Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Electricidad y Computación

Implementación de un agente recepcionista utilizando un modelo de inteligencia artificial conversacional

Proyecto Integrador

Previo la obtención del Título de:

Nombre de la titulación
Ingeniero(a) en Ciencias de la Computación

Presentado por:
Luis Carlos Sánchez Plaza
Fiorella Giulliana Yerovi Nevárez

Guayaquil - Ecuador
Año: 2023

Dedicatoria

El presente proyecto se lo dedico a mis padres quienes me enseñaron el valor de la honestidad y el trabajo duro, quienes me brindaron su amor y apoyo incondicional; quienes me aconsejaron y me enseñaron a nunca rendirme.

Luis Carlos Sánchez Plaza

Agradecimientos

Mi más sincero agradecimiento a mi Padre, a mi Madre y a mi Hermana; porque fueron ellos quienes estuvieron conmigo en los buenos y malos momentos, sin ellos no hubiera llegado a donde estoy hoy; agradezco a Loki quien siempre logra sacarme una sonrisa en mi rostro.

Luis Carlos Sánchez Plaza

Declaración Expresa

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Luis Carlos Sánchez Plaza y Fiorella Giulliana Yerovi Nevárez damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

Firmado electrónicamente por:
LUIS CARLOS SANCHEZ PLAZA

Luis Sánchez Plaza

Firmado electrónicamente por:
FIORELLA GIULLIANA YEROVI NEVAREZ

Fiorella Yerovi Nevárez

Evaluadores

Firmado electrónicamente por:
ERICK VICENTE LAVID CEDENO

Erick Lavid Cedeño

Profesor de Materia

Firmado electrónicamente por:
DENNYS FABIAN PAILLACHO CHILUIZA

Dennys Paillacho Chiluiza

Tutor de proyecto

Resumen

El Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales (CIDIS) tiene problemas relacionados con la falta de un recepcionista dedicado dentro de sus oficinas. Debido a esto, los empleados del centro son los encargados de recurrir a ayudar a los visitantes cuando estos lleguen, lo que conlleva a una reducción en el rendimiento de sus labores; y a su vez, ocasiona un malestar en los visitantes debido a las esperas para obtener asistencia y aclarar sus inquietudes. El objetivo del proyecto fue diseñar y construir un agente recepcionista (AR) que resuelva las dudas propuestas por los visitantes dentro de la oficina del CIDIS. Para implementar la solución se desarrolló y entrenó un modelo de inteligencia artificial capaz de comprender las preguntas del visitante y responderlas. Adicional, se desarrollaron e incorporaron diferentes módulos a la solución, que permitieron que el AR se comunique de manera verbal con el visitante y muestre una interfaz gráfica. Como resultado, el AR se encuentra integrado en el CIDIS preparado para atender las diferentes consultas de los visitantes, lo que lleva a una mejora en la productividad de los empleados y brindando una mejor experiencia al visitante.

Palabras Clave: Inteligencia Artificial, Reconocimiento de Voz, Síntesis de Voz, Comunicación Natural.

Abstract

The Center of Research, Development, and Innovation of Computer Systems (CIDIS) faces issues related to the absence of a dedicated receptionist within its offices. Consequently, the center's employees are entrusted with assisting visitors upon their arrival, leading to a reduction in their work performance. This scenario causes discomfort among visitors due to the wait times for assistance and clarification of their inquiries. The project's objective was to design and construct a Receptionist Agent (AR) that addresses the questions posed by visitors within the CIDIS office. To implement the solution, an artificial intelligence model capable of comprehending visitor questions and providing responses was developed and trained. Additionally, various modules were developed and incorporated into the solution, enabling the AR to engage in verbal communication with visitors and display a graphical interface. As a result, the AR is integrated within CIDIS office, ready to address various visitor questions, resulting in improved employee productivity and an enhanced visitor experience.

Keywords: *Artificial Intelligence, Speech Recognition, Voice Synthesis, Natural Communication.*

Índice general

Abreviaturas

ESPOL	Escuela Superior Politécnica del Litoral
CIDIS	Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales
ML	Machine Learning
PLN	Procesamiento del Lenguaje Natural
CLN	Comprensión del Lenguaje Natural
IA	Inteligencia Artificial
UI	Interfaz de usuario
AR	Agente Recepcionista

Índice de figuras

Índice de tablas

Capítulo 1

1.1 Introducción

La inteligencia artificial es una disciplina de las ciencias de la computación que se enfoca en el desarrollo de algoritmos matemáticos capaces de imitar la inteligencia humana, con el propósito de que realicen tareas de una manera mucho más precisa y efectiva [1]. En los últimos años, la popularidad de la inteligencia artificial ha incrementado debido a que empresas como OpenAI [2], Google [3] y Microsoft [4], demostraron el potencial que tiene la implementación de los modelos de aprendizaje automático en sus productos y servicios que ofrecen.

En la actualidad, la inteligencia artificial se ha consolidado como una herramienta indispensable para solucionar problemas cotidianos. La implementación de modelos de inteligencia artificial en el sector comercial ha demostrado un aumento en la eficiencia y productividad de las empresas [5]. Motivo por el cual muchas empresas han optado por implementar este tipo de modelos en áreas en las que más lo necesitan, en muchos casos estas suelen ser las áreas de atención al cliente.

Usualmente, las empresas cuyo enfoque principal está relacionado con la atención al cliente tienden a hacer uso de agentes conversacionales ya que se ha comprobado que tienen la capacidad de influir positivamente en la satisfacción de los clientes [6]; además, su uso permite reducir costos y aumentar la productividad, dado que son capaces de atender cientos de clientes simultáneamente [7].

El presente trabajo se enfoca en la implementación de un agente recepcionista (AR) utilizando un modelo de inteligencia artificial conversacional para el Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales (CIDIS) de la Escuela Superior Politécnica del Litoral (ESPOL), cuyo propósito es el de mejorar la satisfacción de las personas que acuden a este centro.

1.2 Descripción del problema

Generalmente dentro de una oficina acuden visitantes con dudas relacionadas a los servicios que la empresa ofrece, o acerca de los miembros de la misma; por lo que se suele contar con una recepcionista la cual responda a todas estas inquietudes. Sin embargo, no siempre se cuenta con una recepcionista que pueda atender a los visitantes debido a diferentes razones como: los costos operacionales que trae el contratar a una persona o porque la recepcionista cuenta con horarios laborales establecidos y por tanto no está disponible todo el tiempo.

La falta de un recepcionista puede generar ciertos problemas en las empresas; que, aunque dichos problemas no son críticos pueden llegar a afectar negativamente la experiencia del cliente, debido a que tendrán que esperar para ser atendidos, y a la productividad de los empleados, ya que están propensos a ser interrumpidos y deberán dejar su trabajo a un lado para acudir al llamado del cliente.

En la actualidad, el CIDIS no cuenta con una recepcionista designada, sino que los empleados que trabajan en este centro comparten las tareas de recepción, por lo que no siempre alguien puede contestar las dudas de los visitantes que acuden al centro. Normalmente, las personas que visitan el centro tienen dudas acerca del centro de investigación, de las personas que trabajan en las diferentes áreas del mismo, o desean que les den indicaciones acerca de la ubicación de otros centros de ESPOL. Al no contar con una recepcionista en el centro, se cuenta con posibles malentendidos entre el visitante y el trabajador del CIDIS y poca disponibilidad de atención debido a que deben de dejar su puesto de trabajo para poder atender al visitante.

1.3 Justificación del problema

La resolución de la problemática descrita en la sección anterior traería una serie de beneficios tanto para la eficiencia como eficacia del centro. El contar con una figura que actúe como recepcionista en el centro, daría la posibilidad de que se cuente con una fuente centralizada de información para orientar a los visitantes con respuestas claras y más precisas, disminuyendo los posibles malentendidos. Con esta recepcionista el centro mejoraría la experiencia de los visitantes, mostrándose más profesional y organizado, y así aumentando su reputación.

Además, el contar con un recepcionista reduciría significativamente las interrupciones, hacia los trabajadores del CIDIS, provocadas por los visitantes; mejorando así, la productividad y la calidad del trabajo de los empleados.

Por último, no solo se mejoraría la experiencia de los visitantes y la calidad de trabajo de los empleados, sino que también se podría tener una mejora dentro de la seguridad del centro. Al tener una recepcionista se podría llevar una bitácora de las personas que ingresan al centro, para así tener un control de los visitantes.

1.4 Objetivos

1.4.1 Objetivo general

Implementar un agente recepcionista utilizando un modelo de inteligencia artificial conversacional, permitiendo una conversación natural entre un humano y el agente en un entorno de oficina.

1.4.2 Objetivos específicos

1. Entrenar y ajustar el modelo de inteligencia artificial utilizando conjuntos de datos relevantes para la tarea específica del agente recepcionista.

2. Diseñar una arquitectura de sistema que permita una interacción fluida y natural entre el humano y el agente recepcionista, incorporando los procesos de reconocimiento de voz y síntesis de voz para facilitar la conversación.
3. Implementar la interfaz de usuario y la integración del agente recepcionista en un entorno de oficina, asegurando su fácil uso.

1.5 Marco teórico

La implementación de chatbots como mecanismo para suplir las necesidades en las áreas de atención al cliente, es una táctica redituable [6] que se ha popularizado entre las empresas, a tal punto que en la actualidad existen varias instituciones gubernamentales, bancarias, médicas e incluso hoteleras, las cuales decidieron hacer uso de chatbots [10]. Sin embargo, también se han realizado implementaciones de chatbots en otros sectores, con el objetivo de resolver un problema en particular; como prueba de ello, hablaremos un poco sobre estos casos particulares.

El primer caso involucra una aerolínea mexicana la cual, en el año 2020, buscaba mejorar la calidad de la comunicación con los clientes del Club Premier, de tal manera que se pudiera aumentar la cantidad de usuarios afiliados al mismo; para esto, Garibay, F. decidió implementar un modelo de inteligencia artificial conversacional llamada Aivo el cual contaba con la capacidad de conectarse a redes sociales como Facebook o WhatsApp. Como resultado de la implementación, se pronosticó una reducción en los costos y una mejora en la calidad de la atención al cliente, lo que daba lugar a un aumento en las suscripciones al Club Premier [8].

En el segundo caso, un campamento de ciberseguridad para jóvenes procuraba reducir la carga de trabajo de sus empleados, quienes tenían que responder las preguntas recibidas a través de correo electrónico, muchas de las cuales eran preguntas repetidas. Además, a causa de la pandemia del COVID-19, se buscaba una forma de reemplazar los puestos de información que

requerían de uno o más empleados en el sitio. Para la solución de esta problemática, se empleó la herramienta de DialogFlow de Google con la que se desarrolló un modelo conversacional y como resultados se obtuvo una mejora en el servicio, además, por medio de encuestas a los estudiantes se comprobó que el 82.2% se mostró satisfecho con el chatbot [10].

En el último caso, como parte de un proyecto de titulación, se implementó un prototipo de chatbot para la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato. El motivo por el que se implementó esta solución fue para mejorar y agilizar las respuestas a las preguntas de índole académica, que suelen hacer en el área de recepción de la facultad. Para el desarrollo de la solución se usó Watson Assistant de IBM y como resultados, se obtuvo un correcto funcionamiento en las pruebas realizadas en un ambiente controlado [9].

1.5.1 Modelos de inteligencia artificial conversacional

La inteligencia artificial conversacional permite que los usuarios humanos se comuniquen de manera verbal o escrita con un sistema usando lenguaje natural [11]. Para ser capaz de imitar una interacción humana estos modelos utilizan Machine Learning (ML) y Procesamiento del Lenguaje Natural (PLN).

PLN es un método el cual analiza el lenguaje y consta de cuatro pasos [12]:

1. Generación de la entrada: El usuario provee una entrada a través de una interfaz de manera verbal o escrita.
2. Análisis de la entrada: Dependiendo de si la entrada es verbal o escrita realizará un análisis distinto. En caso de ser escrita, el modelo analiza la entrada del usuario usando la Comprensión del Lenguaje Natural (CLN), en donde se clasifica la entrada de manera semántica para entender la intención del mensaje y la información

relevante del mismo [13]. Mientras que, si es verbal primero realizará un reconocimiento de voz y luego analizará la entrada con CLN.

3. Gestión de diálogos: Se genera la respuesta utilizando lenguaje natural por parte del modelo.
4. Aprendizaje por refuerzo: El modelo mejora las respuestas que da a medida que pasa el tiempo para asegurar una mejor precisión.

1.5.2 Reconocimiento de voz

El reconocimiento de voz permite que el sistema comprenda al usuario humano cuando este provee una entrada de voz. Para convertir una entrada de voz a una cadena de caracteres primero se convierte la entrada de voz a una serie numérica la cual contiene los sonidos vocales de la entrada y luego esta serie numérica se la traduce utilizando 3 bases de datos (el modelo acústico, lista léxica y el modelo de lenguaje) [14].

1.5.3 Síntesis de voz

La síntesis de voz convierte una cadena de caracteres a una señal sonora de habla artificial [15]. Para convertir la cadena de caracteres lo que realiza la síntesis de voz es preprocesar la cadena de caracteres asignando transcripciones fonéticas a cada palabra y luego dividiéndolas en unidades prosódicas como lo son las oraciones o frases. Tanto las transcripciones fonéticas como las unidades prosódicas luego se utilizan para generar el habla de manera artificial con distintos métodos de síntesis [14].

1.5.4 Gesticulación

La gesticulación se realiza con la utilización de un avatar el cual en base a los resultados de la síntesis de voz genera una animación facial de la respuesta. Para lograr que se genere la animación facial del avatar se debe de contar con el avatar ya sea 2D o 3D y luego en una

herramienta de motor gráfico como Unreal Engine, Blender o Unity se debe de determinar las animaciones correspondientes a las transcripciones fonéticas y la posición de la boca (visemas) [16].

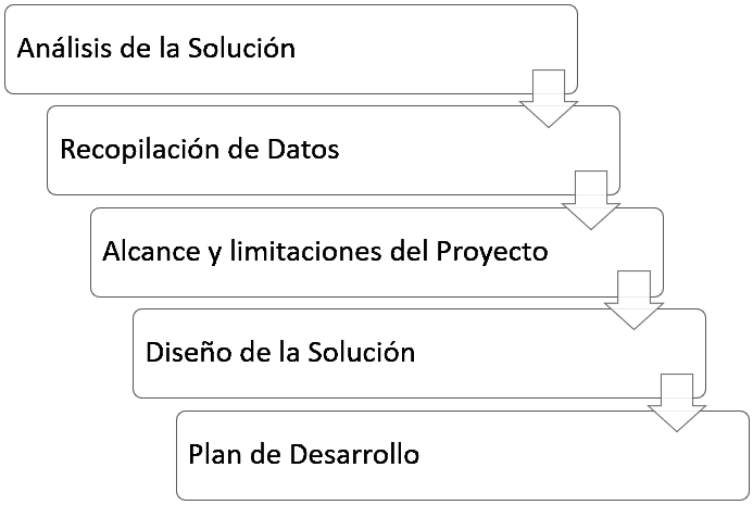
Capítulo 2

2.1 Metodología.

El enfoque de este proyecto fue el de implementar un modelo de inteligencia artificial capaz de simular la interacción que tendría un visitante con un recepcionista, analizando preguntas y generando respuestas concisas. Dejando claro lo anterior, el objetivo principal era hacer uso de ese modelo para crear un agente conversacional que fuera capaz de comprender las preguntas realizadas por los visitantes del CIDIS, las que se daban por medio del lenguaje natural, y responder a ellas de manera verbal y coherente. El proceso que se siguió para lograr el desarrollo de la solución contó con los pasos detallados en la Figura 1.

Figura 1

Pasos a seguir en la metodología



Nota. La figura muestra los pasos que se siguieron para lograr el desarrollo del proyecto.

2.2 Análisis de la solución

El primer paso en el proceso consistió en el levantamiento de los requerimientos del proyecto, puesto que el objetivo era comprender las necesidades y expectativas del cliente para establecer una base de la cual se pueda partir. Para esto, se realizaron varias reuniones con el

cliente y empleados del CIDIS en las que se buscó entender el problema y ver cómo afectaba a quienes trabajaban en la oficina.

Durante las reuniones se logró despejar dudas e identificar aspectos clave del problema, y también surgieron ideas por parte del cliente las cuales deseaba incorporar en la solución. Posteriormente, se realizó un análisis de todos los puntos antes mencionados y en base a estos se levantaron los requerimientos funcionales del proyecto, los cuales se detallan a continuación.

2.2.1 Requerimientos funcionales

1. El AR deberá ser capaz de reconocer el lenguaje natural hablado (voz) en español, para el posterior procesamiento de las solicitudes de los visitantes.
2. El AR deberá de contar con un sistema de activación por comando de voz para poder iniciar la interacción verbal con el visitante.
3. El AR deberá de ser capaz de generar respuestas acertadas y de manera inteligente a las respectivas solicitudes realizadas por los visitantes.
4. El AR deberá ser capaz de proporcionar las respuestas a las solicitudes de los visitantes de manera verbal haciendo uso del lenguaje natural en español.
5. El AR deberá de conocer quiénes trabajan en las diferentes áreas del CIDIS, con el objetivo de brindar información sobre sus miembros en caso de ser necesario.
6. El AR deberá de conocer la ubicación del CIDIS y demás centros de la ESPOL; de tal manera, que pueda brindar indicaciones claras de cómo llegar a los otros centros si un visitante lo solicita.
7. El AR deberá mostrar o mencionar información sobre el CIDIS, actividades e investigaciones, para captar la atención del visitante mientras espera.

8. El AR deberá de guardar la información de los visitantes, como su nombre y la hora en la que ingresó, dentro de una bitácora para conocer quiénes se han acercado al CIDIS.

2.3 Recopilación de Datos

Una vez levantados los requerimientos funcionales de la solución el siguiente paso era empezar a recabar la información necesaria para poder entrenar el modelo. Para ello, se procedió a recopilar información en la página web del CIDIS, la información que se logró recopilar va desde lo más general como: la misión y visión del CIDIS, valores y objetivos, áreas de investigación y proyectos realizados; hasta información más específica como: quiénes son los coordinadores de cada departamento, quiénes trabajan ahí y en qué campos de la ciencia se enfocan sus proyectos. Sin embargo, la información que se recopiló no fue suficiente, ya que solo estaba focalizada a datos puntuales sobre el CIDIS; por ende, se decidió tratar de obtener información de campo, conversando con la persona que se hacía cargo de la recepción del CIDIS con el objetivo de poder indagar sobre la clase de visitantes que recibe el centro, la frecuencia y el contexto de las visitas.

Para recopilar la información de las ubicaciones de los otros centros de la universidad se comenzó realizando una investigación con el objetivo de identificar y localizar los diferentes centros de la ESPOL. Luego se realizó un recorrido en la ESPOL, tomando como punto de partida el CIDIS, hacia los diferentes centros tomando apuntes de cómo llegar a los mismos para así poder generar las respectivas indicaciones.

2.4 Alcance y limitaciones del proyecto

Una vez definidos los requerimientos funcionales y recopilada la información que se usaría para el entrenamiento, se procedió a definir el alcance del proyecto. La solución se enfoca

en la implementación de un modelo de inteligencia artificial que sea capaz de comprender el lenguaje natural y que brinde respuestas claras y precisas; puesto que, el objetivo final era que dicho modelo en las oficinas del CIDIS cumpliera las funciones de un recepcionista. Los aspectos que se incluyeron en el alcance de la solución fueron los siguientes:

- El modelo debía ser capaz de comprender y procesar el lenguaje natural, para luego generar una respuesta coherente y de manera verbal.
- El modelo debía ser capaz de conocer quiénes trabajaban en el CIDIS, además de conocer sobre su información de contacto (correo electrónico).
- El modelo debía ser capaz de dar indicaciones a los visitantes sobre la ubicación de los demás centros de la ESPOL.

Por otro lado, durante el periodo de análisis de los requerimientos se identificó ciertas limitaciones en la solución, las cuales fueron delimitadas tomando en cuenta aspectos como el tiempo y la complejidad de desarrollo:

- El modelo solo será capaz de comprender un idioma, el español.
- El modelo solo interactúa con el visitante de manera verbal, reconociendo y sintetizando la voz.
- El modelo solo desempeñará la labor de recepcionista, lo cual implica que el modelo está enfocado en brindar información específica del centro y sus empleados.
- El modelo será implementado dentro de las oficinas del CIDIS; por ende, no se toma en cuenta la implementación en otras oficinas o centros de ESPOL.

2.5 Diseño

La solución propuesta para este proyecto consistió en la implementación de un modelo de inteligencia artificial conversacional el cual interactúa con un humano mostrando un modelo 3D que gesticula la síntesis de voz, en un robot, con el enfoque de recepcionista en el CIDIS. Para diseñar esta solución se tomó en consideración varios módulos como lo son: el reconocimiento de voz, la síntesis de voz, el modelo de inteligencia artificial conversacional y la renderización del modelo 3D que gesticula en el robot.

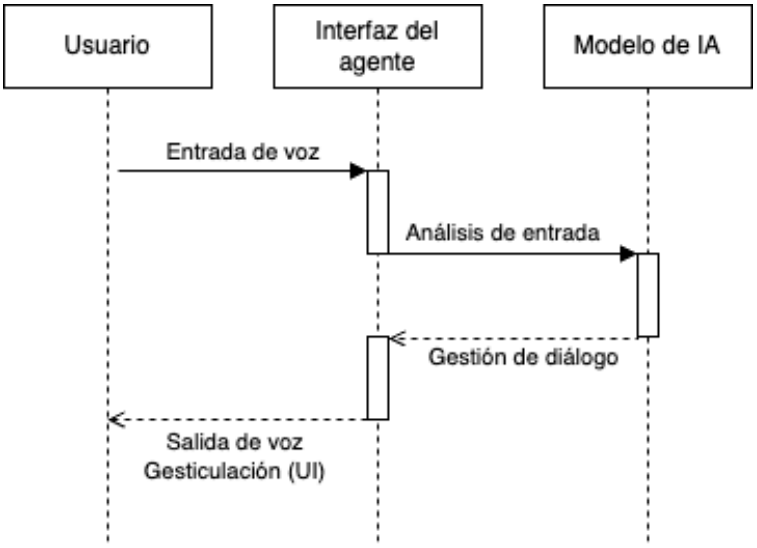
2.5.1 Diseño del sistema

El sistema consiste en un agente recepcionista el cual está alojado dentro de un robot (el cual es un monitor ubicado en una plataforma dentro de la recepción del CIDIS). Este AR es quien interactúa con el visitante mediante voz y mostrando un modelo 3D renderizado.

En la Figura 2 se muestra el diagrama de secuencia del sistema en donde se explica la interacción entre el usuario, la interfaz del agente (agente recepcionista alojado en el robot) y el modelo de inteligencia artificial (IA). El usuario interactúa con la interfaz del agente por entrada de voz, para dar a conocer su requerimiento ante el agente recepcionista. Luego, el agente recepcionista analiza la entrada recibida con reconocimiento de voz, y a partir de la respuesta del modelo de reconocimiento de voz, analiza la entrada usando CLN para que el agente comprenda la intención del mensaje y la información más relevante del mismo. Con base en el análisis de la CLN, el modelo de inteligencia artificial devuelve una respuesta coherente en lenguaje natural, a esto se le denomina gestión de diálogo. Por último, la interfaz del agente devuelve una salida de voz realizando síntesis de voz y también muestra un modelo 3D renderizado que gesticula la salida de voz en la interfaz de usuario (UI).

Figura 2

Diagrama de secuencia del sistema



Nota. La figura muestra el diagrama de secuencia de la solución en donde se tiene como objetos al usuario, interfaz del agente recepcionista y al modelo de inteligencia artificial.

Para cada uno de los módulos antes mencionados se tiene en consideración las siguientes herramientas y modelos para la implementación de los mismos:

Modelo de inteligencia artificial conversacional. Para que el agente recepcionista sea capaz de generar una respuesta necesita de un modelo de inteligencia artificial conversacional. En la actualidad, existe una gran variedad de modelos como ChatGPT, sin embargo, para este proyecto se eligió a Rasa debido a que ofrece un modelo de software libre flexible que, aunque no es tan sofisticado como lo es ChatGPT provee un modelo de inteligencia artificial personalizable [19] lo suficiente potente para responder como recepcionista.

Reconocimiento de voz. Para el reconocimiento de voz existen variedad de modelos de inteligencia artificial los cuáles están entrenados para realizar esta acción. Para este proyecto se

eligió el modelo Whisper AI debido a que es un modelo de software libre el cual cuenta con modelos pre entrenados en el idioma español con un gran volumen de conjunto de voces [17].

Síntesis de voz. Para la síntesis de voz se eligió el sintetizador de voz Bark, quien da una respuesta rápida y clara al sintetizar la voz [18] a partir de la respuesta escrita generada por el modelo de inteligencia artificial.

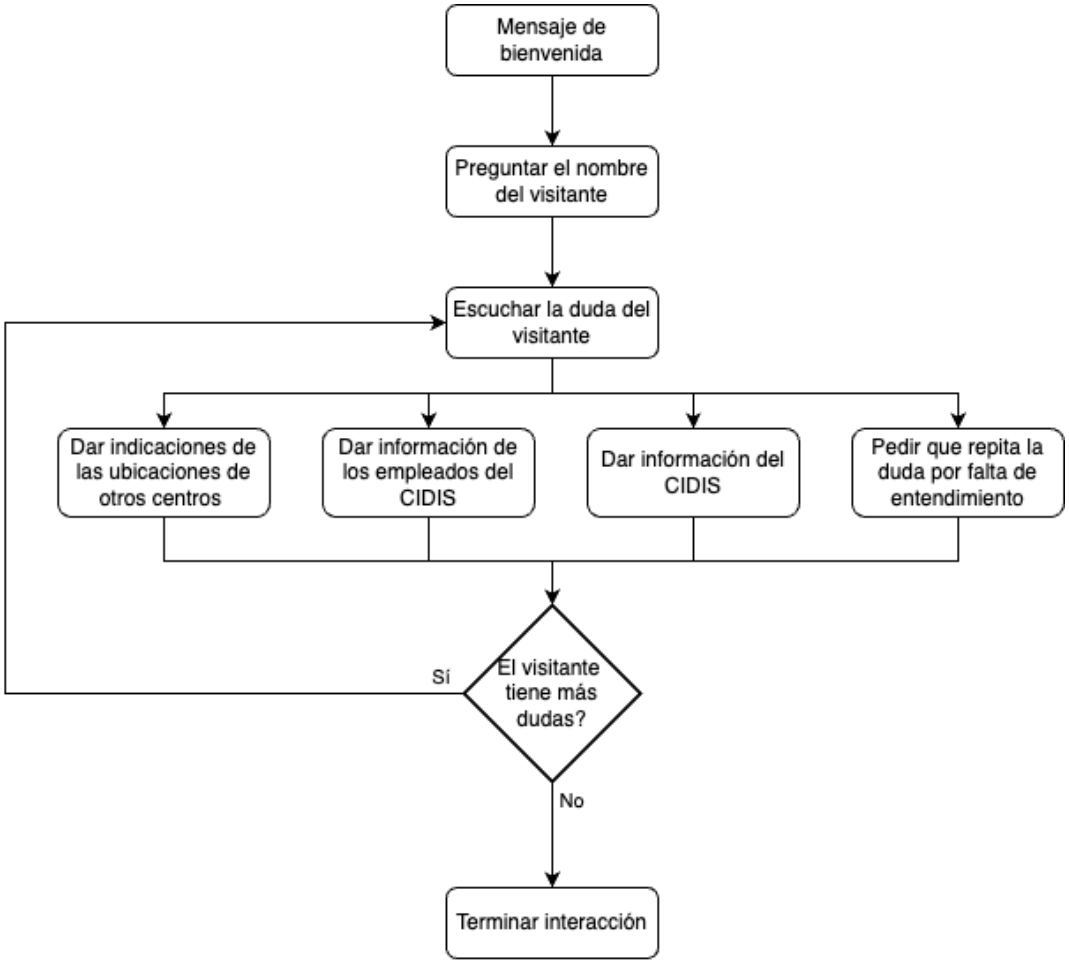
Renderización del modelo 3D. Para la creación de un modelo 3D se utilizó la herramienta Ready Player Me para crear al personaje ya que es una herramienta de software libre la cual permite la creación de personajes de manera sencilla [20]. Teniendo el modelo 3D creado, se utilizó la herramienta Blender [21] para animar este modelo y permitir que gesticule cada frase que el agente recepcionista genere con su síntesis de voz.

2.5.2 Flujo de conversación

El flujo de conversación es una parte esencial al enseñarle al modelo de inteligencia artificial conversacional a interactuar con los humanos. En la Figura 3, se muestra un resumen del flujo de conversación que tuvo el sistema con el visitante. Se inicia con un mensaje de bienvenida al haber sido activado por un comando de voz y pregunta por el nombre del visitante para registrarlo en la bitácora de la oficina. Luego comienza a escuchar las dudas que tenga el visitante, tomando en cuenta que responderá a dudas de temas específicos como: indicaciones de las ubicaciones de otros centros de ESPOL, información acerca de los empleados del CIDIS e información del CIDIS. Y en caso de no comprender la entrada de voz del visitante le pedirá que repita la duda. Cuando el visitante ya no tenga más dudas relacionadas a los temas que conoce el agente recepcionista terminará el flujo de conversación con el mismo.

Figura 3

Flujo de conversación del sistema



Nota. La figura muestra el flujo de conversación del agente recepcionista.

Para implementar el modelo de inteligencia artificial conversacional de Rasa se determinó la siguiente información para entrenarlo:

Datos para el modelo CLN. Se describieron las intenciones (intents) que tendría el visitante al hablar con el agente recepcionista como lo es el pedir indicaciones y pedir información del CIDIS y sus empleados. En la siguiente tabla se observa algunos de los datos

que se utilizaron, tomando en cuenta que se debe de poner varios ejemplos para cada una de las intenciones.

Tabla 1

Datos para el modelo CLN

Intent	Ejemplos
Saludo	‘Hola’
	‘Hey’
	‘Buenos días’
	‘Buenas tardes’
	‘Buenas’
	‘Buenas noches’
	…
Pedir indicaciones	‘¿Dónde está {FIEC}?’
	‘¿Estoy en {FIEC}?’
	‘¿Qué tan lejos está {FIEC}?’
	‘¿Me puedes decir cómo llegar a {FIEC}?’
	‘Quiero ir a {FIEC}’
	…

Nota. Estos son algunos de los datos que se utilizaron para el entrenamiento del CLN.

Respuestas. Conociendo los intents que tendrá el visitante al interactuar con el robot, se debe de especificar algunas respuestas que podría darle en base al intent que tenga el visitante. En la siguiente tabla se observa algunos de los datos que se utilizaron como respuesta, tomando en cuenta que se debe de poner varios ejemplos.

Tabla 2

Datos de respuesta del modelo

Respuesta	Ejemplos
Dar Saludo	‘Hola, ¿cuál es tu nombre?’ ‘Bienvenido al CIDIS, ¿cuál es tu nombre?’ ‘Te doy la bienvenida al CIDIS ¿cómo te llamas?’
Dar indicaciones	Para poder llegar al {FIEC} deberás seguir las siguientes indicaciones {salir de esta oficina y dirigirte hacia la calle principal de ESPOL e ingresar al campus …}

Nota. Estos son algunas de las respuestas que se especificaron para el modelo.

Historias. Teniendo todos los intents y respuestas para el modelo se crearon historias, las cuales son ejemplos de conversación para entrenar al modelo a responder correctamente dependiendo del contexto de la interacción que tiene con el visitante. Por ejemplo, con los dos intents y respuestas que se especificaron anteriormente se realiza la siguiente historia:

1. Intención: Saludo
2. Respuesta: Dar Saludo
3. Intención: Pedir indicaciones
4. Respuesta: Dar indicaciones

En esta historia se muestra el flujo de conversación que tiene el robot para ayudar al visitante a ubicar dónde está el lugar que desea encontrar.

Formularios. Los formularios sirven para recolectar información del visitante por lo que se especificó un formulario para recibir el nombre de la persona al iniciar la conversación.

2.6 Plan de desarrollo

Para implementar la solución se comenzó identificando las necesidades del cliente para así crear el plan de desarrollo que consiste en el desarrollo de 4 módulos y la integración del agente recepcionista dentro de la oficina del CIDIS.

Los módulos que se definieron son: el modelo de inteligencia artificial conversacional, el reconocimiento de voz, la síntesis de voz, y la renderización del modelo 3D que gesticula en el robot.

- ***Módulo del modelo de inteligencia artificial conversacional:*** Utilizando Rasa se entrenó al modelo de inteligencia artificial conversacional que es quien responde las inquietudes de los visitantes del CIDIS.
- ***Módulo de reconocimiento de voz:*** Con Whisper AI y sus modelos pre entrenados de reconocimiento de voz se realizó el módulo de reconocimiento de voz y se verificó la precisión del mismo dentro de la oficina del CIDIS.
- ***Módulo de síntesis de voz:*** Teniendo el módulo del modelo de IA conversacional se utilizó Bark para así transformar el texto en voz y que el agente recepcionista dé la respuesta de manera hablada.
- ***Módulo de renderización del modelo 3D:*** Utilizando las herramientas: Ready Player Me y Blender, se creó un modelo 3D, el cual es la representación visual del agente recepcionista. Luego se le implementó la funcionalidad de gesticular para que el modelo tenga una animación que dependa del resultado del módulo de síntesis de voz.

Por último, se realizó la integración de todos los módulos para conseguir la solución final que es el agente recepcionista dentro de las oficinas del CIDIS. En la siguiente tabla se muestra un resumen del plan de desarrollo que se realizó.

Tabla 3

Plan de desarrollo del proyecto

Tarea	Duración de la tarea
Módulo de modelo de inteligencia artificial conversacional	4 semanas
Módulo de reconocimiento de voz	2 semanas
Módulo de síntesis de voz	2 semanas
Módulo de renderizado del modelo 3D	3 semanas
Integración del agente recepcionista en la oficina	1 semana

Nota. Se especifica las tareas que se realizaron para desarrollar la solución con la duración respectiva de cada tarea.

Capítulo 3

3.1 Resultados del desarrollo

En base al plan de desarrollo que se definió en el Capítulo 2, se procedió a realizar la implementación de los módulos que serían necesarios para completar la solución. A continuación, se presentarán los resultados que se obtuvieron en cada uno de estos módulos los cuales fueron desarrollados dentro de los periodos establecidos en la Tabla 3.

3.1.1 Módulo de modelo de inteligencia artificial conversacional

Este módulo fue desarrollado para identificar, analizar y responder las peticiones proporcionadas por el visitante. Durante el desarrollo se tuvo que tomar en cuenta 2 aspectos importantes: los datos de entrenamiento y los componentes a usar.

El primer aspecto consistía en clasificar los datos de entrenamiento en intents, estos debían de reflejar las preguntas más comunes las cuales el AR debía de ser capaz de responder, de tal manera, que así le fuese fácil aprender el tipo de preguntas que le podrían hacer en un entorno real.

Para el segundo aspecto, se incorporó aquellos componentes de Rasa que proporcionarán una mejora en la precisión del reconocimiento y clasificación de los intents y una correcta extracción de las entities (palabras claves que se extraen del mensaje). Algunos de los componentes que se incorporaron al modelo fueron los siguientes:

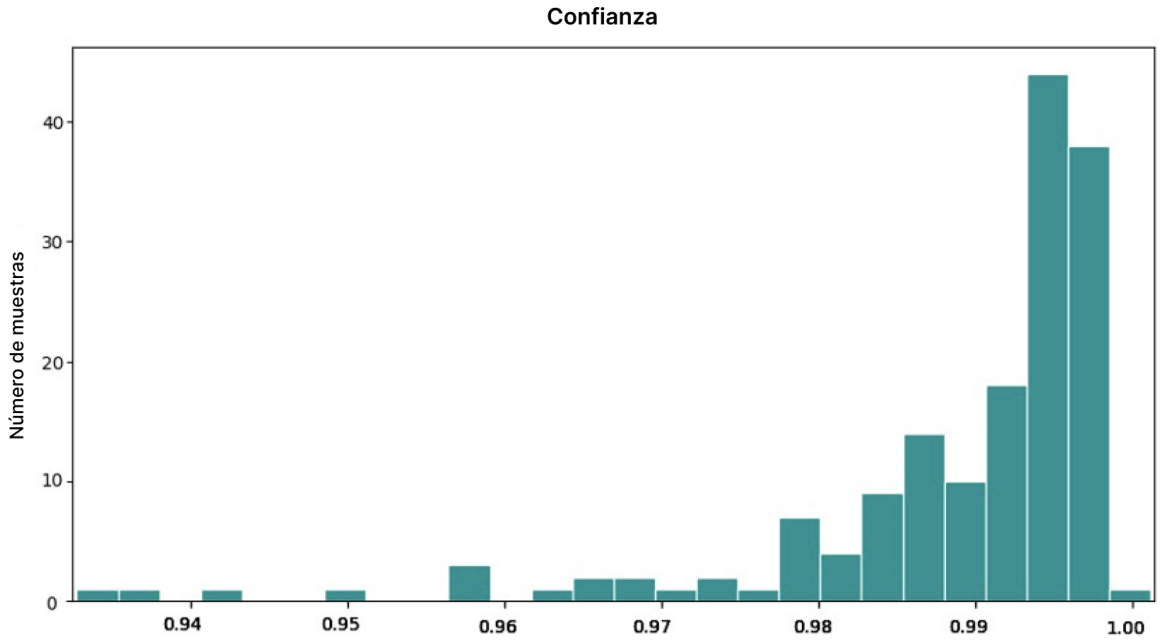
- ***SpacyNLP:*** Componente que incorporaba un modelo del lenguaje en español previamente entrenado.
- ***LexicalSyntacticFeaturizer:*** Componente capaz de crear características léxicas y sintácticas para la pregunta del visitante, de tal manera, que fuera más sencilla la extracción de entities.

- ***DIETClassifier:*** Componente encargado de clasificar los intents y de extraer las entities presentes en el mensaje.
- ***EntitySynonymMapper:*** Componente capaz de mapear los sinónimos de las entities, de tal manera que todas apunten al mismo valor.

Una vez concluido el desarrollo del modelo, se procedió a realizar pruebas para validar la precisión del modelo al reconocer y clasificar los intents. Como se puede observar en la Figura 4, la precisión del modelo para poder clasificar los intents es relativamente alta, lo que asegura que el modelo es capaz de responder las preguntas correctamente.

Figura 4

Distribución de confianza al predecir intents



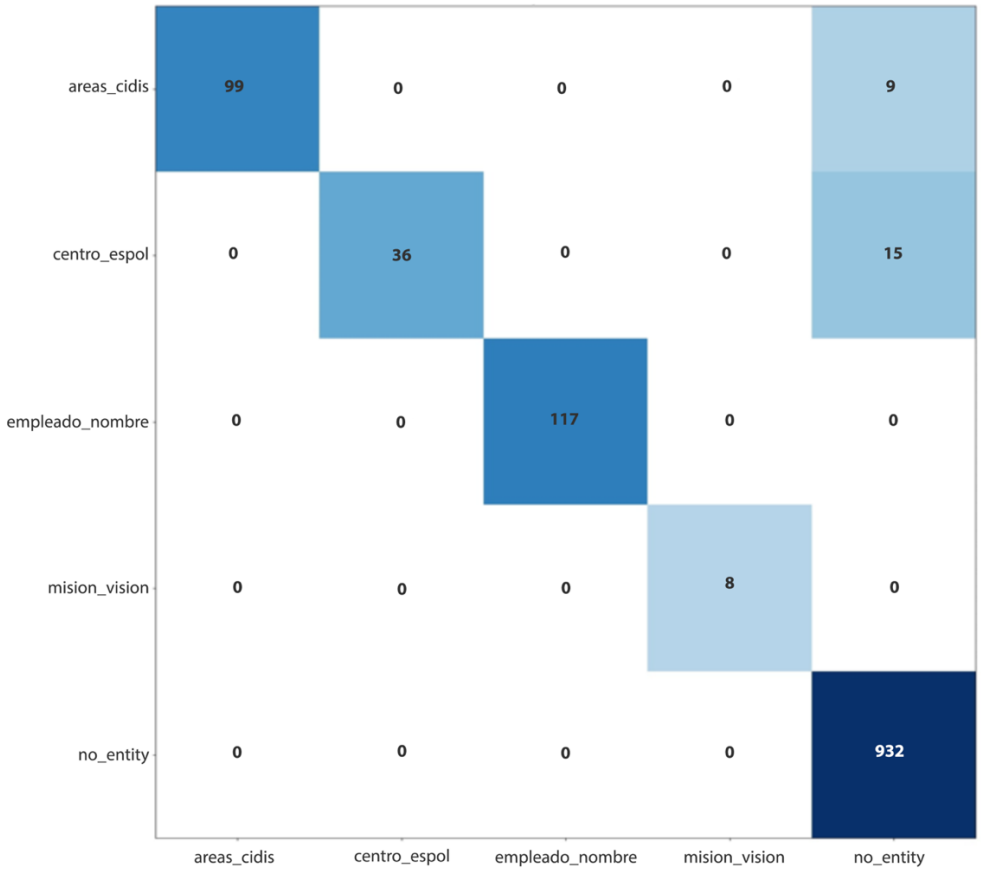
Nota. En el gráfico se muestra el número de intents que el modelo pudo predecir correctamente.

Además, también se procedió a realizar pruebas que nos permitieran corroborar la exactitud que tenía el modelo al momento de extraer las entities de las preguntas generadas por el

usuario. En la Figura 5 se observa que el modelo es capaz de extraer correctamente las entities de los intents recibidos en la mayoría de los escenarios. Sin embargo, en algunos casos el modelo ejecuta el proceso de extracción de entities sobre ciertos intents en los cuales no es necesario hacer dicho proceso, dado que representan preguntas más simples como: "¿Qué es el CIDIS?", en donde podría tratar de extraer la palabra CIDIS como una entity "centro_espol" aunque no es necesario. De igual forma, la probabilidad de que ocurra este error es muy baja y no afecta el rendimiento del modelo.

Figura 5

Matriz de confusión de entities

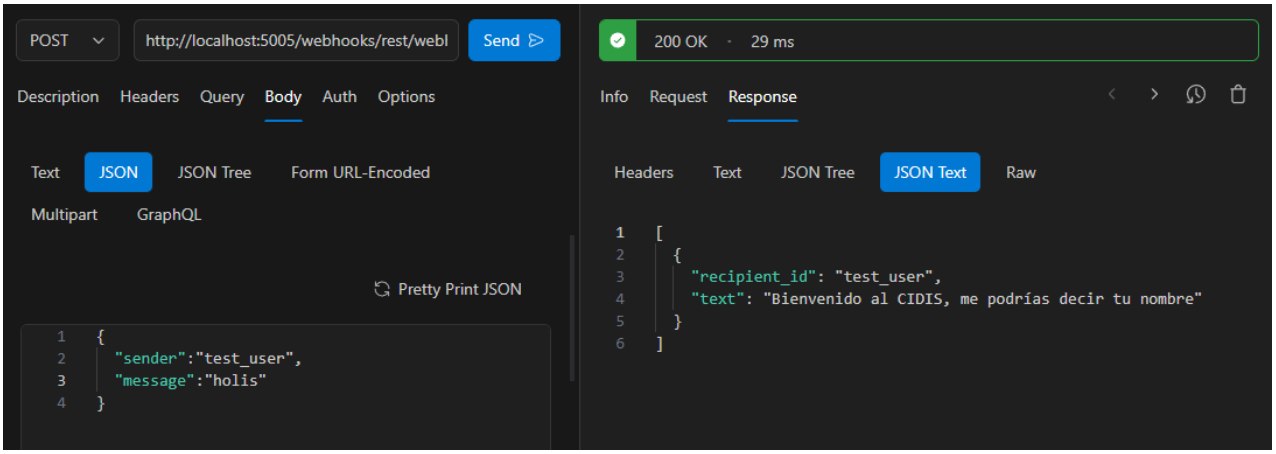


Nota. En el gráfico se muestra el total de entities que se extrajeron por cada intent.

Por otro lado, durante las pruebas se pudo identificar un problema que tenía el modelo, en el cual el modelo no era capaz de reconocer un intent, denominado “Saludo”, si el valor de entrada contaba con faltas ortográficas o si se hacía uso de jergas coloquiales, como es el caso de “Holis”. Se le presentó esta problemática al cliente y se decidió que se trataría de encontrar una solución, por lo que se volvió a entrenar al modelo añadiendo nuevos casos a los ejemplos de entrenamiento, en los que se usaron palabras con faltas ortográficas y jergas coloquiales.

Figura 6

Resultado del método POST al intent saludar



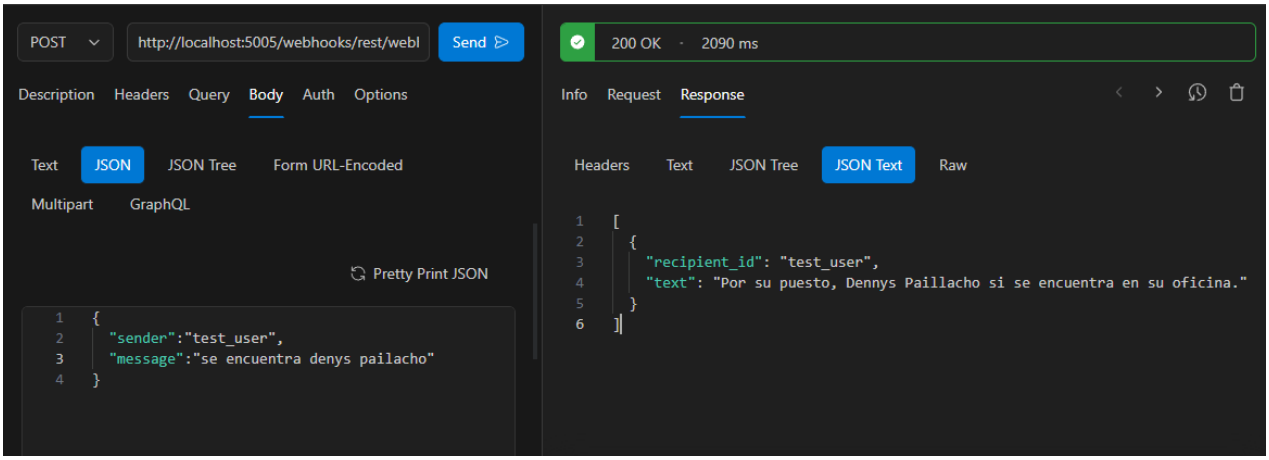
Nota. En el gráfico se puede observar como el modelo es capaz de identificar el Intent a pesar de que la palabra no sea gramaticalmente correcta.

Cabe destacar que también se logró identificar un error similar que ocurría cuando se trataba de obtener la información de un empleado, dado que si el nombre del empleado estaba escrito de una manera diferente a como se la encontraba en los registros, el modelo no devolvería la información del empleado. Por ende, para solucionar dicho error se cambió los métodos de tal manera que, en lugar de hacer una comparación exacta de palabras, se hiciera uso de un

secuenciador para encontrar el porcentaje de similitud entre dos palabras, si esta similitud superaba el Threshold, el cual fue establecido al 70%, devolvería la información del empleado.

Figura 7

Resultado del método POST al recibir nombres mal escritos



Nota. En el gráfico se puede observar que el modelo es capaz de identificar a un empleado a pesar de que el nombre se encuentre mal escrito.

3.1.2 Módulo de reconocimiento de voz

Este módulo fue desarrollado para que el AR detecte y entienda las palabras que el visitante emita. Durante el desarrollo se tuvo varios desafíos como la incapacidad de trabajar en tiempo real, ineficacia en el reconocimiento de ciertas palabras y la falta de modelos especializados solo para el español.

En el transcurso del proyecto, se tuvo en consideración usar WhisperAI para el reconocimiento de voz; sin embargo, este modelo demostró un bajo rendimiento a la hora de realizar el reconocimiento de voz en tiempo real. WhisperAI al realizar el reconocimiento de voz en tiempo real se demoraba de 6 a 10 segundos en transcribir lo que el visitante decía (en una

computadora i5 con 8GB de memoria). Lo cuál era inadecuado para nuestro proyecto porque el AR debía de ofrecer respuestas oportunas y efectivas, y al tener esta deficiencia en su rendimiento no nos era útil.

Además, al haber encontrado solo un modelo multilingüe para WhisperAI en ocasiones no reconocía correctamente los nombres, por ejemplo, se le decía “Dennys Paillacho” y entendía “Pay Jaco” o no comprendía que se le estaba hablando español y transcribía palabras que no se le habían dicho por las alucinaciones que llega a tener a veces este modelo cuando los audios son muy cortos [22].

Después de habernos encontrado con estos problemas, decidimos elegir a Vosk como la solución de reconocimiento de voz de este proyecto. A diferencia de WhisperAI, Vosk ofrece un modelo especializado para cada lenguaje, teniendo así un modelo más pequeño que reconoce mejor lo que se le dice solo en idioma español. Vosk también demostró ser eficiente a la hora de reconocer la voz en tiempo real, transcribiendo lo que se le dice de manera casi inmediata.

3.1.3 Módulo de síntesis de voz

Este módulo fue desarrollado con el objetivo de que el AR pudiera comunicarse con los visitantes de manera verbal, de tal manera que, la comunicación entre el visitante y el AR fuera mucho más sencilla. Durante el desarrollo se consideró que el modelo Bark era bastante prometedor, contaba con soporte para el lenguaje al español y las voces generadas por el modelo sonaban muy reales; el problema consistía en que el tiempo que tardaba en convertir el texto a voz era demasiado alto, tomaba alrededor de 30 segundos en oraciones pequeñas. Por lo que se optó por otro modelo para la síntesis de voz que fue Coqui TTS.

Coqui TTS aparte de ofrecer soporte para el lenguaje en español y tener voces muy realistas, era capaz de convertir el texto a voz en cuestión de segundos [25]. Con este modelo, se

procedió a hacer pruebas en Google Collab para determinar la veracidad de dichas afirmaciones, una vez concluidas las pruebas se encontró que efectivamente los tiempos de conversión era mucho menores a Bark, alrededor de 2 segundos en oraciones pequeñas, y las voces también eran muy realistas.

Para la integración del modelo Coqui TTS, se implementó el modelo en un contenedor de Docker, para que no existiera un conflicto de dependencias; dado que, Rasa hacia uso de las mismas librerías que Coqui TTS, pero en versiones más antiguas.

3.1.4 Módulo de renderizado del modelo 3D

Este módulo fue desarrollado para que el AR cuente con una interfaz gráfica la cual tendrá ciertas animaciones a la hora de interactuar con el visitante. Lo más importante en este módulo es la experiencia visual y realista que pueda aportar este modelo 3D, por lo que se comenzó generando el modelo 3D del AR con la herramienta Ready Player Me, la cual dio una gran variedad de atributos y características personalizables del modelo.

Luego se implementó varias animaciones de movimientos naturales para cuando esté inactivo y hablando el agente recepcionista, las cuales fueron obtenidas de Ready Player Me [23]. Al tener estos movimientos se sincronizó los mismos con el flujo de conversación que tendrá el AR, así creando un socket para que el AR que está programado en Python, se pueda comunicar con nuestra interfaz la cual está programada en JavaScript con la librería Babylon.js, en tiempo real, enviándole las respuestas que produce.

Se optó por utilizar Babylon.js en vez de Blender como herramienta de modelado, debido a que Babylon.js está diseñado específicamente para desarrollar aplicaciones webs en tiempo real, así teniendo un mejor rendimiento y garantizando que la experiencia visual no sea afectada por las posibles limitantes del computador en donde se despliegue [24].

3.2 Resultados y análisis de pruebas de la implementación

Durante la fase de implementación del modelo se realizaron pruebas con 20 usuarios para evaluar la usabilidad y efectividad de sus respuestas en un escenario real. Se tomó en consideración a distintos usuarios los cuales fueron: 15 estudiantes y 5 personas ajenas de ESPOL, cada uno interactuando con el AR sin ningún tipo de ayuda y realizando preguntas acerca del centro, de los trabajadores del centro y de cómo llegar a diferentes partes de la universidad. Luego de la interacción, los usuarios respondieron a un cuestionario, adjunto en el apéndice A, en donde se utilizó la escala de Likert junto con preguntas abiertas para evaluar al sistema.

Los resultados de estas pruebas, adjuntos en el apéndice B, revelaron que la mayoría de los usuarios encontraban la interfaz fácil de usar, debido a que era accesible, sólo tenían que hablarle y les respondería el AR. Sin embargo, en ocasiones, con usuarios que tenían un bajo nivel de familiaridad tecnológica se observó que tenían problemas para iniciar la interacción con el AR.

Los usuarios encontraron que el modelo 3D se mostraba agradable a la vista así haciendo que la interacción con el AR sea atractiva. Por último, ciertos usuarios señalaron que el AR no siempre les reconocía la pregunta que decían debido a que hablaban con un tono de voz bajo, pero en cualquier caso el AR si les llegaba a escuchar sus preguntas en la gran mayoría de casos.

3.3 Análisis de Costos

Para llevar a cabo este proyecto se tuvo en consideración el uso de recursos de código abierto y gratuitos, en donde tanto Rasa, Vosk, Ready Player Me y Coqui TTS ofrecen sus

servicios de manera gratuita y accesible, minimizando así a cero los costos relacionados con servicios de terceros.

El único costo significativo de este proyecto fue el desarrollo del AR por parte del equipo de desarrollo. Tomando en consideración que el equipo fue conformado por dos personas y que el proyecto duró 12 semanas, donde se trabajó 4 horas diarias los 5 días de la semana, se propuso un valor de $5 USD por hora, en concepto de desarrollo por el modelo del AR. Así dando como resultado un total de costos de $2,400.00.

Tabla 4

Análisis de costos del proyecto

Actividad	Cantidad	Costo
Horas de desarrollo por persona	240 * 2	$ 2,400.00
Recursos de código abierto	-	$ 0.00
Total	-	$ 2,400.00

Nota. Se especifica el análisis de costos que se tuvo en el proyecto.

3.4 Cierre del proyecto

El proyecto fue terminado a entera satisfacción de las partes involucradas. El proyecto pasó por diversas etapas de planificación, diseño, desarrollo y pruebas, contando con un sistema de recepción basado en inteligencia artificial que superó las expectativas previstas.

Lo que se incluyó en la entrega fue:

- Código fuente completo del agente recepcionista
- Datos de entrenamiento para el modelo utilizados
- Guía de usuario e implementación (adjuntas en los apéndices)

Para más detalles sobre el cierre del proyecto, se hace referencia al "Acta de Entrega del Proyecto" adjunta en los apéndices.

Capítulo 4

4.1 Conclusiones y recomendaciones

4.1.1 Conclusiones

El presente trabajo se ha enfocado en enfrentar los desafíos que presentaba el CIDIS, específicamente el problema de la falta de un recepcionista que reducía la productividad de los empleados y generaba una mala experiencia en el visitante. Junto con la culminación del proyecto, se logró alcanzar lo siguiente:

- Desarrollo de un modelo de inteligencia de inteligencia artificial capaz de interactuar, por medio del lenguaje natural, con los visitantes del CIDIS; logrando que la resolución de las dudas planteadas por los visitantes sea casi instantánea, mejorando positivamente la experiencia del visitante y facilitando las labores de los empleados del centro, aumentando su productividad.
- Construcción y entrenamiento del modelo de inteligencia artificial enfocado en responder las preguntas más frecuentes que los visitantes suelen hacer cuando visitan el CIDIS, haciendo uso de datos recolectados que reflejaban las preguntas y sus respectivas respuestas, mejorando la capacidad de respuesta del AR.
- Diseño de un sistema con diferentes módulos que se comunican de manera fluida y eficiente, permitiendo que el proceso de reconocimiento y síntesis de voz se ejecuten en los mejores tiempos posibles, logrando que el AR funcione en tiempo real.
- Implementación de una interfaz de usuario que cuenta con un personaje 3D, el cual se mueve de acuerdo a la respuesta obtenida del modelo, lo que permite que la interacción con el visitante sea más amigable y atractiva dentro del entorno de oficina.

4.1.2 Recomendaciones

A continuación, se detallará algunas recomendaciones y consideraciones a tener en cuenta para futuras iteraciones:

- Para el módulo de reconocimiento de voz, se recomienda analizar e implementar mecanismos que faciliten la comunicación entre el AR y el visitante, si este último presenta dificultades para comunicarse de manera verbal, ya sea por enfermedad o por problemas congénitos.
- Para el módulo de reconocimiento de voz, se debería considerar entrenar el modelo de tal manera que el AR pueda hacer uso de varios idiomas con el objetivo de que pueda comunicarse con el visitante, si este último llegará a comunicarse en otro dialecto.
- Se recomienda implementar en la interfaz gráfica una función que permita mostrar imágenes de los diferentes lugares, o puntos de referencia, que el modelo va mencionando mientras le da las indicaciones al visitante.
- Se puede considerar agregar un modelo de reconocimiento facial, con el objetivo de poder ofrecer un saludo personalizado a cada uno de los empleados del centro, así como poder tomarle una foto al visitante y poder asociarlo a una entrada del registro que se lleva.

Referencias

[1] S. J. Russell and P. Norvig, Artificial intelligence: a modern approach. Upper Saddle River: Pearson, 2016.

[2] BBC News Mundo, “GPT-4: qué novedades presenta la nueva versión del chat de inteligencia artificial,” BBC News Mundo, 2023. [Online] Disponible en: https://www.bbc.com/mundo/noticias-64969661

[3] BBC News Mundo, “Así funciona BERT, la mayor actualización del algoritmo de búsqueda de Google en años que toma en cuenta el sentido de las frases,” BBC News Mundo, 2019. [Online] Disponible en: https://www.bbc.com/mundo/noticias-50223408

[4] News Center Microsoft Latinoamérica, “Actualizaciones clave para Bing y el poder de la IA en Windows 11,” News Center Latinoamérica, 2023. [Online]. Disponible en: https://news.microsoft.com/es-xl/microsoft-anuncia-actualizaciones-clave-para-bing-y-lleva-el-poder-de-la-ia-a-windows-11/

[5] W. Condori Quispe, “Desarrollo de un Asistente Virtual Utilizando Facebook Messenger para la Mejora del Servicio de Atención al Cliente en la Universidad Privada de Tacna en el 2017,” Universidad Privada de Tacna, 2017. Disponible en: https://repositorio.upt.edu.pe/handle/20.500.12969/163

[6] S. Jusoh, “Intelligent Conversational Agent for Online Sales” 2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 2018. doi: https://doi.org/10.1109/ecai.2018.8679045.

[7] M. Adam, M. Wessel, and A. Benlian, “AI-based chatbots in customer service and their effects on user compliance,” Electronic Markets, vol. 31, no. 2, 2020. doi: https://doi.org/10.1007/s12525-020-00414-7.

[8] F. Garibay, “Diseño e implementación de un asistente virtual (chatbot) para ofrecer atención a los clientes de una aerolínea mexicana por medio de sus canales conversacionales”, 2020. Disponible en: https://infotec.repositorioinstitucional.mx/jspui/bitstream/1027/402/1/INFOTEC_MGITIC_FAGO_27082020.pdf

[9] R. Bonilla, “Prototipo de Chatbot para la Resolución y Atención de Inquietudes Académicas de la Secretaría de Ingeniería en Sistemas Computacionales e Informáticos”, Tesis de Grado, Facult. De Ing. Sistemas, Electrónica e Industrial, Univ. Tec. De Ambato, Ambato, Ecuador, 2021.

[10] J. He and C. Xin, “Developing an AI-Powered Chatbot to Support the Administration of Middle and High School Cybersecurity Camps,” Journal of Cybersecurity Education, Research and Practice, vol. 2021, no. 1, 2021. Disponible en: https://files.eric.ed.gov/fulltext/EJ1338316.pdf

[11] E. Ruane, A. Birhane, and A. Ventresque, “Conversational AI: Social and Ethical Considerations,” 2019. Disponible en: https://ceur-ws.org/Vol-2563/aics_12.pdf

[12] IBM, “¿Qué es la IA conversacional? | IBM,” www.ibm.com. [Online] Disponible en: https://www.ibm.com/es-es/topics/conversational-ai

[13] D. Schnelle-Walka, S. Radomski, B. Milde, C. Biemann, and M. Mühlhäuser, “NLU vs. Dialog Management: To Whom am I Speaking?,” 2016. doi: https://doi.org/10.13140/RG.2.1.1928.4247.

[14] P. A. Angga, W. E. Fachri, A. Elevanita, Suryadi, and R. D. Agushinta, “Design of chatbot with 3D avatar, voice interface, and facial expression,” IEEE Xplore, 2015. Disponible en: https://ieeexplore.ieee.org/abstract/document/7407826

[15] J. Gómez Guinovart, “Fundamentos de Lingüística Computacional: bases teóricas, líneas de investigación y aplicaciones”, core.ac.uk, 1998. Disponible en: https://core.ac.uk/reader/38981350

[16] A. Jiménez Godínez, “Adaptación de Chatbots a Avatares Virtuales con MetaHuman Animación automática de un Avatar Virtual en Unreal Engine”, UNIVERSIDAD DE MÁLAGA, 2022. Disponible en: https://riuma.uma.es/xmlui/bitstream/handle/10630/23847/Jim%C3%A9nez%20God%C3%ADnez%20Antonio%20Memoria.pdf?sequence=1&isAllowed=y

[17] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and Ilya Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision”, 2022. doi: https://doi.org/10.48550/arxiv.2212.04356.

[18] Suno-AI, “Bark - TTS 0.16.3 documentation,” *tts.readthedocs.io,* 2020. https://tts.readthedocs.io/en/dev/models/bark.html.

[19] Rasa Technologies Inc, “Open source conversational AI,” Rasa, 2022. Disponible en: https://rasa.community

[20] Ready Player Me, “https://readyplayer.me,” Ready Player Me. Disponible en: https://readyplayer.me

[21] Blender, “Blender - About,” Blender, 2019. Disponible en: https://www.blender.org

[22] Open AI Forum, “Whisper hallucination - how to recognize and solve?,” OpenAI Developer Forum, 2023. Disponible en: https://community.openai.com/t/whisper-hallucination-how-to-recognize-and-solve/218307

[23] ReadyPlayerMe, “Game-Ready Animation Library by Ready Player Me,” GitHub, 2023. Disponible en: https://github.com/readyplayerme/animation-library

[24] BabylonJS, “Home | Babylon.js Documentation,” doc.babylonjs.com. Disponible en: https://doc.babylonjs.com

[25] Coqui, “Coqui TTS,” GitHub, 2021. Disponible en: https://github.com/coqui-ai/TTS

Apéndices

Apéndice A

Cuestionario de evaluación de usabilidad y efectividad del AR

Información general

1. Edad
2. ¿Es usted estudiante o persona ajena a la ESPOL?
   a) Estudiante
   b) Persona ajena de ESPOL
3. ¿Cuál es su nivel de familiaridad con la tecnología?
   a) Bajo
   b) Medio
   c) Alto

Preguntas acerca del AR

Indique su nivel de acuerdo con las siguientes afirmaciones, utilizando la siguiente escala:

1	Totalmente en desacuerdo
2	En desacuerdo
3	Ni de acuerdo ni en desacuerdo
4	De acuerdo
5	Totalmente de acuerdo

1. La interfaz del AR fue fácil de usar
2. Pude interactuar con el AR sin necesidad de ayuda
3. El personaje mostrado en el monitor era agradable a la vista
4. El AR reconoció la mayoría de las preguntas que hice
5. Tuve problemas para iniciar la interacción con el AR
6. Me fue fácil hacer preguntas sobre el centro y trabajadores del CIDIS
7. Me fue fácil entender las indicaciones para llegar a distintas partes de la universidad

Preguntas abiertas acerca del AR

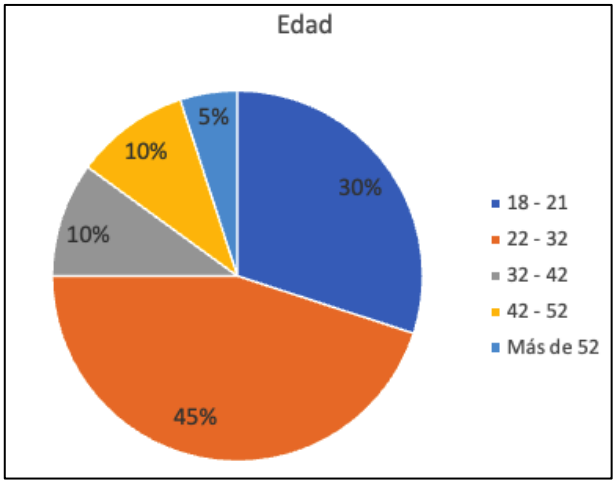
1. ¿Cuáles fueron los aspectos que más le gustaron de la experiencia de uso del AR?
2. ¿Hubo algún aspecto que no le gustó o le pareció difícil de usar?
3. ¿Experimentó algún tipo de retraso en la respuesta del AR?
4. ¿En algún momento se sintió frustrado durante la interacción con el AR?

Apéndice B

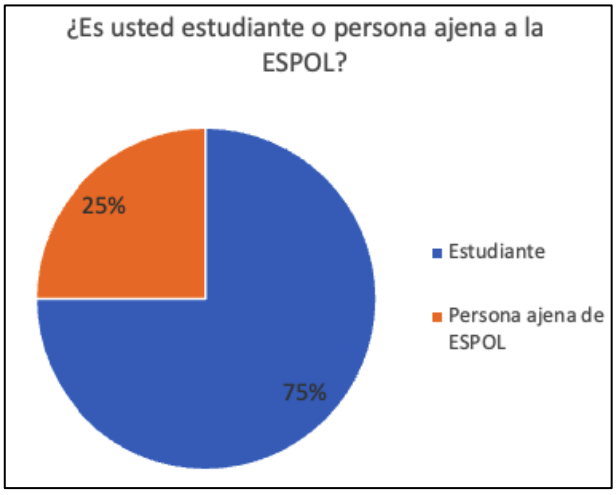
Respuestas del cuestionario de evaluación de usabilidad y efectividad del AR

Información general

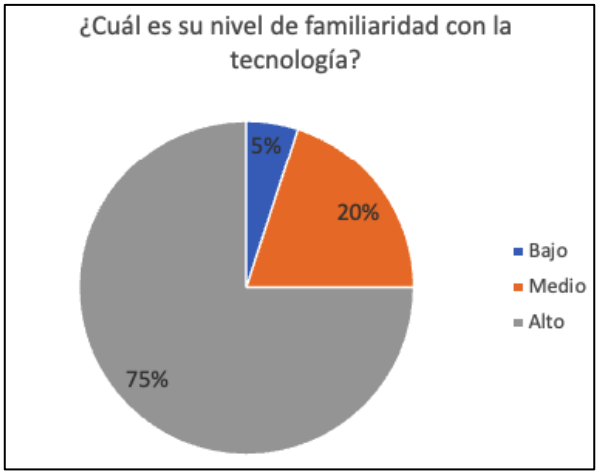
1. Edad



2. ¿Es usted estudiante o persona ajena a la ESPOL?

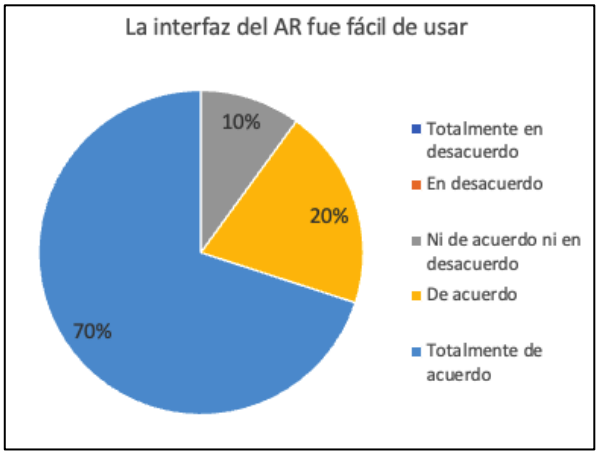


3. ¿Cuál es su nivel de familiaridad con la tecnología?

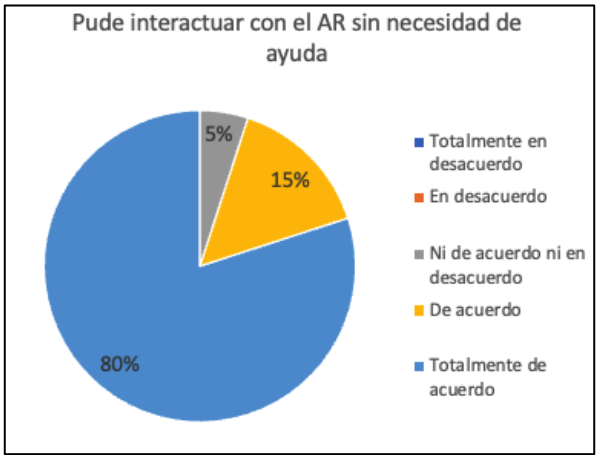


Preguntas acerca del AR

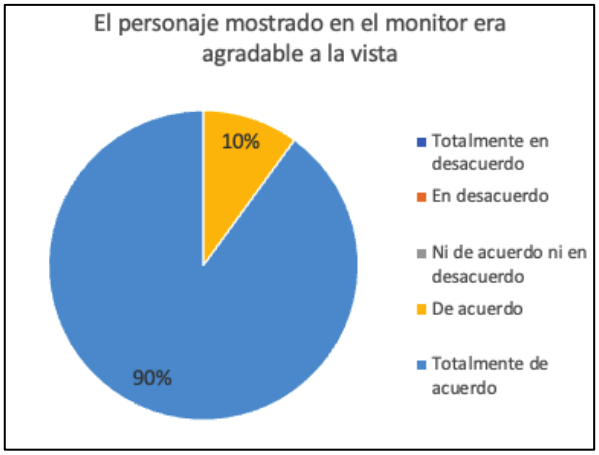
1. La interfaz del AR fue fácil de usar



2. Pude interactuar con el AR sin necesidad de ayuda

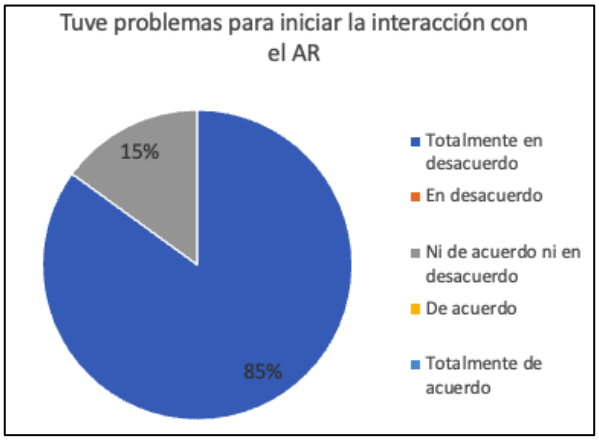


3. El personaje mostrado en el monitor era agradable a la vista

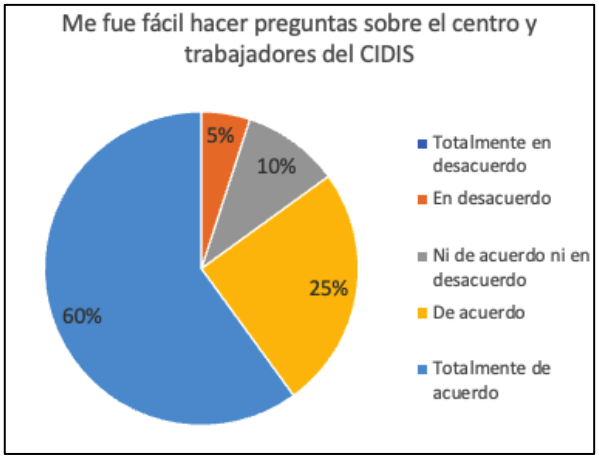


4. El AR reconoció la mayoría de las preguntas que hice

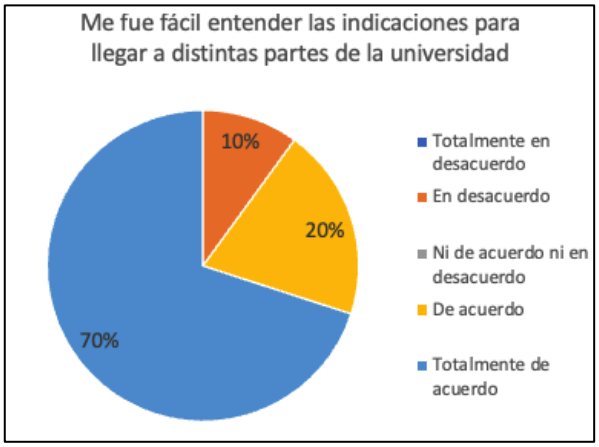
5. Tuve problemas para iniciar la interacción con el AR



6. Me fue fácil hacer preguntas sobre el centro y trabajadores del CIDIS



7. Me fue fácil entender las indicaciones para llegar a distintas partes de la universidad



Apéndice C

Acta de Entrega del Proyecto

Acta de Entrega de Proyecto de Agente Recepcionista basado en Inteligencia Artificial

En la ciudad de Guayaquil, el día 24 de agosto de 2023, se reúnen en las instalaciones del CIDIS los siguientes representantes de las partes involucradas en el proyecto del Agente Recepcionista basado en Inteligencia Artificial, con el fin de formalizar y dar por concluida la entrega del proyecto de acuerdo a los términos y condiciones previamente establecidos:

Cliente	Dennys Paillacho
Equipo de desarrollo	Luis Sánchez Fiorella Yerovi

Descripción del Proyecto

El proyecto consistió en el diseño, desarrollo e implementación de un sistema de recepción basado en inteligencia artificial que actúa como agente recepcionista virtual. El sistema ha pasado por diversas etapas, incluyendo la planificación, diseño de la arquitectura, desarrollo del código fuente y pruebas exhaustivas.

Entregables

En esta reunión se procede a realizar la entrega formal del proyecto, entregando los siguientes elementos:

- **Código Fuente Completo del Agente Recepcionista:**
  Se entrega el código fuente del sistema de agente recepcionista basado en inteligencia artificial en su versión final.
- **Datos de Entrenamiento para el Modelo Utilizados:**
  Se entregan los conjuntos de datos utilizados para entrenar el modelo de inteligencia artificial del agente recepcionista. Estos datos son fundamentales para comprender cómo se ha entrenado el modelo y para posibles futuras mejoras o ajustes.
- **Guía de Usuario e implementación**
  Se proporciona una guía de usuario detallada que explica cómo interactuar con el agente recepcionista. Además se proporciona una guía de implementación, para poder volver a implementar este mismo agente recepcionista en caso de pérdida del sistema.

Satisfacción de las Partes Involucradas

Todas las partes involucradas en el proyecto, tanto los desarrolladores como el cliente, expresan su satisfacción con el resultado final del proyecto.

Firma de Conformidad

En constancia de que los entregables y términos del proyecto han sido aceptados y están en conformidad, el cliente firma el presente acta.

_____

Dennys Paillacho

Con la firma de este acta, se da por concluida la entrega del proyecto del Agente Recepcionista basado en Inteligencia Artificial en completa satisfacción de ambas partes involucradas.

Apéndice D

Guía de Implementación

Esta guía proporciona un conjunto de pasos detallados para la implementación del agente recepcionista con inteligencia artificial. El proyecto combina módulos de Python y JavaScript para crear una solución integral.

Para poder realizar la correcta implementación del AR, el cual fue desarrollado como parte de la solución propuesta para la problemática que afectaba al CIDIS, se deben cumplir ciertos requisitos y especificaciones.

Requisitos de Hardware:

- Memoria RAM de 4GB o superior.
- Espacio en disco de 5GB o superior.

Requisitos de Software:

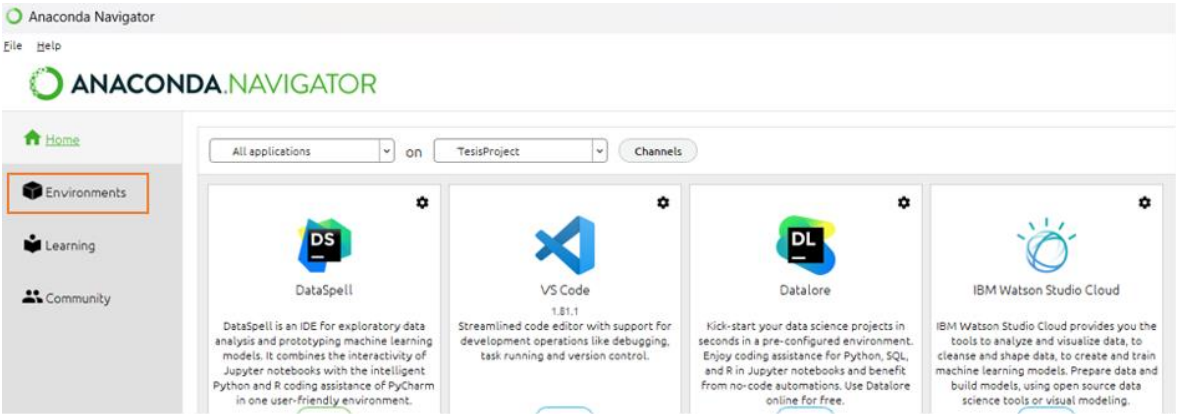
- Python Versión 3.10.11.
- Pip Versión 23.3.1.
- Anaconda Versión 23.3.1.
- Docker Versión 20.10.17.
- NodeJS 18.17.1.

Pasos para la implementación de los módulos desarrollados en Python

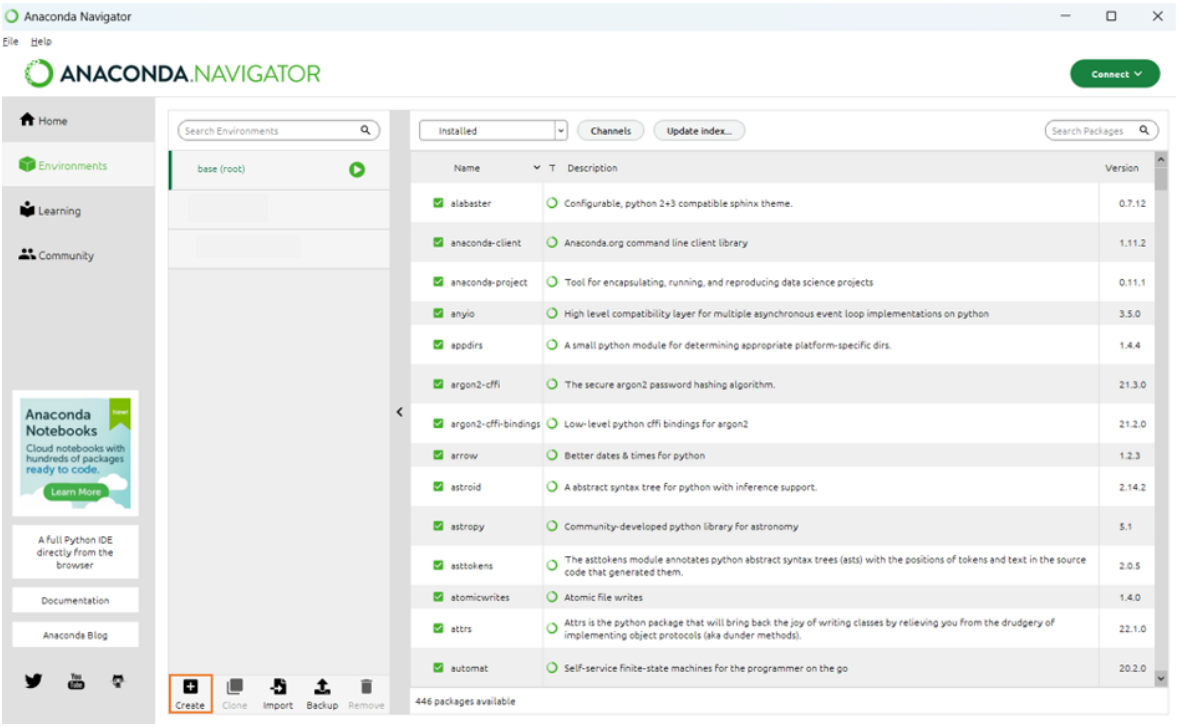
1. Solicitar acceso, de ser necesario, al repositorio del proyecto dado que es un repositorio privado.
2. Clonar el repositorio el cual se encuentra alojado en el siguiente link: https://github.com/lcsanche/TesisProject.git
3. Ingresar a la carpeta que contiene el proyecto clonado.
4. Crear un ambiente virtual para ejecutar el proyecto. A continuación, se presentan dos maneras de hacerlo:

Usando Anaconda

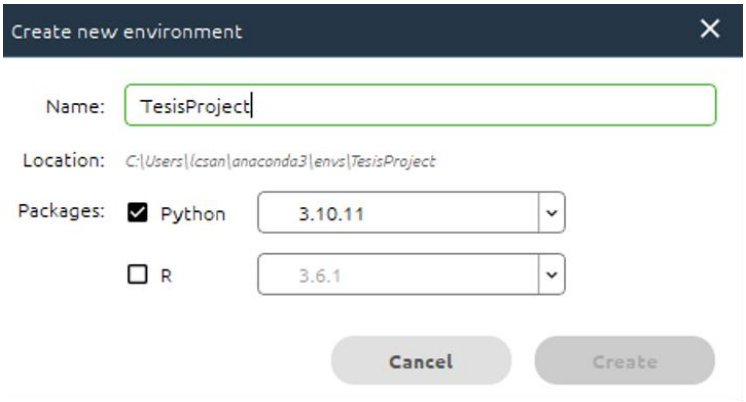
Abrir Anaconda Navigator y en el menú lateral seleccionar la opción “Environments” en el menú lateral.



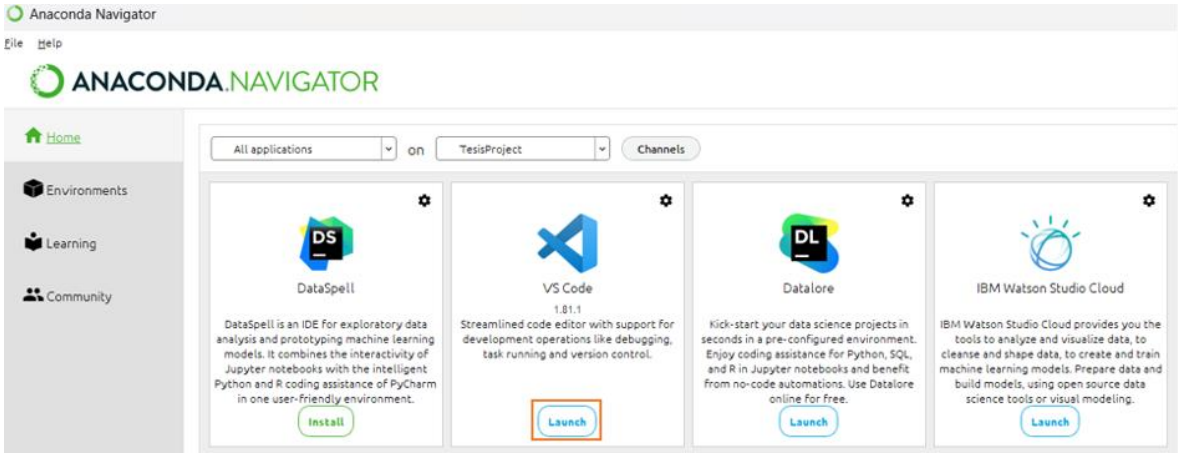
Seleccionar la opción “Create”.



En la ventana emergente, escribir el nombre que tendrá el ambiente virtual y seleccionar la versión de Python especificada anteriormente, posterior dar clic en "Create".



Una vez terminada la creación del ambiente virtual, volver a la pestaña “Home” y abrir Visual Studio Code.

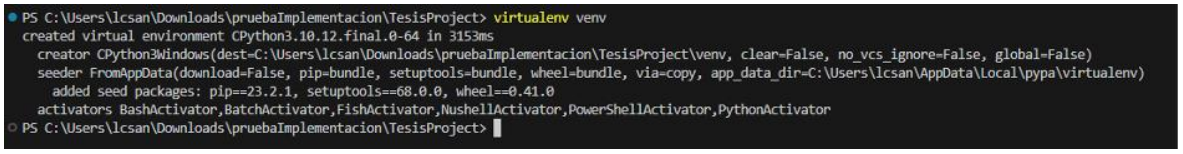


Una vez iniciado VS Code, abrir la carpeta donde se encuentra el proyecto clonado.

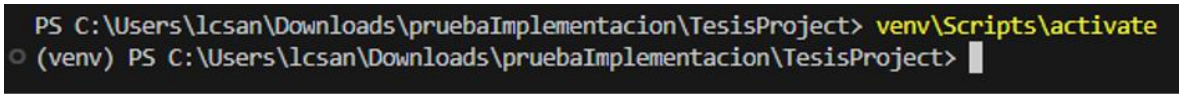
Usando VirtualEnv

Abrir la consola de comandos y ejecutar el comando “pip install virtualenv”.

Crear el ambiente virtual haciendo uso del comando “virtualenv venv”.

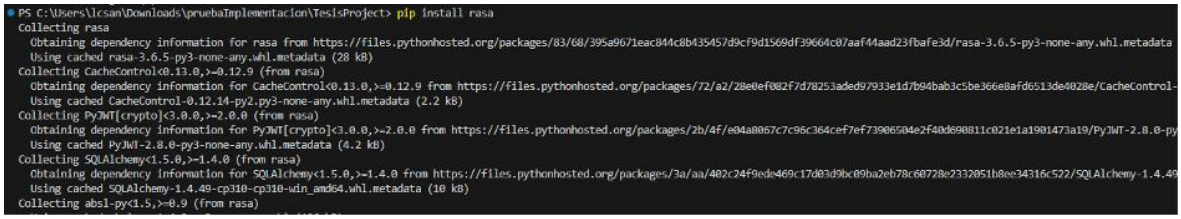


Activar el ambiente virtual por medio del comando “venv\Scripts\activate”.*



En caso de usar Linux utilizar el comando “source venv/bin/activate”

5. Una vez creado el ambiente virtual, acceder a la terminal de comandos y se procederá a instalar las dependencias del proyecto. Primero comenzaremos instalando Rasa.



6. Posterior a esto, se procederá a instalar la librería Spacy.

7. Por último, se procederá a instalar el resto de las dependencias del proyecto.*

```
PS C:\Users\lcsan\Downloads\pruebaImplementacion\TesisProject> pip install -r requirements.txt
Collecting es-core-news-md@ https://github.com/explosion/spacy-models/releases/download/es_core_news_md-3.6.0/es_core_news_md-3.6.0-py3-none-any.whl#sha256=b3d69542
  Downloading https://github.com/explosion/spacy-models/releases/download/es_core_news_md-3.6.0/es_core_news_md-3.6.0-py3-none-any.whl (42.3 MB)
     ---------------------------------------- 42.3/42.3 MB 9.2 MB/s eta 0:00:00
Requirement already satisfied: absl-py==1.4.0 in c:\users\lcsan\anaconda3\envs\prueba11\lib\site-packages (from -r requirements.txt (line 1)) (1.4.0)
Requirement already satisfied: aio-pika==8.2.3 in c:\users\lcsan\anaconda3\envs\prueba11\lib\site-packages (from -r requirements.txt (line 2)) (8.2.3)
Requirement already satisfied: aiofiles==23.2.1 in c:\users\lcsan\anaconda3\envs\prueba11\lib\site-packages (from -r requirements.txt (line 3)) (23.2.1)
Requirement already satisfied: aiogram==2.25.1 in c:\users\lcsan\anaconda3\envs\prueba11\lib\site-packages (from -r requirements.txt (line 4)) (2.25.1)
Requirement already satisfied: aiohttp==3.8.5 in c:\users\lcsan\anaconda3\envs\prueba11\lib\site-packages (from -r requirements.txt (line 5)) (3.8.5)
Requirement already satisfied: aiohttp-retry==2.8.3 in c:\users\lcsan\anaconda3\envs\prueba11\lib\site-packages (from -r requirements.txt (line 6)) (2.8.3)
```

En caso de usar Linux antes de instalar el resto de las dependencias instalar la librería portaudio con el siguiente comando: "sudo apt-get install python3-pyaudio"

Pasos para la implementación del módulo desarrollado en JavaScript:

1. Solicitar acceso, de ser necesario, al repositorio del proyecto dado que es un repositorio privado.
2. Clonar el repositorio el cual se encuentra alojado en el siguiente link: https://github.com/fioyerovi/tesisCIDISModelo.git
3. Ingresar a la carpeta que contiene el proyecto clonado.
4. En el terminal correr "npm install"
5. Ejecutar el archivo websocket-server.js con "node websocket-server.js"

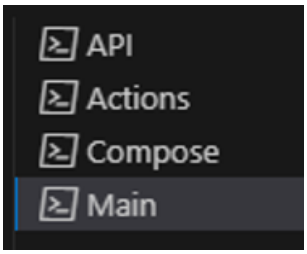
```
PS C:\Users\fioye\Downloads\Modelo3dRecepcionista\modelo3DTesis> node .\websocket-server.js
WebSocket server is listening on port 5501.
Client connected.
```

6. Ejecutar el proyecto con "node serverBabylon.js"

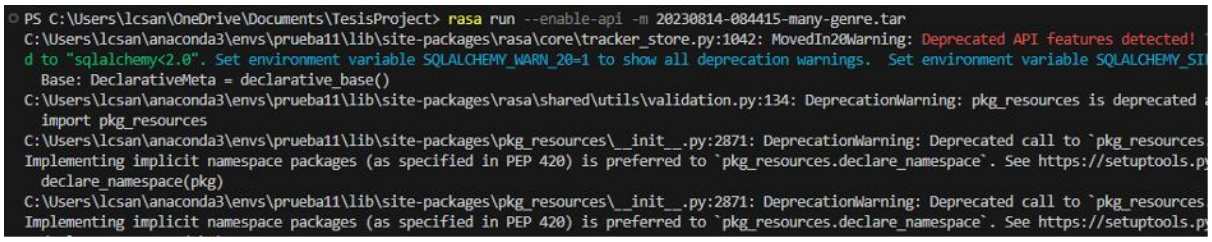
```
PS C:\Users\fioye\Downloads\Modelo3dRecepcionista\modelo3DTesis> node .\serverBabylon.js
Server is running on http://localhost:3000
```

Pasos para la ejecución del proyecto:

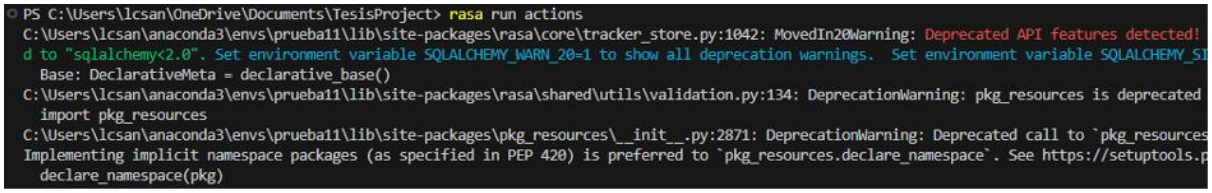
1. Para ejecutar el proyecto, es necesario abrir 4 terminales, cada terminal cumplirá una función en específico.



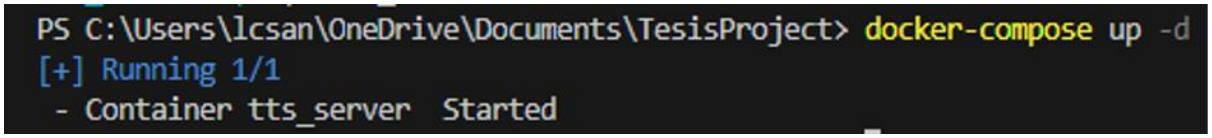
2. En la primera terminal denominada “API”, se ejecutará el modelo de Rasa previamente entrenado activando el modo API. Es importante añadir el nombre del modelo que se va a usar.



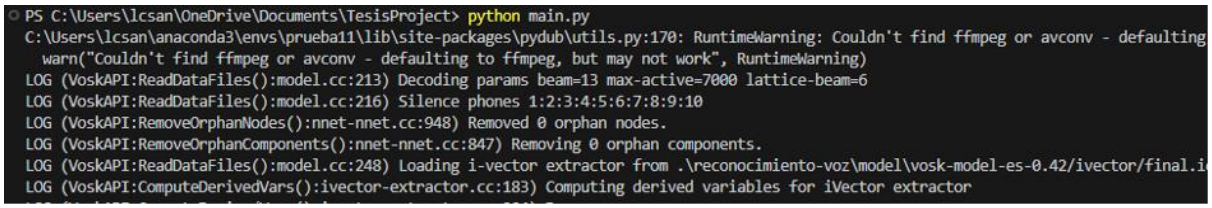
3. En la segunda terminal denominada “Actions”, se ejecutará el servidor con las acciones personalizadas.



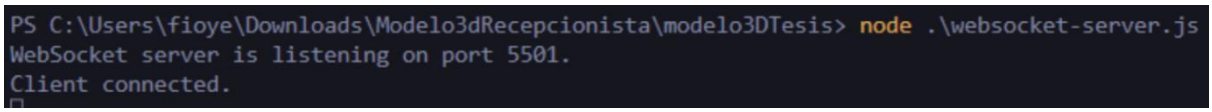
4. En la tercera denominada “Compose”, la usaremos para levantar el contenedor de Docker con el modelo TTS.



5. Por último, en la cuarta terminal denominada “Main”, se ejecutará el proyecto principal.



Al haber ejecutado estos 5 pasos para ejecutar el proyecto deberá dirigirse al terminal del websocket para verificar que se haya conectado tanto el modelo 3D como los módulos de Python. Esto lo podrá comprobar al ver si existen 2 conexiones online al websocket.



Apéndice E

Guía de usuario

Esta guía te ayudará a comprender cómo son las interacciones con el agente y aprovechar al máximo sus capacidades para gestionar tus necesidades de manera eficiente.

1. Introducción

¿Qué es el agente recepcionista con Inteligencia Artificial?

Nuestro agente recepcionista es una herramienta basada en inteligencia artificial que te permite interactuar de manera natural para realizar preguntas y obtener respuestas a diversas consultas acerca del CIDIS y sus trabajadores. Utiliza tecnologías avanzadas de procesamiento de lenguaje natural para comprender y responder a tus solicitudes.

Beneficios de usar el agente recepcionista

- *Accesibilidad 24/7:* El agente está disponible en todo momento para atender tus necesidades.
- *Respuestas Rápidas:* Obtén respuestas instantáneas a tus preguntas y solicitudes.
- *Interacción Natural:* Puedes comunicarte en lenguaje natural, como si estuvieras hablando con una persona real.

2. Interacción con el Agente

Acceso a la Interfaz de Usuario

Para comenzar, en el navegador web de la computadora del CIDIS en http://localhost:3000 estará el agente recepcionista. El cual podrá ser visualizado por los visitantes al entrar a las oficinas.

Realizar Preguntas y Solicitudes

Comienza la interacción con el agente recepcionista diciéndole Hola y te preguntará por tu nombre para comenzar. Luego podrás realizarle cualquier tipo de pregunta del siguiente estilo relacionado al CIDIS, sus trabajadores y cómo llegar a ciertos centros de la ESPOL.

Preguntas acerca del CIDIS	¿Qué es el CIDIS? ¿Cuál es la misión/visión del CIDIS?
Preguntas acerca de los trabajadores del CIDIS	¿Cuál es el correo de *Dennys Paillacho*? ¿Cuál es el número de teléfono de *Dennys Paillacho*?
Preguntas acerca de la disponibilidad de un trabajador del CIDIS	¿Está *Dennys Paillacho* en la oficina?
Preguntas de la ubicación de ciertos centros de la ESPOL	¿Cómo puedo llegar a rectorado?

3. Consejos para una Mejor Experiencia

Expresión Clara de Preguntas

Para obtener respuestas precisas, formula tus preguntas de manera clara y concisa. Evita la ambigüedad para recibir la información que necesitas.

4. Preguntas Frecuentes

¿Qué tipo de preguntas puedo hacer?

Puedes hacer una amplia variedad de preguntas, como obtener información sobre el CIDIS, sus trabajadores y de ubicaciones de distintos centros de ESPOL. El agente está aquí para ayudarte con cualquier pregunta que tengas.

¿El Agente guarda mis datos?

Sí, el agente guarda información como tu nombre y hora en la que comenzaste a interactuar con el mismo. Sin embargo, tu privacidad es importante para nosotros, y tus datos serán tratados con confidencialidad y de acuerdo con nuestras políticas de privacidad.

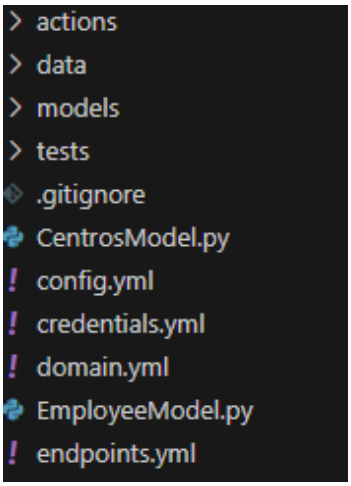
Apéndice F

Guía de entrenamiento de Rasa

Esta guía proporciona una explicación de los pasas que se deben de seguir para poder reentrenar el modelo de Rasa, ya sea para agregarle más ejemplos de entrenamiento o agregarle nuevas funcionalidades.

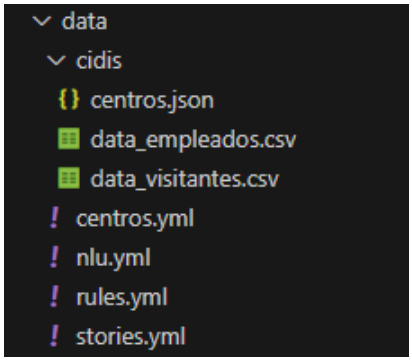
Estructura de las carpetas:

A continuación, se muestra cuáles son las carpetas que conforman el proyecto de rasa.



Carpeta Actions: En esta carpeta se encuentra el archivo actions.py, el cual contiene las clases a las que Rasa accederá cuando quiera ejecutar una acción personalizada.

Carpeta Data: En esta carpeta se encuentran los archivos que son indispensables para el entrenamiento del modelo, la estructura de la carpeta es la siguiente:



Dentro de la carpeta “cidis” se encuentra los archivos que contienen información sobre el cidis, información la cual será usada por los CustomActions para responder a las preguntas. A continuación, se detallarán cual es la función de los demás archivos.

- **centros.yml:** Este archivo contiene una lista de los nombres de los diferentes centros de la ESPOL, los cuales serán mapeados a la “entity” denominada: centros_espol, la cual es usada en los ejemplos de entrenamiento.
- **nlu.yml:** Este archivo contiene los diferentes “intents”, nombre con el que se denomina a las situaciones se abordarán; cada “intent” contiene diferentes ejemplos de preguntas, algunas de ellas contienen una referencia a un “entity” en particular.
- **rules.yml:** Este archivo contiene las reglas, es decir ciertos eventos que deberán ocurrir cuando suceda algo en particular, por ejemplo: el AR deberá saludar cada vez que el visitante diga hola.
- **stories.yml:** Este archivo contiene las “stories”, aquí se detalla el flujo que tendrá la conversación con el AR, de tal manera que, durante el entrenamiento el modelo pueda aprender dicho flujo y así pueda identificar de manera más fácil cuales son las preguntas por responder.

Carpeta Models: En esta carpeta se guarda el modelo ya entrenado.

Archivo Config.yml: En este archivo se detalla cuáles serán los pipelines a usar durante el entrenamiento.

Archivo Domain.yml: En este archivo se detallará todo lo mencionado anteriormente, se deberá especificar cuáles serán las entities, los intents y los actions a usar.

Agregando funcionalidades:

En caso de que se quiera entrenar al modelo con una nueva pregunta o funcionalidad, se deberá seguir los siguientes pasos:

1. Ir al archivo nlu.yml y crear un nuevo intent, el nombre del intent deberá ser único, y luego agregar las preguntas de ejemplo, siguiendo esta estructura:

```
- intent: proyectos_cidis
  examples: |
    - Cuales son los proyectos que hace el cidis
    - Que tipo de proyectos realiza el CIDIS
    - Cuales son los enfoques que tienen los proyectos del Cidis
    - Que clase de proyectos realiza el ciris
    - que tipo de proyectos ha realizado el Civis
```

   Nota: Si la nueva pregunta hará uso de una "entity", es necesario crear una tabla "lookup" en el mismo archivo, la cual contendrá los ejemplos de la "entity", la estructura es la siguiente:

```
- lookup: areas_cidis
  examples: |
    - Vision por Computador
    - Robotica de Servicio y de Campo
    - Aprendizaje de Maquinas
    - Control Avanzado de Sistemas de Maquinas
```

   Luego referenciar dicha "entity" en las preguntas de ejemplo del "intent", como lo muestra la siguiente imagen:

```
- intent: coordinador_area_cidis
  examples: |
    - Quien es el coordinador del area de [Vision por Computador](areas_cidis)
    - Quien lidera el area de [Aprendizaje de Maquinas](areas_cidis)
    - me puedes decir quien es el coordinador del area de [Robotica de Servicio y de Campo]
    - En el area de [Control Avanzado de Sistemas de Maquinas](areas_cidis), quien es el co
    - quien es el lider del departamento de [Robotica de Servicio y de Campo](areas_cidis)
```

2. Ir al archivo stories.yml y crear una nueva “story” con los pasos que se deberán seguir para poder responder la pregunta, siguiendo la siguiente estructura:

```yaml
- story: ruta_coordinador_area_cidis
  steps:
  - intent: coordinador_area_cidis
    entities:
      - areas_cidis: "Robotica de Servicio y de Campo"
  - action: action_find_coordinator_area
```

3. En caso de querer crear un custom_actions, deberás ir al archivo actions.py de la carpeta actions y en ella establecer una nueva clase, la cual representará tu custom_action, para ello deberás seguir la siguiente estructura:

```python
class CustomAction(Action):
    def name(self) -> Text:
        return "name_action"

    def run(
        self,
        dispatcher: CollectingDispatcher,
        tracker: Tracker,
        omain: Dict[Text, Any]
    ) -> List[Dict[Text, Any]]:
        # Codigo
        return
```

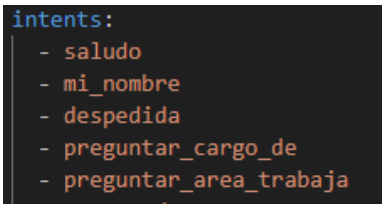
   Donde, la función “name” define el nombre de la acción, mismo nombre por el cual Rasa identificará la accion; y la función “run” define cual es el código que se ejecutará al ser llamada.

4. Ir al archivo domain.yml y declarar todos los intent, entity y actions creadas dentro de los respectivos campos.

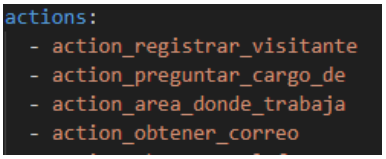
   Para declarar un “entity”, añadirlo a esta sección:

```yaml
entities:
  - empleado_name
  - mision_vision
  - areas_cidis
  - centro_espol
```

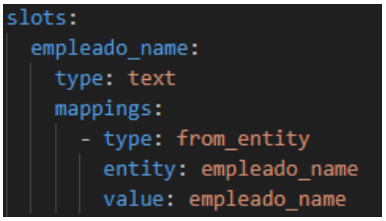
Para declarar un “intent”, añadirlo a esta sección:



Para declarar un “actions”, añadirlo a esta sección:



Nota: Es importante declarar las entities dentro de los slots:



Entrenar el modelo:

Para entrenar al modelo solo es necesario abrir una consola de comandos dentro de la carpeta y ejecutar el comando: “rasa train”