 'IniciarEncuesta' completada con éxito.
▶	2025-01-27T10:36:28.730+01:00	Respuesta del usuario para 'IniciarEncuesta': si
▶	2025-01-27T10:36:28.730+01:00	Usuario desea iniciar la encuesta. Pasando a 'PreguntaUno'.
▶	2025-01-27T10:36:28.731+01:00	END RequestId: c1408b42-5099-42cc-b1ce-fac78accfd31
▶	2025-01-27T10:36:28.731+01:00	REPORT RequestId: c1408b42-5099-42cc-b1ce-fac78accfd31 Duration: 1.58 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 31 MB
▶	2025-01-27T10:36:36.942+01:00	START RequestId: fcc53a1a-5c8f-4a91-8f03-fa80ea687cf2 Version: \$LATEST
▶	2025-01-27T10:36:36.942+01:00	Procesando intent: PreguntaUno
▶	2025-01-27T10:36:36.942+01:00	Puntaje acumulado antes de procesar el intent: 0
▶	2025-01-27T10:36:36.942+01:00	Procesando la pregunta número 1.
▶	2025-01-27T10:36:36.942+01:00	Respuesta válida: Siempre. Puntaje actualizado: 0
▶	2025-01-27T10:36:36.942+01:00	Pasando a la siguiente pregunta: PreguntaDos.

Figura 4.5 : Registro de Logs de una ejecución del agente

#### 4.4.2. Función Lambda

AWS Lambda es la parte más importante del funcionamiento del chatbot en Amazon Lex, gracias a esta vamos a ser capaces de controlar completamente el flujo y comportamiento de este además de poder imprimir información relevante de las entradas del usuario, como por ejemplo poder ver cuando se decide pasar una pregunta sin ser contestada o el rango de tiempo entre los que el chatbot estuvo trabajando. [\[26\]](#)

Lo primero en nuestra función será crear una función que nos valide el valor de los slots para asegurarnos que ninguno contenga ningún valor erróneo o nulo. Esto lo haremos viendo la variable “violatedSlot: intent” la cual nos permitirá comprobar el contenido del slot del intent que nosotros queramos. Teniendo esto pasaremos a crear una función que nos permita al bot moverse entre los intent. Para ello tras comprobar que la entrada a la pregunta sobre iniciar la encuesta es positiva, haremos un bucle empleando el nombre de los intents de las preguntas, Pregunta1, Pregunta2, etc., como índices para formar un bucle que nos haga recorrer las preguntas hasta que el usuario termine de responder a las preguntas.

Para calcular la puntuación, primero asociaremos valores a las tres posibles respuestas del test tal cual aparece indicado en la escala, y un valor nulo para cuando el usuario no quiera responder una pregunta para así distinguirlo del valor cero lo que hará que podemos detectar en los logs cuando una pregunta no fue respondida por el usuario. Tras esto crearemos una variable global e iremos sumando los valores dentro del bucle donde la función irá asociando cada una de las respuestas al valor correspondiente y sumándolo. En caso de recibir una entrada que el chatbot no entienda, pedirá al usuario repetir esa respuesta recomendando que en caso de no entender que intente hacerlo con una de las predefinidas por la escala para así poder avanzar con la prueba.

Cuando se llegue a la última pregunta se terminará el bucle y el agente dará las gracias al usuario para a continuación proporcionarle la puntuación final a la que se llegó.

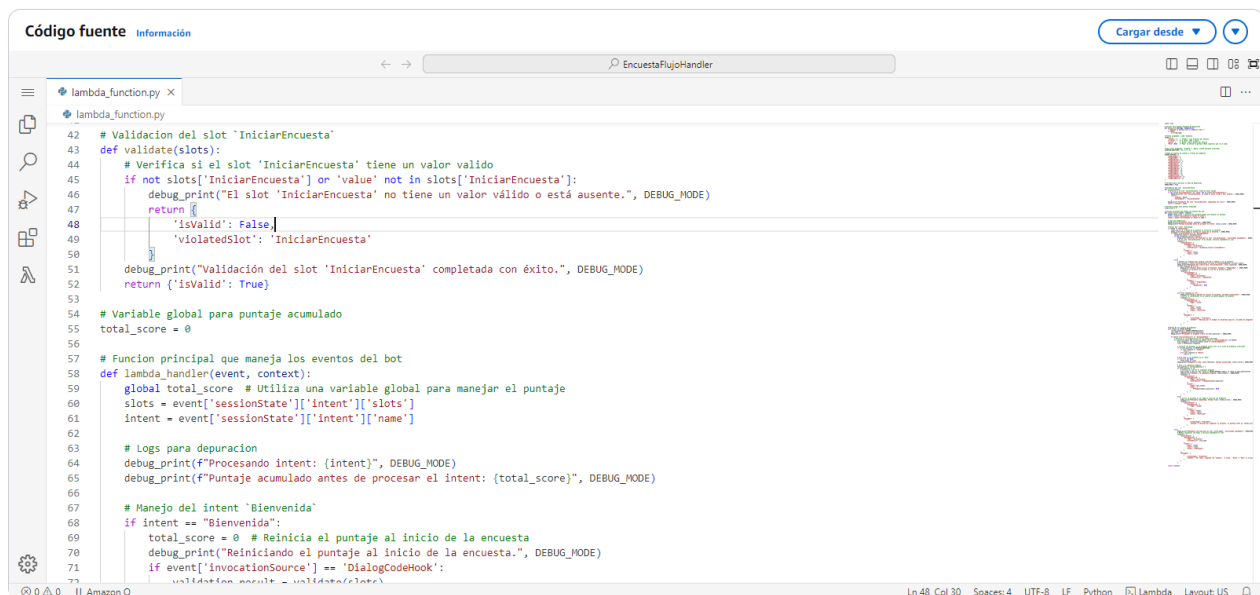


Figura 4.6 : Función Lambda para el control del flujo

#### 4.4.3. Creación de las Frases de Entrenamiento

Las frases de entrenamiento son muy importantes para entrenar el agente para que este pueda entender y clasificar las intenciones del usuario. Debido a esto es necesaria una buena cantidad de frases para que sea capaz de identificar las respuestas a las preguntas del test.

Debido a que la escala sólo tenía tres respuestas posibles se intentó generar de forma manual una batería de frases de entrenamiento. Tras una serie de pruebas comprobé que era insuficiente y los resultados a los que llegaba no eran óptimos. Algo importante a la hora de generar estas frases es crear frases de entrenamiento que sean contrarias, pero manteniendo estructuras similares para que así el agente pueda clasificarlas de mejor forma.

A partir de aquí decidí generar frases de forma artificial para ayudar así a crear una mayor cantidad y mejorar la clasificación de las entradas. Para esto se empleó los modelos vistos previamente en la sección 3.3 para que generasen respuestas básicas que se adaptasen a las preguntas, para ello seguí los siguientes pasos para asegurarse de recibir unas respuestas adaptadas a las necesidades del agente, primero les ofrecí ejemplos de las frases de entrenamiento que había creado manualmente y después les fui proporcionando cada una de las preguntas ofreciendo contexto específico para que las respuestas estuviesen lo más adaptadas posibles.



Figura 4.7 : Ejemplos de frases de entrenamiento utilizadas

Tras conseguir una batería de frases de entrenamiento, se trasladan a Amazon Lex y se entrena el agente de forma manual para así intentar conseguir un mayor nivel de presión a la hora de clasificar y entender las respuestas.

#### 4.4.4. Implementación en aplicaciones

Tras haber finalizado con el desarrollo de nuestro agente conversacional, me dispuse a realizar su implementación en varias aplicaciones. Esta implementación tenía como objetivo que cualquier usuario pudiese tener acceso al agente en cualquier dispositivo tanto un ordenador como en un dispositivo móvil además de aumentar considerablemente la comodidad de su uso.

Para ello me dispuse a intentar implementarlo tanto en los servicios de chat de Facebook Messenger y Slack, como creando una skill para emplear en dispositivos de Alexa.

Facebook Messenger es una aplicación de mensajería desarrollada por Meta que permite realizar conversaciones entre usuarios a través de mensajes de texto permitiendo la integración de bots mediante su API. Además, cuenta con una amplia compatibilidad multiplataforma lo que nos ayuda a expandir el agente en muchos dispositivos. [\[27\]](#)

En cuanto a Slack, es una plataforma de comunicación diseñada principalmente para entornos laborales en la que destaca su versatilidad a la hora de realizar integraciones de herramientas externas. Además, es capaz de guardar un historial de las conversaciones lo que nos permitirá

revisar de otro modo cualquier prueba realizada y tiene también al igual que Messenger una alta compatibilidad multiplataforma. [28]



Figura 4.8 : Conversación del agente mediante la plataforma de Slack

Para integrar nuestro agente en ambos dispositivos será necesario crear un canal a través de Lex que nos permita realizar la comunicación entre las herramientas.

Integraciones de canales (2) <a href="#">Info</a>						<a href="#">Eliminar</a>	<a href="#">Copiar URL</a>	<a href="#">Agregar canal</a>
<input type="text" value="Search canales"/>						< 1 > ⚙		
	Nombre del canal	Plataforma	Alias	Idioma	Creado			
<input type="radio"/>	ChatbotSoledad	Slack	TestBotAlias	Spanish (ES)	1 mes hace			
<input type="radio"/>	ChatbotSoledad	Facebook	TestBotAlias	Spanish (ES)	1 mes hace			

Figura 4.9 : Canales para implementación en Messenger o Slack

Para la integración con Messenger tendremos que acceder al portal de desarrolladores y crear una nueva aplicación con una página de Facebook asociada. Tras haber creado la base tendremos que integrar la comunicación entre Lex y nuestra aplicación en Messenger, iremos a nuestro canal creado en Lex e ingresaremos los tokens que se nos generó Messenger, al activar la integración se nos devolverá una URL de devolución de llamadas que deberemos integrar posteriormente en nuestra aplicación para que así se realice la comunicación entre ambos softwares. La integración mediante Slack es similar a la anterior, pero desarrollando todo esto en el portal de desarrolladores de Slack. Gracias a esto habremos sido capaces de integrar nuestro agente tanto en una aplicación de Facebook con una página asociada que pueda visitar cualquier usuario, como en una aplicación de Slack que puede ser sencillamente integrada en cualquier grupo o canal que deseemos.



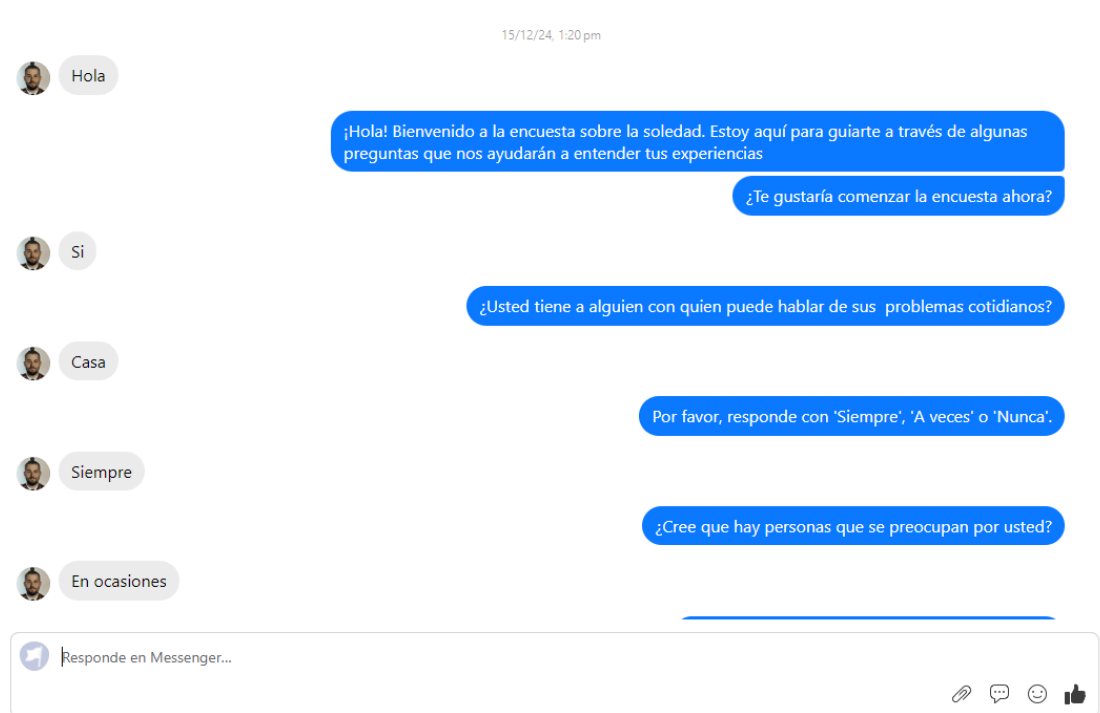


Figura 4.10 : Conversación del agente mediante la plataforma de Messenger

A la hora de realizar la integración en una skill de Alexa me encontré con muchos problemas a la hora de migrar el agente. Construí toda la base del agente sobre la plataforma de Alexa Skill con el fin de integrar mi bot en Lex y mi función Lambda, pero me encontré una serie de problemas a la hora de realizar la comunicación entre estos. Además, no es una práctica al parecer muy común, aunque ambas herramientas formen parte del ecosistema de Amazon ya que no conseguí encontrar ningún ejemplo ni explicación del desarrollo por ningún lado. [\[29\]](#)

## 5. PRUEBAS

A partir de aquí se detallarán las pruebas que se realizaron sobre el agente y los resultados obtenidos.

### 5.1. Plan de Pruebas

Para estudiar el comportamiento y funcionamiento del agente, se realizaron una serie de pruebas:

- Primero se realizaron pruebas sobre el chatbot en Lex, para ello se realizaron continuas ejecuciones, cambiando continuamente las entradas que recibía en cada una de las preguntas que conforman la encuesta, intentando hacer salir algún tipo de error a la hora de seleccionar las puntuaciones de las respuestas o en el flujo de la conversación.

- Se realizaron una serie de pruebas con usuarios reales con las que se pudo evaluar la fiabilidad y precisión del agente desarrollado y con el fin de obtener feedback para realizar posibles mejoras.

Se realizaron pruebas con 5 usuarios de entre 30 y 55 años habiendo tanto hombres como mujeres entre ellos. Para realizar las pruebas se siguieron los siguientes pasos:

- Se hizo una breve explicación sobre en qué consistía el proyecto, la finalidad de estas pruebas incitando en todo momento a que opinaran sobre cualquier aspecto a mejorar.
- Antes de comenzar con las pruebas les informé a los usuarios que no debían ser sus verdaderas respuestas pero que intentasen seguir una misma línea. En caso de que el chatbot no les entendiese les comenté que intentasen cambiar la estructura de la frase o que se ciñeran a las respuestas base que tiene la escala.
- Cada vez que se terminaba una prueba, realicé una comprobación en los logs de la función lambda para comprobar que fuesen detectadas correctamente las entradas que había realizado el usuario previamente.
- Para cada prueba se cronometró el tiempo que se tardó en completar la encuesta desde que se activó el agente hasta que terminó con la encuesta.

El objetivo de todas estas pruebas fue:

- Evaluar la calidad del agente en Lex a la hora de realizar la Escala Este II.
- Evaluar también el comportamiento del agente desarrollado con usuarios reales para verificar y revisar su usabilidad, revisar áreas en las que se puedan realizar mejoras y recabar opiniones de los usuarios.

## 5.2. Resultados de las Pruebas en Amazon Lex

Las pruebas realizadas en Lex a través de texto estuvieron formadas por distintas pruebas sobre cada una de las respuestas posibles para comprobar la capacidad del bot para entender y clasificar las respuestas de entrada. Para ello, se aplicó el framework PARADISE para calcular el coeficiente kappa del agente, mediante 30 pruebas divididas en 10 respuestas para cada una de las tres posibles del test, "Siempre", "A veces" y "Nunca". Con los resultados de las pruebas se generó una matriz de confusión a partir de la cual se calculó el **coeficiente kappa** obteniendo un valor de **k = 0.9**.

		P1			P2			P3			P4		
	DATA	SIEMPRE	A VECES	NUNCA	SIEMPRE	A VECES	NUNCA	SIEMPRE	A VECES	NUNCA	SIEMPRE	A VECES	NUNCA
P1	SIEMPRE	8	1	2									
	A VECES	2	9										
	NUNCA			8									
P2	SIEMPRE				9	2							
	A VECES				1	7	1						
	NUNCA					1	9						
P3	SIEMPRE							8		1			
	A VECES							1	9				
	NUNCA							1	1	9			
P4	SIEMPRE										10	1	1
	A VECES											8	
	NUNCA											1	9

Figura 5.1 : Matriz de confusión de las pruebas de texto en Lex

### 5.3. Resultado Pruebas con Usuarios

Las pruebas con usuarios reales se realizaron en dos plataformas distintas, en Lex y en Slack, para investigar cuál de las dos generaba una mayor tasa de satisfacción. Durante el desarrollo de las pruebas se tomó el tiempo de conversación con el agente de los usuarios para conseguir un valor de tiempo medio para completar la escala, este valor fue de **9 minutos y 32 segundos** de media para las ejecuciones en Lex y de **6 minutos y 10 segundos** para las pruebas realizadas en la plataforma de Slack. La diferencia entre estos valores es que al realizar las pruebas con la plataforma de Lex se empleó en todo momento la conversación mediante voz y en cambio en Slack se empleó el método escrito.

Tiempo de pruebas en Lex

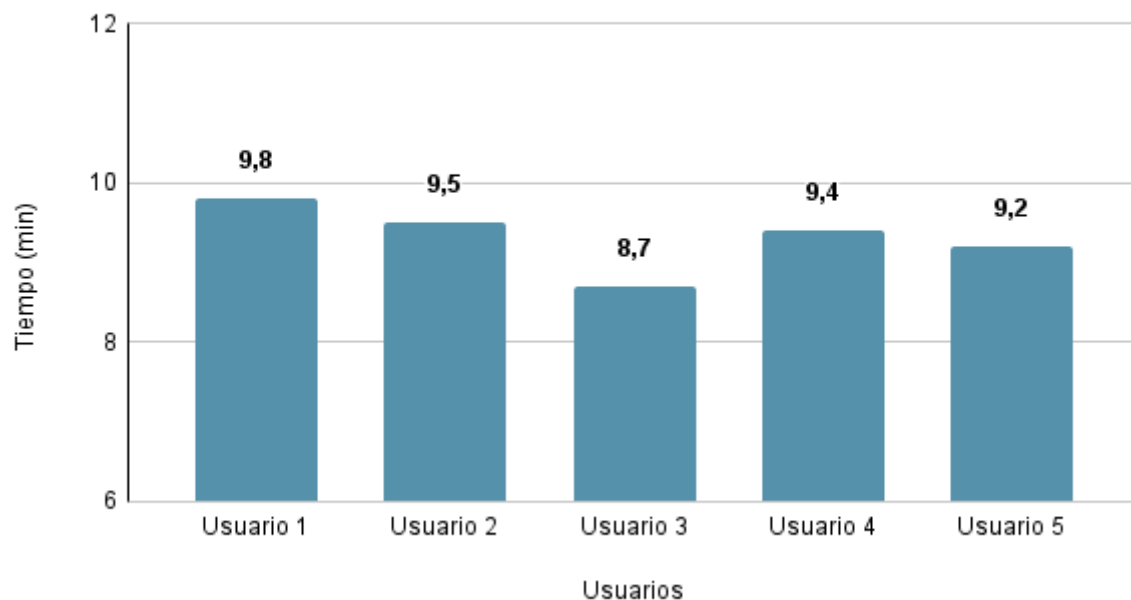


Figura 5.2 : Gráfica de tiempos en Lex

## Tiempo de pruebas en Slack

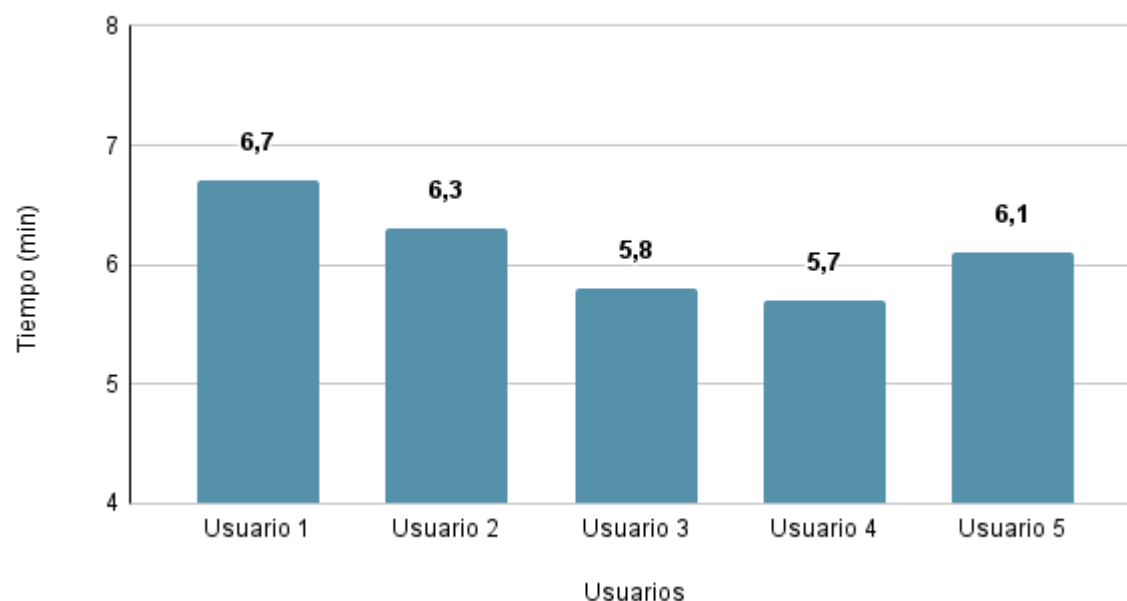


Figura 5.3 : Gráfica de tiempos en Slack

A la hora de comprobar la funcionalidad del agente, se comprobó que se generó algún error en las pruebas realizadas en Lex, debido en gran parte a fallos en la transcripción de audio a texto que tiene la herramienta. Igualmente los resultados obtenidos a través de las pruebas en Slack a través de texto generaron buenos resultados, el agente fue capaz de clasificar correctamente las respuestas de los usuarios, no siempre entendiendo la primera entrada, pero la capacidad de pedir una nueva respuesta al usuario genera robustez en el agente.

PREGUNTAS	USUARIO 1	USUARIO 2	USUARIO 3	USUARIO 4	USUARIO 5
EDAD	32	30	34	55	52
SEXO	HOMBRE	MUJER	HOMBRE	HOMBRE	MUJER
1. ¿Vd. tiene a alguien con quien puede hablar de sus problemas cotidianos?	SIEMPRE	SIEMPRE	SIEMPRE	SIEMPRE	A VECES
2. ¿Cree que hay personas que se preocupan por usted?	SIEMPRE	SIEMPRE	SIEMPRE	SIEMPRE	SIEMPRE
3. ¿Tiene amigos o familiares cuando le hacen falta?	SIEMPRE	SIEMPRE	SIEMPRE	SIEMPRE	SIEMPRE
4. ¿Siente que no le hacen caso?	A VECES	NUNCA	NUNCA	NUNCA	A VECES
5. ¿Se siente triste?	NUNCA	A VECES	NUNCA	NUNCA	NUNCA

6. ¿Se siente usted solo?	NUNCA	NUNCA	NUNCA	NUNCA	NUNCA
7. ¿Y por la noche, se siente solo?	NUNCA	NUNCA	A VECES	NUNCA	NUNCA
8. ¿Se siente querido?	SIEMPRE	SIEMPRE	SIEMPRE	SIEMPRE	SIEMPRE
9. ¿Utiliza Ud. el teléfono móvil?	SIEMPRE	SIEMPRE	SIEMPRE	SIEMPRE	SIEMPRE
10. ¿Utiliza Ud. el ordenador (consola, juegos de la memoria)?	SIEMPRE	A VECES	SIEMPRE	NUNCA	NUNCA
11. ¿Utiliza Ud. Internet?	SIEMPRE	SIEMPRE	SIEMPRE	SIEMPRE	SIEMPRE
12. Durante la semana y los fines de semana ¿le llaman otras personas para salir a la calle?	SIEMPRE	SIEMPRE	SIEMPRE	A VECES	A VECES
13. ¿Le resulta fácil hacer amigos?	SIEMPRE	SIEMPRE	SIEMPRE	SIEMPRE	SIEMPRE
14. ¿Va a algún parque, asociación, hogar del pensionista donde se relacione con otros mayores?	NUNCA	NUNCA	NUNCA	NUNCA	NUNCA
15. ¿Le gusta participar en las actividades de ocio que se organizan en su barrio/pueblo?	SIEMPRE	SIEMPRE	SIEMPRE	A VECES	A VECES
<b>PUNTUACIONES</b>	<b>3</b>	<b>4</b>	<b>3</b>	<b>6</b>	<b>8</b>

Tabla 5.1 : Resultado pruebas en usuarios

Al terminar cada una de las pruebas se les fue pasando el cuestionario de satisfacción que comenté en el apartado 3.6 y estos fueron los resultados obtenidos:

<b>PREGUNTAS</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>ÍNDICE DE SATISFACCIÓN</b>	<b>MUY BAJO</b>	<b>BAJO</b>	<b>MEDIO</b>	<b>ALTO</b>	<b>MUY ALTO</b>
1. ¿Estás satisfecho/a con la dificultad de utilizar la herramienta?	0%	0%	0%	20%	80%
2. ¿Te sientes cómodo/a compartiendo tus sentimientos con el agente?	0%	0%	0%	60%	40%
3. ¿Te sientes más cómodo/a respondiendo las preguntas por voz que a través de mensajes de texto?	0%	40%	40%	20%	0%

4. ¿Te sientes cómodo/a empleando el asistente en una plataforma como Slack?	0%	0%	0%	20%	<b>80%</b>
5. ¿Consideras que puede ser una herramienta útil?	0%	0%	20%	<b>60%</b>	20%
6. ¿Recomendarías el uso de la herramienta?	0%	0%	20%	<b>40%</b>	<b>40%</b>

Tabla 5.2 : Resultado cuestionario de evaluación de los usuarios

## 6. ANÁLISIS DE RESULTADOS

Los resultados obtenidos muestran que es capaz de aplicar de forma consistente la escala de soledad social Este II en los usuarios, siendo capaz de clasificar las respuestas de los usuarios a lo largo de las 15 preguntas del cuestionario de forma notablemente satisfactoria. Además el agente tiene la capacidad de pedir al usuario una nueva respuesta en caso de no haberla entendido por lo que se consigue mantener el flujo de la conversación. En esta prueba tras aplicar Paradise, se obtuvo un **coeficiente Kappa k = 0.9**, valor que al estar cerca de 1, indica que el agente es capaz de clasificar las respuestas a las preguntas de forma bastante satisfactoria.

En pruebas realizadas a los usuarios se pudo comprobar en mayor medida el funcionamiento de este desde pudiendo comprobar que es capaz de recibir y clasificar las respuestas generadas por los usuarios en buena medida obteniendo mejores resultados a través de texto en Slack que a partir de audio Lex lo cuál producía errores en la transcripción de las entradas de los usuarios que terminaba por afectar de igual forma a la clasificación de las respuestas.

Si nos fijamos en las gráficas de tiempos ([Figura 5.2](#), [Figura 5.3](#)), las pruebas realizadas a través de la plataforma de Slack se completaron en tiempos menores si las comparamos con las realizadas en la plataforma de Lex. Estos valores se deben a que por un lado, las primeras pruebas se realizaron en la plataforma de Lex, por lo que los usuarios se iban a someter al test por primera vez. Otro de los factores fue el tipo de medio utilizado para impartir el test y es que, al realizar el test mediante chat de voz en Lex, se generaban tiempos mayores ya que el agente se va comunicando con nosotros a partir de una voz artificial.

Los resultados de las pruebas ([Tabla 5.1](#)) muestran valores de soledad social muy bajos ([Tabla 3.2](#), menos de 10 puntos), hay que tener en cuenta que los usuarios que se sometieron a las pruebas del agente, tienen edades por debajo de los 55 años por lo que estamos en un rango de edad relativamente bajo si tenemos en cuenta objetivo final al que está destinado el agente. Aun teniendo esto en cuenta, podemos ver un ligero aumento en la puntuación de los usuarios de más edad que se sometieron al test, esto puede ser debido a la disminución de encuentros sociales o a problemas con la adaptación al uso de nuevas tecnologías. Si observamos los resultados en base al género de los usuarios, se puede ver un ligero aumento en la puntuación de los usuarios mujer frente a los hombres pero considero que no es un cambio lo suficientemente relevante como para

poder hablar de que sea un factor determinante, para investigar esto de una forma más exacta se tendría que volver a realizar las pruebas aumentando el número de usuarios para tener una mayor muestra de resultados.

Los resultados del cuestionario de evaluación del agente conversacional ([Tabla 5.2](#)) nos revelan una alta satisfacción en los usuarios. La mayoría de los usuarios consideran que la herramienta es fácil de usar además de sentirse bastante cómodos a la hora de expresar sus sentimientos al agente. Pasando a las preguntas sobre la herramienta usada para realizar las pruebas, pudiendo ser Amazon Lex o Slack, se pudo comprobar que los usuarios no se sienten especialmente cómodos realizando las pruebas a través de voz, y que una plataforma como Slack, que en cierto modo es parecida a *Whatsapp* y la cual es una plataforma empleada globalmente por muchos usuarios hoy en día, sería una opción más amigable y cómoda para los usuarios. En cuanto a las últimas dos preguntas del cuestionario nos revelan que los usuarios la consideran una herramienta que puede llegar a ser útil para la sociedad y que además la recomendarían a otras personas.

## 7. CONCLUSIONES

Este proyecto se basó en el estudio y desarrollo del uso de agentes conversacionales como herramienta de apoyo para detectar casos de soledad social en la población. Más en concreto este proyecto intentó administrar la **Escala Este II** de soledad social mediante un agente conversacional tratando de que este fuese capaz de entender y realizar de manera fluida la conversación con el usuario.

Además del desarrollo del del agente se implementó en varios servicios de mensajería como lo son Facebook Messenger y Slack, además se intentó crear una skill que sirviese para exportarlo en dispositivos Alexa aunque finalmente no se llegó a conseguir.

Este proyecto consiguió demostrar experimentalmente la viabilidad del uso de agentes conversacionales para la administración de escalas demostrando su capacidad para entender las respuestas de los usuarios. Este prototipo tiene el potencial para ser una alternativa tecnológica válida para evaluar la soledad de forma menos invasiva y más intuitiva, por voz. Pero la evaluación siempre será hecha por un profesional a partir de esta prueba, y aquellas que estime para evaluar el nivel de soledad y/o depresión de una persona mayor.. Las pruebas realizadas en usuarios reales demostraron además la aceptación actual por este tipo de herramientas y su impacto social, además de destacar su facilidad de uso a la hora de realizar el test.

Pasando con las limitaciones que me encontré a lo largo del desarrollo del proyecto, destacaría principalmente la curva de aprendizaje del entorno de AWS, este cuenta con muchas herramientas que trabajan entre sí, lo que genera ciertas dificultades para hacerse con dichas herramientas y supone un reto el llevarlo a cabo en el tiempo de desarrollo estimado para el TFG.. Otra de las limitaciones encontradas es el reconocimiento de voz propio de la herramienta de Lex el cual a la hora de transcribir los mensajes de voz suele cometer errores lo que afecta al correcto funcionamiento del agente.

Las líneas futuras a seguir para continuar con el desarrollo del proyecto serían:

- Realizar un mejor y más amplio entrenamiento al agente para mejorar los resultados actuales y conseguir hacerlo más robusto y eficiente.
- Implementar el agente en una skill y con ello poder utilizarlo en dispositivos Alexa y así permitir una mayor comodidad a los usuarios además de añadir la capacidad realizar enteramente la conversación por voz.
- Uso del agente en un piloto para realizar pruebas en la Cruz Roja de Lugo lo cual otorgaría una muestra de usuarios más representativa en cuestiones de edad y cantidad de usuarios y nos podrían ofrecer resultados más amplios.



- Replicar el proyecto con otro tipo de escala a administrar con el fin de abordar otras áreas asociadas a la salud mental.

# APÉNDICE

En este apéndice se desarrolla la información necesaria para poder probar el agente desarrollado a lo largo del proyecto de forma efectiva.

Lo primero que tendremos que hacer es crearnos una cuenta en el entorno de AWS para ser capaces de utilizar la herramienta de Amazon Lex y AWS Lambda. Después de crear la cuenta, iremos al siguiente repositorio de GitHub : [https://github.com/Bruniloox/TFG\\_ChatBot.git](https://github.com/Bruniloox/TFG_ChatBot.git)

Dentro del GitHub nos interesan los dos archivos.zip, el archivo **Chatbot.zip** contiene toda la estructura del agente desarrollado, y el archivo **lambda\_function.zip**, contiene la función lambda que gestiona el flujo de la conversación y eventos del agente.

Para importar el agente nos iremos a Lex, después nos iremos al **apartado de Bots**, iremos al botón de **Acción** y seleccionamos **Importar**:



Figura Apéndice 1: Bot en Lex

El siguiente paso será ponerle un nombre al bot, importar nuestro archivo, **Chatbot.zip**, y seleccionar un rol con permisos básicos de AWS.

The screenshot shows the 'Importar bot' form in the AWS Lex console. It has two main sections. The first section, 'Archivo de entrada', contains a text field for 'Nombre del bot' with the value 'Prueba', a button 'Examinar archivo', and a password field labeled 'Contraseña - opcional'. The second section, 'Permisos de IAM', contains a radio button selection for 'Rol de tiempo de ejecución', with 'Crear un rol con permisos básicos de Amazon Lex' selected. The form includes various informational text blocks and links for help.

Figura Apéndice 2: Importar el bot en Lex

Una vez importado el bot, nos iremos a la **sección de Intents** y buscaremos y entraremos en el **intent “Bienvenida”**. Dentro del intent iremos a la esquina superior derecha y pulsaremos en el botón **Crear** cargar definitivamente el bot.

El siguiente paso será entrar en AWS Lambda y crear una función, añadiremos el nombre que queramos, seleccionaremos el lenguaje de Python 3.13 y crearemos la función.

Seleccione una de las siguientes opciones para crear la función.

☒ Crear desde cero  
Empezar con un sencillo ejemplo "Hello World".

☐ Utilizar un proyecto  
Cree una aplicación Lambda utilizando un código de muestra y los ajustes de configuración predefinidos de casos de uso comunes.

☐ Imagen del contenedor  
Seleccione una imagen de contenedor para implementar para la función.

---

**Información básica**

**Nombre de la función**  
Escriba un nombre para describir el propósito de la función.  
  
El nombre de la función debe tener entre 1 y 64 caracteres, debe ser exclusivo de la región y no puede incluir espacios. Los caracteres válidos son a-z, A-Z, 0-9, guiones (-) y guiones bajos (\_).

**Tiempo de ejecución** Información  
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.  

Python 3.13

**Arquitectura** Información  
Elija la arquitectura del conjunto de instrucciones que desea para el código de la función.  




☒ x86\_64  
☐ arm64

**Permisos** Información  
De forma predeterminada, Lambda creará un rol de ejecución con permisos para cargar registros en Amazon CloudWatch Logs. Puede personalizar este rol predeterminado más adelante al agregar los disparadores.

► Cambiar el rol de ejecución predeterminado

Figura Apéndice 3: Importar el bot en Lex

Entramos ahora en nuestra función Lambda y cargaremos el archivo **lambda\_function.zip**. Una vez que tenemos la función ya cargada volveremos a Amazon Lex al intent “Bienvenida” en donde nos encontrábamos anteriormente y volveremos a pulsar el botón **Crear**. Una vez terminado de cargar todo, pulsaremos el botón **Prueba**, dentro de esta sección seleccionaremos el icono de rueda, y escogeremos nuestra función Lambda y guardaremos la configuración. Con todo esto ya podremos enviarle al agente una de las frases de inicio que veremos en la sección “**Ejemplos de enunciados**” dentro del intent en el que nos encontramos para empezar el test.

**Prueba Versión borrador**   

Última compilación enviada:  
23 minutos hace

**Inspeccionar**

▼ **Función Lambda - opcional**  
La función Lambda se invoca para inicialización, validación y cumplimiento.

**Fuente**

Prueba ▲

Q |

Ninguna

EncuestaFlujoHandler

Prueba

Figura Apéndice 4: Comunicación Lex - Lambda

# BIBLIOGRAFÍA

[1] Consejo General de Psicología de España:

<https://www.infocop.es/1-de-cada-2-personas-mayores-esta-en-riesgo-de-soledad/>

[2] Trabajo de fin de Máster sobre el uso de agentes conversacionales para la gestión de trastornos depresivos:

<https://openaccess.uoc.edu/bitstream/10609/147518/3/mbarotdTFM0123memoria.pdf>

[3] Artículo sobre agentes conversacionales, tipos, usos e información relevante:

<https://blog.tenea.com/agente-conversacional/>

[4] Ministerio de salud: Porcentaje de población de 15 y más años que padece determinados problemas crónicos, registrado en atención primaria:

<https://www.sanidad.gob.es/estadEstudios/sanidadDatos/tablas/tabla5.htm>

[5] Sánchez, W., Martínez, A., Campos, W., Estrada, H., Pelechano, V. (2015). Inferring loneliness levels in older adults from smartphones:

<https://journals.sagepub.com/doi/abs/10.3233/AIS-140297>

[6] Bond, R.R., Zheng, H., Wang, H., Mulvenna, M.D., McAllister, P., Delaney, K., Walsh, P., Keary, A., Riestra, R., Guaylupo, S., Hemmje, M., Becker, J., Engel, F. (2017). SenseCare: Using affective computing to manage and care for the emotional wellbeing of older people:

[https://link.springer.com/chapter/10.1007/978-3-319-49655-9\\_42](https://link.springer.com/chapter/10.1007/978-3-319-49655-9_42)

[7] Balbuena, J., Samamé, H., Almeyda, S., Mendoza, J., Pow-Sang, J.A. (2021). Depression Detection Using Audio-Visual Data and Artificial Intelligence: A Systematic Mapping Study:

[https://link.springer.com/chapter/10.1007/978-981-15-5859-7\\_29](https://link.springer.com/chapter/10.1007/978-981-15-5859-7_29)

[8] Artículo sobre el chatbot de soledad social SERENA:

<https://catedra.cuatroochenta.com/en/puede-la-tecnologia-reducir-la-soledad-y-el-aislamiento-social/>

[9] Proyecto de investigación sobre soledad no deseada en la era digital en el que se trata el uso de SERENA:

[https://www.fundaciogrifols.org/documents/4438882/5016815/Q62\\_soledad\\_no\\_deseada.pdf/db4e6643-a4a4-e1de-53d2-48529ad83c84](https://www.fundaciogrifols.org/documents/4438882/5016815/Q62_soledad_no_deseada.pdf/db4e6643-a4a4-e1de-53d2-48529ad83c84)

[10] Artículo presentado por la Universidad de Vigo en el que se trata el asistente virtual Celia:

<https://vigotecnologico.uvigo.es/gl/atlanttic-presenta-celia-un-asistente-de-conversa-en-whatsapp-para-combater-a-soledade-non-desexada/>

[11] Documentación de Amazon Lex v2:

<https://docs.aws.amazon.com/lexv2/latest/dg/what-is.html>

[12] Plataformas de mensajería, Amazon Lex v2:

<https://docs.aws.amazon.com/lexv2/latest/dg/deploying-messaging-platform.html>

[13] Instrumento de Medición de Soledad Social: ESCALA ESTE II:

<http://envejecimiento.csic.es/documentos/documentos/rubio-soledad-este2.pdf>

[14] Instrumento de Medición de Soledad Social: ESCALA ESTE I:

<http://envejecimiento.csic.es/documentos/documentos/rubio-soledad-02.pdf>

[15] Escala de Soledad de Russell

[https://rua.ua.es/dspace/bitstream/10045/97430/1/RevPsicolSalud\\_6\\_1\\_03.pdf](https://rua.ua.es/dspace/bitstream/10045/97430/1/RevPsicolSalud_6_1_03.pdf)

[16] Escala de Soledad Social y emocional de Jong Gierveld:

<https://www.soledades.es/recursos/escala-de-soledad-de-de-jong-gierveld>

[17] Escala de Evaluación de Soledad Social y Emocional en Adultos SESLA-S de Yárnoz:

<https://www.redalyc.org/pdf/560/56080109.pdf>

[18] ChatGPT OpenAI:

<https://chatgpt.com/?ref=dotcom>

[19] Aria:

<https://www.opera.com/es-419/features/aria>

[20] Gemini Google:

<https://deepmind.google/technologies/gemini/>

[21] Marilyn A. Walker, Diane J. Litman, Candace A. Kamm and Alicia Abella. PARADISE: A Framework for Evaluating Spoken Dialogue Agents

<https://dl.acm.org/doi/pdf/10.3115/976909.979652>

[22] Rasa:

<https://rasa.com/product/rasa-platform/>

[23] Dialogflow:

<https://cloud.google.com/products/conversational-agents?hl=es-419>

[24] Escala de Soledad de la UCLA:

[https://rua.ua.es/dspace/bitstream/10045/97430/1/RevPsicolSalud\\_6\\_1\\_03.pdf](https://rua.ua.es/dspace/bitstream/10045/97430/1/RevPsicolSalud_6_1_03.pdf)

[25] La Escala de Valoración Sociofamiliar TSO:

[https://cendocps.carm.es/documentacion/2020\\_Escala\\_valoracion\\_sociofamiliar.pdf](https://cendocps.carm.es/documentacion/2020_Escala_valoracion_sociofamiliar.pdf)

[26] Herramienta AWS Lambda:

<https://aws.amazon.com/es/lambda/>

[27] Plataforma de mensajería Facebook Messenger:

<https://www.facebook.com/messenger/>

[28] Plataforma de mensajería Slack:

<https://slack.com/intl/es-es/help/articles/115004071768-¿Qué-es-Slack>

[29] Plataforma de Alexa Skill Kit

<https://developer.amazon.com/es-ES/alexa/alexa-skills-kit>