

La investigación reportada en esta tesis es parte de los programas de investigación del CICESE (Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California).

La investigación fue financiada por el CONAHCYT (Consejo Nacional de Humanidades, Ciencias y Tecnologías).

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México). El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

CICESE © 2023, Todos los Derechos Reservados, CICESE

Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California



Maestría en Ciencias en Ciencias de la Computación

Detección de comportamiento no verbal en interacción humano-robot

Tesis

para cubrir parcialmente los requisitos necesarios para obtener el grado de
Maestro en Ciencias

Presenta:

Ernesto Adrián Lozano De la Parra

Ensenada, Baja California, México

2023

Tesis defendida por

Ernesto Adrián Lozano De la Parra

y aprobada por el siguiente Comité

Dr. Irvin Hussein López Nava

Codirector de tesis

Dr. Jesús Favela Vara

Codirector de tesis

Dr. Ubaldo Ruiz López

Dr. Miguel Ángel Alonso Arévalo



Dr. Pedro Gilberto López Mariscal

Coordinador del Posgrado en Ciencias de la Computación

Dra. Ana Denise Re Araujo

Directora de Estudios de Posgrado

Resumen de la tesis que presenta Ernesto Adrián Lozano De la Parra como requisito parcial para la obtención del grado de Maestro en Ciencias en Ciencias de la Computación.

Detección de comportamiento no verbal en interacción humano-robot

Resumen aprobado por:

Dr. Irvin Hussein López Nava

Codirector de tesis

Dr. Jesús Favela Vara

Codirector de tesis

La comunicación no verbal desempeña un papel vital en la interacción humana. En el contexto de la interacción humano-robot (IHR), los robots sociales están diseñados principalmente para la comunicación verbal con los humanos, dejando a la comunicación no verbal como un área de investigación abierta. En este trabajo, se presenta una arquitectura flexible y abierta llamada *Software Architecture for Nonverbal Interaction in Human-Robot Interaction* (SANI-HRI) diseñada para facilitar las interacciones no verbales en IHR. Entre sus componentes se encuentra un Cuaderno Computacional P2P basado en navegador web, aprovechado para codificar, ejecutar y compartir programas reactivos. Pueden incluirse modelos de aprendizaje automático para el reconocimiento en tiempo real de gestos, poses y estados de ánimo, empleando protocolos como MQTT. Otro componente clave es un Broker para distribuir datos entre distintos dispositivos físicos, como robots, dispositivos vestibles y sensores ambientales, así como modelos de aprendizaje automático que comprendan diferentes tipos de datos. Se demuestra la utilidad de esta arquitectura mediante tres escenarios de interacción: (i) el primero que emplea la proxémica y la dirección de la mirada para iniciar un encuentro improvisado, (ii) un segundo que utiliza técnicas de visión por computadora para detectar y analizar expresiones faciales y corporales, así como el uso sensores biométricos para obtener datos de ritmo cardíaco durante una rutina de ejercicio, y (iii) un tercero que incorpora el reconocimiento de objetos y Modelos de Lenguaje Grandes para sugerir comidas a cocinar en función de los ingredientes disponibles. Estos escenarios ilustran cómo los componentes de la arquitectura pueden integrarse para abordar nuevos escenarios, en los que los robots necesitan inferir señales no verbales de los usuarios.

Palabras clave: Interacción humano-robot, Comunicación no verbal, Broker MQTT, Notebook computacional, Modelos lingüísticos grandes, SANI-HRI.

Abstract of the thesis presented by Ernesto Adrián Lozano De la Parra as a partial requirement to obtain the Master of Science degree in Computer's Science.

Detection of non-verbal behavior in human-robot interaction

Abstract approved by:

PhD Irvin Hussein López Nava

Thesis Co-Director

PhD Jesús Favela Vara

Thesis Co-Director

Nonverbal communication plays a vital role in human interaction. In the context of Human-Robot Interaction (HRI), social robots are designed primarily for verbal-based communication with humans, making nonverbal communication an open research area. We present a flexible, open framework called *Software Architecture for Nonverbal Interaction in Human-Robot Interaction* (SANI-HRI) designed to facilitate nonverbal interactions in HRI. Among its components it has a P2P Browser-Based Computational Notebook, leveraged to code, run, and share reactive programs. Machine-learning models can be included for real-time recognition of gestures, poses, and moods, employing protocols such as MQTT. Another key component is a broker for distributing data among different physical devices like the robot, wearables, and environmental sensors and also machine learning models. We demonstrate this framework's utility through three interaction scenarios: (i) the first one employing proxemics and gaze direction to initiate an impromptu encounter, (ii) a second that uses computer vision techniques to detect and analyze facial and body expressions, as well as the use of biometric sensors to obtain heart rate data during a workout routine, and (iii) a third one incorporating object recognition and a Large-Language Model to suggest meals to be cooked based on available ingredients. These scenarios illustrate how the framework's components can be seamlessly integrated to address new scenarios, where robots need to infer nonverbal cues from users.

Keywords: Human-robot interaction, Nonverbal communication, Broker MQTT, Computational notebook, Large language models, SANI-HRI.

Dedicatoria

A mi madre, por ser el pilar más importante y por demostrarme siempre su cariño y apoyo incondicional. A mi padre, por su apoyo en mi maestría sin importar las adversidades que se tuvieron. A mi hermano, por ser un gran ejemplo en el trayecto de mi tesis, un gran amigo en el cual pude contar en cualquier momento. A mis familiares y amigos, que me han apoyado en las buenas y en las malas.

Agradecimientos

Al Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California (CICESE) por la oportunidad de realizar los estudios de posgrado en él, en especial al Departamento de Ciencias de la Computación, que me apoyó durante mis estudios del posgrado, por proporcionarme las instalaciones y los medios necesarios para realizar mi investigación.

Al Consejo Nacional de Humanidades Ciencias y Tecnologías (CONAHCyT) por brindarme el apoyo económico para realizar mis estudios de maestría. No. de becario: 21264931

A mis directores de tesis **Dr. Irvin Hussein López Nava** y **Dr. Jesús Favela Vara** por su esfuerzo y dedicación. Sus conocimientos, sus orientaciones, su manera de trabajar, su persistencia, sobre todo paciencia y su motivación han sido fundamentales para mi formación como investigador. A su manera, han sido capaces de ganarse mi lealtad y admiración, así como sentirme en deuda con ellos por todo lo recibido durante el periodo de tiempo que ha durado la tesis.

A los miembros de mi comité de tesis Dr. Ubaldo Ruiz López y al Dr. Miguel A. Alonso Arévalo por sus comentarios y observaciones que ayudaron a mejorar mi trabajo.

A los profesores del Departamento de Ciencias de la Computación por sus excelentes clases y por el conocimiento transmitido.

A mis compañeros y amigos de Ciencias de la Computación, en especial a Carlos Eduardo Sánchez Torres, por su gran ayuda, esfuerzo y dedicación durante el proceso de desarrollo de mi investigación.

Tabla de contenido

	Página
Resumen en español	ii
Resumen en inglés	iii
Dedicatoria	iv
Agradecimientos	v
Lista de figuras	viii
Lista de tablas	x
 Capítulo 1. Introducción	
1.1. Antecedentes	1
1.2. Planteamiento del problema	2
1.3. Preguntas de investigación	3
1.4. Objetivos	3
1.4.1. Objetivo general	3
1.4.2. Objetivos específicos	4
1.5. Propuesta de solución	4
1.6. Metodología	4
1.7. Contribuciones esperadas	5
1.8. Estructura de la tesis	6
 Capítulo 2. Fundamentos teóricos	
2.1. Arquitectura de software	7
2.2. Robots sociales	8
2.3. Comunicación no verbal	9
2.3.1. Expresiones faciales	10
2.3.2. Gestos	11
2.3.3. Postura y lenguaje corporal	12
2.3.4. Proxémica	13
2.3.5. Seguimiento ocular	14
2.3.6. Señales fisiológicas	14
2.4. Diseño de escenarios	15
2.4.1. Escenario 1: Mirada y proxémica para iniciar una interacción	15
2.4.2. Escenario 2: Rutina de ejercicio	16
2.4.3. Escenario 3: Recomendaciones de recetas de comida	16
2.5. Análisis de los escenarios	17
2.6. Conclusiones del capítulo	18
 Capítulo 3. Trabajo relacionado	
3.1. Robots sociales	20
3.2. Arquitecturas de robots sociales	21
3.3. Comunicación no verbal	22
3.4. Tipos de datos	23
3.5. Técnicas para el reconocimiento de comportamientos no verbales	24

3.5.1.	Seguimiento facial	25
3.5.2.	Parpadeo del ojo	26
3.5.3.	Proxémica	27
3.5.4.	Detección y seguimiento de objetos	28
3.5.5.	Señales fisiológicas y expresiones faciales	29

Capítulo 4. Arquitectura de interacción no verbal SANI-HRI

4.1.	Robot social EVA	31
4.2.	Sensores vestibles y ambientales	33
4.3.	Broker MQTT	34
4.4.	Rhasspy	36
4.5.	EVAnotepad	37
4.6.	Modelos de análisis de datos	40
4.6.1.	MediaPipe	41
4.6.2.	OpenCV	42
4.6.3.	Análisis de procrustes	42
4.6.4.	Whisper	43
4.6.5.	ChatGPT	44
4.6.6.	LangChain	44
4.6.6.1.	Prompts	45
4.6.6.2.	Cadenas	45
4.6.6.3.	Memoria	45
4.7.	Conclusiones de la arquitectura	46

Capítulo 5. Implementación de escenarios

5.1.	Escenarios principales propuestos para la arquitectura	47
5.1.1.	Escenario 1: Mirada y proxémica para iniciar una interacción	47
5.1.2.	Escenario 2: Rutina de ejercicio	51
5.1.3.	Escenario 3: Recetas de comida con ChatGPT	54
5.2.	Escenarios adicionales	56
5.2.1.	Escenario adicional 1: Degustación del té	57
5.2.2.	Escenario adicional 2: Terapia de reminiscencia	58
5.2.3.	Escenario 3: Lengua de señas mexicana	59

Capítulo 6. Conclusiones y trabajo a futuro

6.1.	Conclusiones	62
6.2.	Contribuciones	63
6.3.	Limitaciones	63
6.4.	Trabajo a futuro	64

Literatura citada	66
------------------------------------	----

Anexos	72
-------------------------	----

Lista de figuras

Figura	Página
1. Metodología del trabajo de investigación	4
2. Panorama general de la arquitectura de control del robot	7
3. Ejemplos de robots sociales comerciales con comunicación no verbal. a) Robot Cozmo realizando un gesto de asombro. b) Robot Jibo interactuando mediante reconocimiento facial.	9
4. Tipos de comunicación no verbal. Tomada de (Kumar, 2023)	10
5. Tipos de gestos en la comunicación no verbal.	12
6. Distancias proxémicas (Hans & Hans, 2015).	13
7. Definición de Robot Social por (Hegel et al., 2009)	21
8. Resultados del modelo de seguimiento facial. Tomada de (Soomro et al., 2023).	25
9. Resultados del modelo de parpadeo de ojos. Tomada de (Soomro et al., 2023).	26
10. Controlador proxémico autónomo utilizando un enfoque basado en muestras para estimar cómo los agentes interactuantes experimentarían una interacción social basada en modelos en el marco proxémico socialmente situado; filtro de partículas (con remuestreo) utilizado para maximizar el desempeño social de todos los agentes que interactúan. Tomada de (Mead & Mataric, 2014).	27
11. Detección y seguimiento de objetos usando el robot NAO. Tomada de (Zhou et al., 2018).	28
12. Estimación de expresiones faciales. En la primera etapa, detección de rostros mediante un modelo de aprendizaje profundo convolucional y preprocesamiento del rostro detectado. En la segunda etapa, extracción de características y clasificación mediante un conjunto de modelos de aprendizaje profundo convolucional. Tomada de (Val-Calvo et al., 2020).	30
13. Componentes hardware del robot social EVA.	32
14. Sensores vestibulares y ambientales utilizados en el proceso de la captura de datos durante las interacciones dadas con el robot EVA. a) Sensor portátil Polar H10 para monitorear ritmo cardíaco. b) Cámara Intel RealSense D435i integrada en el robot EVA. c) Arreglo de micrófonos y leds Matrix Voice para el reconocimiento de voz integrados en el robot EVA.	33
15. Arquitectura publicador/suscriptor del protocolo MQTT.	35
16. Interfaz web del sistema Rhasspy.	36
17. Ejemplo de flujo de datos en nuestra arquitectura basada en eventos	39
18. EVAnotebook con código para detectar la dirección de atención del usuario.	40
19. Representación de la forma de una mano a partir de la extracción de los puntos de referencia utilizando la técnica de análisis de procrustes. a) Imagen original de la forma de la mano. b) Análisis de procrustes aplicada a la imagen original para alterar sus características originales (rotación, traslación, escalado y normalización).	42
20. Robot EVA inicia proactivamente una interacción basada en la proximidad del usuario y la dirección de la mirada.	48

21.	Estimación de la orientación del rostro a partir del modelo de Mediapipe face mesh. . . .	50
22.	EVA realiza el seguimiento facial de una persona a través del movimiento de su cabeza para mantener un ángulo visible de la cámara. a) Girando su cuello hacia la derecha. b) Manteniendo su cuello en la posición inicial. c) Girando su cuello hacia la izquierda. . . .	50
23.	Diagrama de secuencia de un usuario realizando ejercicio y siendo monitoreado por el robot EVA.	51
24.	Resultados de las gráficas obtenidas de la frecuencia cardíaca transmitida de una persona realizando ejercicio, a partir de dos sensores fisiológicos. a) Apple Watch. b) Polar H10 .	53
25.	Diagrama de secuencia de un usuario pidiendo a EVA una receta con los ingredientes que tiene a mano.	54
26.	Usuario interactuando con el robot EVA para preguntarle sobre recetas de comida con los ingredientes que se encuentran en una mesa. a) Usuario señalando con la mano el plato con los ingredientes. b) EVA agachando la cabeza para ver los objetos sobre la mesa. c) EVA sugiriendo las recetas que se pueden cocinar con los ingredientes que reconoció. . .	56
27.	Interacción con el robot EVA de un usuario adivinando los ingredientes de un té utilizando el olfato. a) EVA reconoce que el usuario está oliendo el té. b) EVA indicándole al usuario una pista de algún ingrediente que contiene el té.	57
28.	Interacción con el robot EVA de un usuario conversando sobre sus acontecimientos relevantes en Cuernavaca. a) EVA realizando preguntas al usuario sobre su pasado en Cuernavaca. b) Usuario respondiendo ante las preguntas del robot.	58
29.	Etapas de lenguajes gesto-espaciales.	60
30.	Interpretación de un lenguaje gesto-espacial.	61
31.	Modelo Face Mesh para la estimación de la orientación del rostro a partir de la extracción de puntos faciales. Tomada de (Grishchenko et al., 2020).	74
32.	Modelo Pose Landmark para la detección de posturas corporales a partir de la extracción de puntos corporales. Tomada de (Singh et al., 2021).	75

Lista de tablas

Tabla	Página
1. Escenarios de interacción humano-robot propuestos para la implementación de la arquitectura de software	17

Capítulo 1. Introducción

En este primer capítulo se presentan los antecedentes sobre el área de interacción humano-robot, además se presenta el planteamiento del problema. Más adelante, se exponen las preguntas de investigación y los objetivos que surgen de este trabajo, así como la propuesta de solución. Posteriormente, se presenta la metodología a seguir y las contribuciones esperadas. Por último se presenta una breve descripción de la organización de la tesis.

1.1. Antecedentes

Durante los últimos años se han generado avances importantes en el campo de la robótica, donde el objetivo no es solo proporcionarle al ser humano una asistencia física, sino estimular a las personas a través de la interacción entre humanos y robots. La inclusión de robots en entornos humanos implica un conocimiento de los diversos aspectos que intervienen en la interacción que tiene lugar, o bien, en las posibles interacciones entre humanos y robots en diversos escenarios. Con la integración de nuevas tecnologías en nuestra vida diaria, como asistentes virtuales, chatbots y robots de servicio, la interacción entre humanos y máquinas se ha vuelto más común, por lo que se hace cada vez más necesario diseñar robots sociales que sean capaces de adaptarse a las situaciones y expectativas de los seres humanos en su entorno (Wilcox et al., 2013).

En los primeros diseños de robots sociales (Breazeal et al., 2016), las señales no verbales que están presentes en la interacción humana se han utilizado activamente para enriquecer las interacciones con el robot. Estas mismas suele combinarse con el robot para proporcionar información complementaria sobre el estado interno o las intenciones del robot. Por ejemplo Kismet, uno de los primeros robots sociales, utilizaba posturas, como inclinarse hacia atrás o hacia delante, para expresar y hacer participar a las personas en la interacción (Wistort & Breazeal, 2009). En contraste Keepon, un robot social minimalista, utilizaba la mirada y el movimiento reactivo para expresar atención y el contacto (Kozima et al., 2009).

Estos robots deberían ser capaces de percibir y entender el mundo desde su propia perspectiva, comunicarse entre sí y aprender de manera explícita (Terrence, 2003). Además, son parte de un campo de estudio llamado computación afectiva (Picard, 2000), la cual se enfoca en simular procesos emocionales o aprovechar las emociones humanas en la interacción entre personas y computadoras (Xia et al., 2005).

Los beneficios de dotar a robots sociales con la capacidad de detectar diferentes tipos de comportamientos no verbales son enormes. Por ejemplo, la detección automática de lo que una persona está haciendo en un momento dado puede mejorar su comprensión de las intenciones humanas, los estados emocionales y las preferencias que estos presentan (Juan et al., 2013).

En el caso de la interacción humano-robot (IHR), campo que estudia a los humanos, robots y las formas en que se influyen entre sí (Fong et al., 2003b), en el cual su relevancia tiene lugar en un entorno social, se espera que el robot sea capaz de adherirse a las expectativas inherentes a dicho entorno. Un aspecto clave de los robots sociales es ser perceptivos del comportamiento del compañero de interacción y ofrecer una respuesta afectiva adecuada. Esto lleva a la necesidad de desarrollar arquitecturas de control para la generación de comportamientos.

1.2. Planteamiento del problema

En la última década, la creciente presencia de robots en diversos ámbitos de la sociedad ha generado un cambio significativo en la dinámica de las interacciones humanas. Los robots están siendo cada vez más integrados en entornos cotidianos, como hogares, lugares de trabajo, hospitales y espacios públicos. Esta integración plantea una serie de desafíos relacionados con la IHR, que deben abordarse para garantizar una convivencia armoniosa y efectiva entre humanos y máquinas (Wang et al., 2005).

El reconocimiento preciso del comportamiento no verbal de los seres humanos es un desafío técnico significativo debido a la naturaleza compleja e inherentemente ambigua de las señales no verbales. Los robots actuales a menudo carecen de la capacidad para captar y analizar adecuadamente estas, lo que puede resultar en interacciones robóticas ineficientes y frustrantes para los usuarios. Además, una detección incorrecta o inexacta del comportamiento no verbal puede llevar a respuestas inapropiadas por parte del robot, lo que afecta negativamente la aceptación social y la confianza de los usuarios hacia estas tecnologías.

Trabajar con información multimodal en la detección de comportamientos no verbales es un desafío complejo que requiere la integración de datos de diferentes modalidades, la adaptación a la variabilidad individual y la consideración de preocupaciones éticas y de privacidad. Además, la fusión de características, la interpretación contextual y la evaluación precisa son elementos clave para desarrollar sistemas efectivos en este campo. La falta de avances significativos en la detección de comportamiento no verbal

puede limitar el potencial de los robots para colaborar con los humanos de manera más efectiva y natural, lo que restringe su adopción en aplicaciones críticas y de amplio alcance. Por lo tanto, es esencial abordar este problema para mejorar la calidad de la interacción humano-robot y maximizar la aceptación social de estas tecnologías.

1.3. Preguntas de investigación

Con base en lo anterior, surgen las siguientes preguntas de investigación:

- ¿Qué características debe tener una arquitectura de software que facilite el desarrollo de interacciones con robots que involucren comunicación no verbal?
- ¿Cómo puede un robot social incorporar el reconocimiento de comportamientos no verbales para mejorar su interacción con los humanos?
- ¿Qué beneficios aporta a desarrolladores de IHR el implementar una arquitectura de software en la detección de comportamientos no verbales?

1.4. Objetivos

A continuación se presentan los objetivos, general y específicos que surgen de las preguntas de investigación:

1.4.1. Objetivo general

Diseñar e implementar una arquitectura de software que integre datos multimodales de diversos dispositivos y modelos de aprendizaje para la detección de comportamientos no verbales, y su evaluación en escenarios de interacción humano-robot.

1.4.2. Objetivos específicos

- Definir escenarios de interacción humano-robot que puedan hacer uso de la arquitectura a proponer.
- Proponer, integrar y evaluar modelos que incorporen datos unimodales/multimodales para la detección de gestos faciales, posturas corporales y señales fisiológicas.
- Implementar un conjunto de los escenarios definidos anteriormente para evaluar la arquitectura propuesta, en términos de adaptabilidad, flexibilidad y rendimiento.

1.5. Propuesta de solución

Para atender el problema planteado, se propone el desarrollo de una arquitectura que permita a investigadores y desarrolladores mejorar continuamente las capacidades de los robots sociales. A medida que se desarrollan nuevos sensores y modelos, podrán integrarse en dicha arquitectura, ampliando la capacidad del robot para reconocer y responder a una gama más amplia de comportamientos y tareas no verbales. Esta adaptabilidad fomenta la innovación y allana el camino para robots sociales más sofisticados, capaces de responder a diversas necesidades y contextos humanos.

1.6. Metodología

A continuación, en la Figura 1 se describe la metodología para el desarrollo de esta tesis la cual consta de cinco fases:

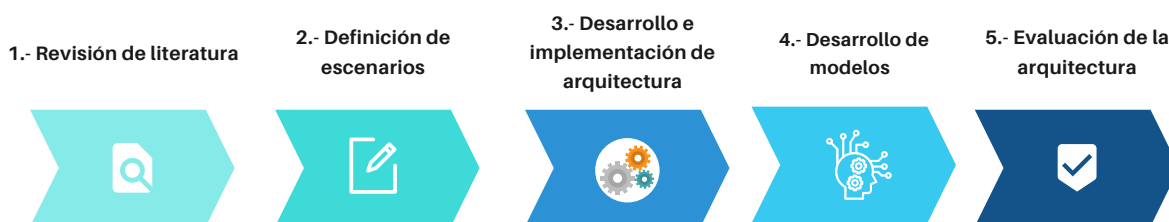


Figura 1. Metodología del trabajo de investigación

- **Fase 1: Revisión de literatura.** Se realizó una revisión de literatura con respecto a temas relacionados con la arquitectura a implementar, el uso de comunicación no verbal en IHR y los modelos que existen en el estado del arte para reconocer diferentes comportamientos no verbales.
- **Fase 2: Definición de escenarios.** Se establecieron diferentes tipos de escenarios en IHR que permitieran establecer los requisitos necesarios para el funcionamiento de la arquitectura propuesta con el objetivo de detectar comportamientos no verbales en distintos tipos de interacciones.
- **Fase 3: Desarrollo e implementación de arquitectura.** Se desarrolló un *framework* destinado a facilitar la integración de diversos modelos y dispositivos para el reconocimiento de comunicación no verbal para adaptar el comportamiento y respuesta del robot social EVA.
- **Fase 4: Desarrollo de modelos.** En esta etapa, se desarrollaron e implementaron modelos de lenguaje grande (LLMs), reconocimiento de voz y visión por computadora que funcionaban a través de dispositivos que recopilaban datos unimodales y multimodales para que con el uso de modelos externos, se pudieran detectar gestos faciales, posturas corporales e incluso señales fisiológicas.
- **Fase 5: Evaluación de la arquitectura.** En esta última etapa, se evaluó la arquitectura propuesta a partir de implementar los escenarios definidos, esto con el objetivo de analizar el rendimiento, la flexibilidad y adaptabilidad al momento de detectar comportamientos no verbales.

1.7. Contribuciones esperadas

Como resultados de este trabajo se esperan las siguientes contribuciones:

- Desarrollo e implementación de una arquitectura de software que facilite la integración de múltiples modelos y dispositivos para abordar diversos desafíos vinculados a la interacción no verbal.
- Facilitar el desarrollo de escenarios de interacción humano-robot que apoyen el reconocimiento de señales no verbales del usuario para adaptar el comportamiento de un robot social.
- Mejorar el rendimiento del robot social EVA, así como reducir significativamente los recursos requeridos para llevar a cabo una tarea específica que demande un elevado nivel de capacidad computacional.

1.8. Estructura de la tesis

El resto de la tesis se desglosa de la siguiente manera:

En el **Capítulo 2** (*Fundamentos teóricos*) se introducen los conceptos de robot social, comunicación no verbal, los diferentes tipos de interacciones no verbales y se describen tres escenarios de comunicación no verbal desarrollados para identificar los requisitos de diseño de la arquitectura propuesta.

En el **Capítulo 3** (*Trabajo relacionado*) se realiza una revisión sobre los trabajos mas importantes en el área de comunicación no verbal en interacción humano-robot, las diferentes arquitecturas que se han desarrollado, así como las características con las que cuenta.

En el **Capítulo 4** (*Arquitectura de interacción no verbal en IHR*) se presenta el proceso que se llevó a cabo para el desarrollo e implementación de la arquitectura propuesta. Se describe brevemente el robot social EVA usado como base para el desarrollo, los conceptos de protocolo de comunicación broker MQTT, Rhasspy, EVAnotebook y los modelos mencionados anteriormente que se utilizaron para la detección y comprensión de comportamientos no verbales.

En el **Capítulo 5** (*Implementación de escenarios*) se describe el diseño y desarrollo de escenarios en IHR que se llevaron a cabo en este trabajo, utilizando la arquitectura desarrollada previamente.

Por último, en el **Capítulo 6** (*Conclusiones y trabajo a futuro*) se presentan las conclusiones derivadas del trabajo de tesis, las contribuciones, limitaciones y el trabajo a futuro.

Capítulo 2. Fundamentos teóricos

En este segundo capítulo se introducen los conceptos básicos utilizados en el presente trabajo de tesis. En la primera sección se define el concepto de arquitectura de software, en la segunda y tercera sección se describen los conceptos de robot social, la comunicación no verbal y sus tipos. En la cuarta sección se describen los escenarios propuestos para la implementación de la arquitectura propuesta. Por último, en la quinta sección y quinta sección se realiza un análisis de los escenarios para identificar que requerimientos necesita la arquitectura y se termina con una conclusión del capítulo.

2.1. Arquitectura de software

En la actualidad, el software está presente en gran cantidad de objetos que nos rodean: desde los teléfonos y otros dispositivos que llevamos con nosotros de forma casi permanente, hasta los sistemas que controlan las operaciones de organizaciones de toda índole o los que operan las sondas robóticas que exploran otros planetas. Uno de los factores clave del éxito de los sistemas es su buen diseño; de manera particular, el diseño de lo que se conoce como arquitectura de software. La definición general que propone el Instituto de Ingeniería de Software (SEI, por sus siglas en inglés) es la siguiente: La arquitectura de software de un sistema es el conjunto de estructuras necesarias para razonar sobre el sistema. Comprende elementos de software, relaciones entre ellos, y propiedades de ambos (Bass et al., 2012).

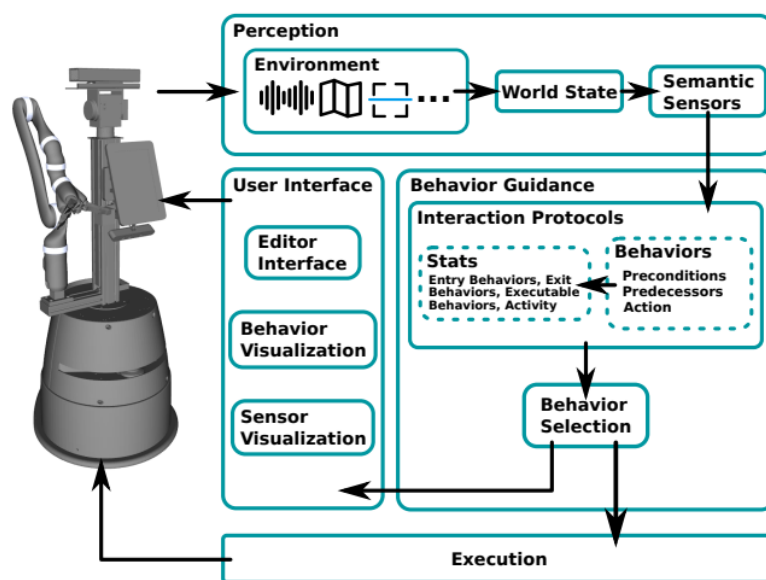


Figura 2. Panorama general de la arquitectura de control del robot

En el trabajo de (Hindemith et al., 2021) se desarrolló una arquitectura de software diseñada para hacer que el proceso de toma de decisiones y el comportamiento del robot sean mas comprensibles para los usuarios que no sean expertos en el área. El objetivo de esta arquitectura fue el mejorar la comprensión por parte de los usuarios, abordando tanto la confiabilidad de los robots como la comprensión de las limitaciones de estos. En la Figura 2 se muestra la arquitectura que se emplea en el trabajo mencionado anteriormente.

2.2. Robots sociales

A medida que la presencia de robots incrementa en diversos ámbitos de nuestra vida, surge también la necesidad de que estos tengan cierta inteligencia social. La razón radica en que, mediante esta inteligencia, pueden interactuar de manera eficaz con humanos. El uso debido de la comunicación no verbal por parte de robots puede ser crucial, ya que permite la interacción intuitiva entre humanos y robots. El comportamiento y la comunicación no verbal son más instintivos e involuntarios que la comunicación verbal (Wharton, 2009). Consecuentemente, son una representación más verdadera de los pensamientos y emociones humanas.

Un robot social es un agente físico autónomo que se puede comunicar con los humanos (Fong et al., 2003a). Si bien la comunicación verbal sigue siendo el principal modo de interacción entre los robots sociales y los humanos, se ha explorado la comunicación no verbal para mejorar la interacción humano-robot (Saunderson & Nejat, 2019). Esto implica la promulgación de actos de comunicación por parte del robot, como gestos con las manos o la cara y/o el reconocimiento por parte del robot de interacciones humanas no verbales, como la mirada o el movimiento del cuerpo.

Los robots sociales pueden adoptar formas muy variadas y también pueden tener diferentes posibilidades de representar y reconocer señales de comunicación no verbal, mostrando diferentes tipos de interacciones no verbales entre humanos y robots. Algunos ejemplos conocidos son: NAO, que puede seguir caras, establecer contacto visual, imitar gestos humanos y responder al tacto. También puede mostrar emociones a través de sus luces LED y movimientos corporales (Shamsuddin et al., 2011).

El robot Cozmo (Ver Figura. 3a) utiliza sus expresivos ojos digitales, su lenguaje corporal y sus sonidos para comunicarse con los usuarios y mostrar entusiasmo, frustración y curiosidad (Kusumota et al., 2018). Otro ejemplo es Jibo (Ver Figura. 3b), que utiliza el reconocimiento facial y animaciones expresivas para

establecer conexiones emocionales con los usuarios mediante señales no verbales como asentir con la cabeza, parpadear y expresar curiosidad mediante movimientos de la cabeza (Rane et al., 2014).



Figura 3. Ejemplos de robots sociales comerciales con comunicación no verbal. a) Robot Cozmo realizando un gesto de asombro. b) Robot Jibo interactuando mediante reconocimiento facial.

Anteriormente, el estudio de los robots sociales se centraba principalmente en abordar cuestiones algorítmicas enfatizadas en el reconocimiento de gestos corporales o faciales, con el propósito de ampliar las habilidades de los robots para mejorar su comunicación con los seres humanos (Hegel et al., 2009). En los últimos años, la tecnología ha alcanzado una robustez suficiente para permitir cierta autonomía durante las interacciones de los usuarios con los robots sociales.

Se han experimentado avances notables en áreas como la síntesis de voz a partir de texto (TTS, por sus siglas en inglés) y la conversión de voz a texto (STT), así como en los sistemas encargados de comprender el lenguaje natural. Además, el incremento de los adultos mayores que padecen enfermedades que les impidan vivir solos, ha conducido al uso de robots sociales como iRobot y Paro para realizar investigaciones en el campo de la interacción humano-robot (Leite et al., 2013).

2.3. Comunicación no verbal

La comunicación no verbal se define generalmente como cualquier transferencia de mensajes que no implique el uso de palabras (Hall et al., 2019). Mientras que la comunicación verbal tiende a dominar las interacciones humanas, la comunicación no verbal, como los gestos y la mirada, aumenta y amplía la comunicación hablada. La comunicación no verbal se produce bidireccionalmente en una interacción,

por lo que los robots sociales deben ser capaces de reconocer y generar comportamientos no verbales.

A su vez, la comunicación no verbal, desempeña un papel fundamental en la interacción social, ya que proporciona información valiosa sobre el estado emocional, las intenciones y las actitudes de los individuos. En el contexto de la interacción humano-robot, esta forma de comunicación puede mitigar la barrera entre lo humano y lo mecánico, mejorando la empatía y la comprensión mutua.

En la interacción humano-robot pueden darse varios tipos de interacciones no verbales (Ver Figura. 4). Estas interacciones abarcan una amplia gama de modalidades y señales de comunicación que van más allá del lenguaje hablado (Urakami & Seaborn, 2023).

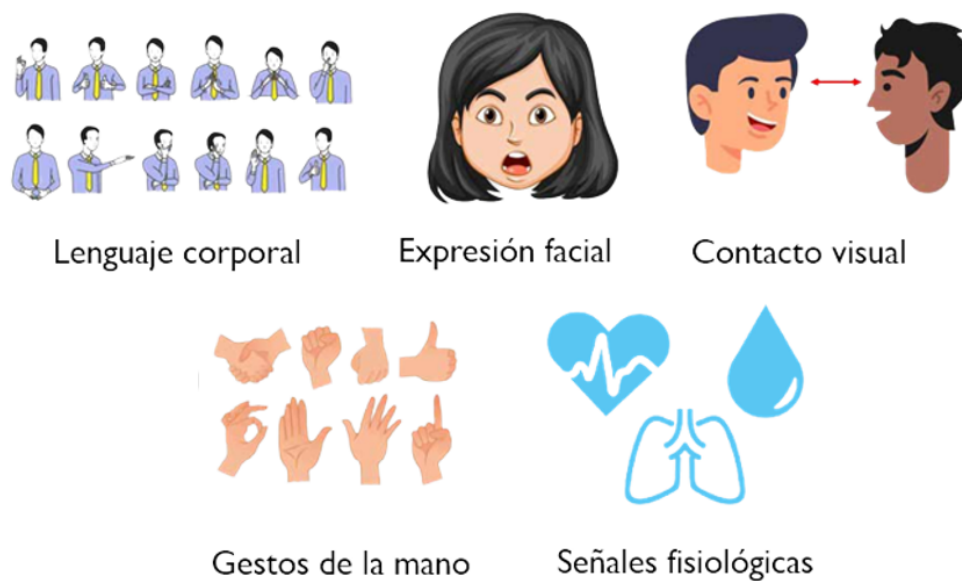


Figura 4. Tipos de comunicación no verbal. Tomada de (Kumar, 2023)

Aunque exhibimos y experimentamos las señales no verbales en varias modalidades a la vez, como el sonido, el movimiento y la mirada, podría valer la pena considerar cada canal de comunicación por separado (modalidad unimodal) al tratar de implementar las señales no verbales, para después abordarlas en conjunto (modalidad multimodal).

2.3.1. Expresiones faciales

Una expresión facial puede describirse como un movimiento o posición de los músculos en la cara de un individuo, que comunican una amplia gama de emociones, sentimientos y estados internos (Faria et al.,

2017). Otras personas pueden saber mucho sobre el estado emocional de alguien simplemente observando su expresión facial, por lo cual es un medio para comunicarse sin necesidad de usar comunicación verbal.

Las expresiones faciales son un componente fundamental de la comunicación no verbal humana, que desempeña un papel esencial en la transmisión de emociones, intenciones y estados mentales. En el contexto de la interacción humano-robot, las expresiones faciales cobran una nueva dimensión al ser empleadas por los robots para comunicarse con los humanos.

Varios estudios en el área de IHR han discutido si se debería otorgar a los robots la capacidad de reconocer expresiones faciales, explorando el significado y la relevancia de estas últimas en la interacción entre humanos y robots. Dichas investigaciones destacan el papel de las expresiones faciales en la construcción de conexiones emocionales y en la mejora de la comunicación.

Además de interpretar emociones, las expresiones faciales también ayudan a que la comunicación entre humanos y robots sea más fluida y natural. Cuando el robot social detecta que el usuario está confundido, puede adaptar su manera de hablar y explicar las cosas de una manera más clara. Si el usuario se encuentra emocionado, el robot podría mostrar entusiasmo en su voz y movimientos, creando así una conexión emocional más fuerte.

2.3.2. Gestos

Después del habla, el gesto es quizás una de las formas más evidentes de proporcionar información durante una interacción. Los gestos pueden sustituir, acompañar o ir junto con el habla, y a menudo se clasifican según su papel en la comunicación. Los gestos deícticos se refieren a señalar cosas específicas en el entorno y pueden ser importantes para establecer una atención conjunta. Los gestos icónicos suelen acompañar al habla, apoyando e ilustrando lo que se dice.

Por ejemplo, abrir los brazos mientras se dice que se está sosteniendo una gran pelota sería un gesto icónico, como lo sería mover suavemente la mano hacia arriba mientras se explica como gestos simbólicos (Yang et al., 2007). El reconocimiento de gestos a partir de robots sociales representa una herramienta poderosa para mejorar la calidad y la naturaleza de nuestras interacciones con estos mismos, abriendo nuevas posibilidades en la forma en que los robots pueden complementar y enriquecer nuestras vidas cotidianas.



Figura 5. Tipos de gestos en la comunicación no verbal.

En la Figura 5 se muestran los diferentes tipos de gestos que existen en la comunicación no verbal con distintos ejemplos ilustrados.

2.3.3. Postura y lenguaje corporal

Las personas también se comunican a través de la postura de todo el cuerpo y la forma en que se mueven. Junto con la expresión facial, las posturas pueden servir para interpretar el estado emocional de una persona. Los movimientos lentos, los hombros caídos y los gestos letárgicos sugieren un estado de ánimo abatido, mientras que los movimientos rápidos y el porte erguido son signos de una actitud positiva.

Estos tipos de señales posturales son especialmente importantes cuando la cara de una persona no es visible, pero también pueden proporcionar pistas adicionales sobre el estado de ánimo de una persona incluso cuando podemos ver su expresión facial. Los investigadores han descubierto que las personas pueden interpretar estos tipos de señales no verbales no sólo cuando ven el cuerpo completo de la persona, sino también en pantallas de puntos luminosos minimalistas que representan los movimientos de una persona (Alaerts et al., 2011).

Las posturas corporales pueden proporcionar una capa adicional de expresividad a los robots. Por ejemplo, cuando un robot carece de rasgos faciales expresivos, el cuerpo puede utilizarse como forma principal de comunicar emociones. Se ha demostrado que las posturas corporales eficaces pueden mejorar la comprensión de las personas sobre el estado emocional de un robot (Beck et al., 2010).

2.3.4. Proxémica

La proxémica se define como el distanciamiento físico y psicológico de los demás (Hans & Hans, 2015). Las personas pueden percibir a los robots que no muestran un comportamiento de distanciamiento adecuado como una amenaza y una perturbación de su entorno social y su práctica laboral (Mutlu & Forlizzi, 2008).

Se ha estudiado en (Walters et al., 2005; Rossi et al., 2017) que las personas prefieren mantener una distancia considerada con un robot dependiendo de su apariencia física y en qué humor se encuentra. Sin embargo, se podría dar una solución al problema anterior diseñando comportamientos proxémicos en los robots, lo cual podría fomentar relaciones más estrechas entre humanos y robots, y permitir una aceptación generalizada de los robots, contribuyendo a su integración sin fisuras en la sociedad.

Las distancias interpersonales físicas deben ajustarse a las normas sociales (distancias relativas entre las personas) que se expresan en cuatro zonas distintas, es decir, espacio íntimo, espacio personal, espacio social y espacio público (Silva et al., 2015), como se muestra en la Figura 6. Cabe mencionar que existen robots sociales los cuales tienen diferente tipo de zona proxémica: El robot PARO, el cual es un robot terapéutico diseñado para brindar apoyo emocional y social a personas (Lane et al., 2016), especialmente en entornos de atención médica y terapéutica, fue diseñado para un espacio íntimo. El robot Jibo fue diseñado para un espacio personal (Rane et al., 2014), debido a que utiliza el reconocimiento facial para establecer conexiones emocionales con los usuarios mediante señales no verbales.

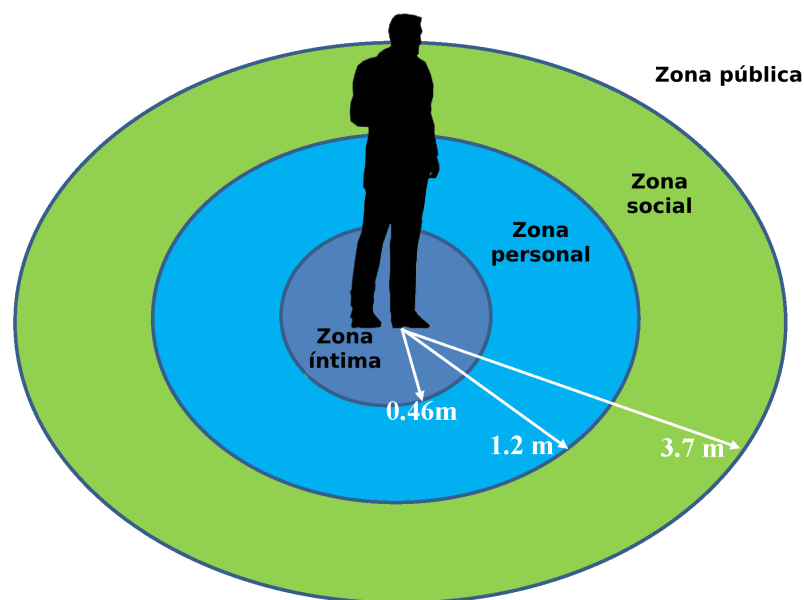


Figura 6. Distancias proxémicas (Hans & Hans, 2015).

2.3.5. Seguimiento ocular

La mirada es una señal sutil e importante para gestionar la interacción social. La mirada señala el interés, la comprensión, la atención, la capacidad y disposición de las personas para seguir la conversación. El uso de la metodología de seguimiento ocular para evaluar los patrones de la mirada puede proporcionar información sobre el procesamiento de la información y la cognición humana.

Un componente particularmente bien establecido del comportamiento de la mirada en la interacción humana es la atención conjunta. La atención conjunta consiste en que los interlocutores atienden a la misma zona u objeto al mismo tiempo. La atención conjunta se ha incorporado al IHR de varias maneras, por ejemplo, en (Imai et al., 2003) la utilizaron para facilitar la comunicación con las personas, de forma que éstas supieran de qué está hablando el robot, tanto en combinación con el habla como sin ella.

Cuando se utilizan en IHR, las señales de la mirada del robot suelen generar efectos similares a los de las interacciones humanas. Esto puede deberse a que los investigadores han utilizado el comportamiento de la mirada humana para derivar modelos de comportamiento visual en robots. Han demostrado que las señales resultantes de la mirada pueden ser empleadas para inducir a las personas a adoptar diferentes roles conversacionales, ya sea como destinatarios, espectadores o no participantes (Mutlu et al., 2009). En un estudio de IHR, (Andrist et al., 2014) utilizaron movimientos de seguimiento facial para realizar miradas mutuas y evasiones intencionadas de la mirada. Este estudio demostró que dichas señales pueden conferir al robot una apariencia más intencionada y reflexiva.

2.3.6. Señales fisiológicas

Algunas de las señales fisiológicas propuestas para utilizar en interacción humano-robot incluyen la conductancia de la piel, la frecuencia cardíaca, la dilatación de las pupilas y la actividad cerebral y muscular. Aunque las señales fisiológicas tienen el potencial de proporcionar medidas objetivas del estado interno de los humanos, son difíciles de interpretar. Esto se debe en parte a su variabilidad de una persona a otra y a la activación de múltiples estados emocionales para una señal fisiológica. Por lo tanto, puede ser difícil comprender el estado emocional interno del usuario. La literatura de psicofisiología recomienda utilizar un sistema fisiológico multimodal capaz de detectar simultáneamente una variedad de señales fisiológicas correspondientes a un estado específico (Val-Calvo et al., 2020).

2.4. Diseño de escenarios

En esta sección, se presentan los escenarios de diseños propuestos que representan situaciones de IHR que se benefician de comunicación no verbal. Estos escenarios se desarrollaron con base en la literatura de aplicaciones de robots sociales y la experiencia de desarrollo de IHR del Departamento de Ciencias de la Computación del CICESE:

2.4.1. Escenario 1: Mirada y proxémica para iniciar una interacción

EVA es un robot social que puede desplegar sus habilidades para brindar apoyo a Roberto, un paciente que afronta los desafíos de la demencia mientras vive solo en su hogar. La presencia de EVA ha traído beneficios a Roberto, ya que la tecnología innovadora y empática ha demostrado ser una compañera invaluable en su día a día. Los síntomas de demencia han dejado a Roberto a menudo desorientado y sintiéndose solo en su propio espacio.

Sin embargo, gracias a EVA, nunca se encuentra realmente solo. Cuando la necesidad de conexión humana se presenta, Roberto se acerca a EVA en busca de una interacción significativa. La cámara integrada en el robot permite que EVA determine la distancia exacta entre ambos, asegurando que esté lista para responder de manera adecuada. Si la distancia entre ellos excede los tres metros (zona social), EVA responde de inmediato al movimiento de Roberto, ajustando su posición moviendo su cuello para seguirlo, creando así una sensación de compañía en su entorno y así evitar que la interacción termine por completo.

Si Roberto se acerca aún más (zona personal), el robot percibe su cercanía con él e intenta atraer su atención con una interacción. La habilidad de EVA para detectar las intenciones de Roberto y leer su lenguaje corporal le permite cambiar su comportamiento de forma intuitiva. Cuando Roberto fija su mirada a EVA durante más de dos segundos, el robot comprende que está buscando interactuar y establecer una conexión. Esto marca el inicio de una interacción especial. Con su voz suave y amable, EVA rompe el hielo: **"Hola Roberto, ¿cómo te ha ido en tu día?"**. Esta simple pregunta es la que empieza con la interacción y que puede abrir las puertas a conversaciones que pueden variar desde anécdotas personales hasta otro tipo de conversaciones.

2.4.2. Escenario 2: Rutina de ejercicio

Antonio, una persona mayor se encuentra realizando actividades físicas diariamente, como una rutina de ejercicio. EVA tiene la capacidad de medir la frecuencia cardiaca de la persona a través de un sensor fisiológico y monitorear su condición física durante la rutina. Cuando Antonio comienza la rutina de ejercicios, EVA activa el sensor fisiológico para medir continuamente la frecuencia cardíaca. Si Antonio supera un valor predefinido de la frecuencia cardíaca que se pudiera considerar peligrosa o inadecuada para su salud, EVA tomará medidas para garantizar la seguridad de él.

En primera instancia, EVA emitirá una advertencia verbal a la persona que hace ejercicio. Utilizando su capacidad de generación de voz, EVA informará a la persona sobre el problema y le aconsejará que detenga o disminuya la intensidad del ejercicio para evitar riesgos para su salud. Simultáneamente, EVA se conectará a servicios externos para enviar un correo electrónico a las personas designadas como contactos de emergencia, notificándoles el problema que está ocurriendo, y además de esto, enviará adicionalmente mensajes de texto.

En el correo electrónico y el SMS, EVA incluirá información relevante, como el nombre de Antonio, su ubicación y su frecuencia cardiaca actual, para que los contactos puedan tomar las medidas necesarias para ayudar a la persona en caso de emergencia. Pueden ponerse en contacto directamente con la persona o llamar a los servicios médicos de urgencia si es necesario.

2.4.3. Escenario 3: Recomendaciones de recetas de comida

Luis, un entusiasta joven apasionado por la rica cocina mexicana, se encuentra en la encrucijada de profundizar en el conocimiento de los platos emblemáticos de su país. A pesar de su limitada experiencia en la preparación de estas delicias culinarias, decide recurrir a la perspicaz asistencia de EVA, una robot social dotada de habilidades culinarias.

Con curiosidad palpable, Luis dirige su atención hacia la mesa cuidadosamente dispuesta con una selección de ingredientes de comida. Lleno de expectativas, inicia la conversación con EVA con una pregunta precisa: **"Hola, EVA. ¿Que puedo preparar con esto?"**.

EVA, con la ayuda de la cámara que tiene integrada, realiza el reconocimiento y analiza los ingredientes

que tiene a su disposición para posteriormente ofrecerle a Luis una serie de sugerencias de recetas de comida que se alinean perfectamente con los ingredientes que se encuentran sobre la mesa. Estas propuestas reflejan la autenticidad y la diversidad de la cocina mexicana, abriendo ante Luis una amplia variedad de posibilidades para cocinar.

A medida que Luis navega entre las alternativas presentadas por EVA, emerge un diálogo constructivo en el que ambos discuten las virtudes de cada receta y su viabilidad en función de los ingredientes disponibles y el nivel de habilidad culinaria de Luis. Finalmente, Luis toma una decisión informada sobre la receta que emprenderá. EVA, con su característica empatía y conocimiento detallado, asume el papel de guía culinaria personal, desglosando la receta elegida en pasos minuciosos y claros.

2.5. Análisis de los escenarios

El diseño y desarrollo de una arquitectura de software apta para detectar y comprender comportamientos no verbales en escenarios de IHR demanda una cuidadosa planificación y consideración de varios elementos cruciales. Estos requisitos fundamentales buscan crear una experiencia fluida, natural y efectiva para los usuarios, promoviendo así una comunicación enriquecedora y empática entre humanos y robots. En la tabla 1 se muestran los escenarios de IHR que se proponen en este trabajo, su tipo de comunicación no verbal y el dispositivo de sensado que se requiere para hacer la captura de los datos.

Tabla 1. Escenarios de interacción humano-robot propuestos para la implementación de la arquitectura de software

Escenario	Tipo de comunicación no verbal	Dispositivo requerido para la captura de datos
Mirada y proxémica para iniciar una interacción	Seguimiento ocular Proxémica	Cámara de profundidad
Rutina de ejercicio	Expresión facial Postura corporal Señal fisiológica	Cámara RGB Sensor fisiológico
Recetas de comida	Seguimiento ocular Postura corporal Gestos	Cámara RGB Micrófono

En el Escenario 1 que se categoriza como unimodal, donde la mirada y la proxémica son los puntos de partida para la interacción, se hace imperativo contar con sensores visuales y de proximidad avanzados. Cámaras con capacidades de reconocimiento de rostros y seguimiento ocular permitirán capturar y analizar el comportamiento visual del usuario, mientras que las cámaras de profundidad determinarán

la cercanía entre ambos. La interpretación de la mirada dirigida hacia el robot podría ser un indicativo clave de interés o disposición para la conversación. Adicionalmente, el análisis de la proxémica podría ofrecer información sobre el nivel de comodidad del usuario y su disposición a interactuar de manera cercana.

En el Escenario 2, centrado en el apoyo a personas mayores en su rutina de ejercicios y monitoreo de ritmo cardíaco, el enfoque principal radica en la precisión de la recopilación de datos fisiológicos y vídeo. Dispositivos portátiles en el usuario y el uso de cámaras, como cámaras de seguridad o relojes inteligentes, podrían garantizar una supervisión constante y precisa de lo que un adulto mayor se encuentre realizando en el momento. La arquitectura debe incluir algoritmos de análisis capaces de identificar patrones anómalos en el ritmo cardíaco, posturas corporales e incluso en expresiones faciales, y a partir de esto, adaptar el apoyo del robot en consecuencia, pudiendo sugerir pausas o ajustes en la intensidad de ejercicio que requieren módulos para la notificación a profesionales de la salud y familiares.

Por último, en el Escenario 3 se combinan tanto la comunicación verbal y no verbal, debido a que se utilizan datos multimodales para llevar a cabo la interacción, donde el robot debe analizar los ingredientes presentes en un plato de comida para sugerir recetas. Aquí, la arquitectura de software debe integrar capacidades de visión por computadora avanzada para identificar los ingredientes en el plato de comida del usuario de manera precisa, y a su vez, tener la capacidad de analizar la voz del usuario que se capta mediante un micrófono que tiene instalado el robot para pasárselo a un modelo de lenguaje.

Para iniciar con la interacción, el robot tendrá que detectar el movimiento que realice el usuario para indicarle donde se encuentra la comida y analizar que le está diciendo mediante un procesamiento de lenguaje natural. Para lograr esto, el robot podría estar equipado con cámaras de alta resolución y sistemas de visión que sean capaces de capturar imágenes detalladas de la comida. A través del análisis de estas imágenes, el robot debe ser capaz de reconocer y clasificar los ingredientes presentes. Esto implica el uso de técnicas de procesamiento de imágenes y algoritmos de visión por computadora, incluyendo la detección de formas, colores y texturas características de diferentes tipos de alimentos.

2.6. Conclusiones del capítulo

Además de los aspectos técnicos, la seguridad y privacidad son componentes esenciales de esta arquitectura. Los datos biométricos y la información personal compartida en estas interacciones deben ser

tratados con la máxima confidencialidad y respeto por la privacidad del usuario. La arquitectura debe incorporar medidas de cifrado, autenticación y protección de datos para garantizar un entorno seguro y confiable.

Los escenarios propuestos tienen diferente nivel de implementación y de dificultad por simples razones: en el primer escenario se requiere de una cámara de vídeo para ir calculando la distancia y orientación en la que se encuentra el usuario en el momento que el robot detecte que alguien está frente a él. El segundo y tercer escenario son más complejos debido a la diversidad de factores que están actuando en paralelo para que los resultados sean los más precisos. En el segundo se necesita del uso de sensores fisiológicos y de la cámara de vídeo para tener el control sobre la captura de datos del usuario sin necesidad de interferir en la interacción. El último escenario requiere de diversos sistemas de reconocimiento y análisis de voz e imagen para detectar cuando un usuario señale un plato de comida.

Capítulo 3. Trabajo relacionado

En este capítulo se revisan los trabajos más importantes en el área de detección de comportamiento no verbal en IHR a partir del análisis de datos multimodales utilizando algoritmos de visión por computadora y otras técnicas de reconocimiento de voz e imagen. En la primera sección se presentan los requisitos y habilidades que un robot social debe tener para mejorar la interacción con los humanos. En la segunda sección se presentan algunas arquitecturas de software que se han implementado en robots sociales para el reconocimiento de comportamientos no verbales. En la tercera sección se presentan los trabajos relacionados con la comunicación no verbal. En la cuarta sección se introducen los niveles de datos para representar las señales no verbales. Por último, en la quinta sección se mencionan algunas técnicas para reconocer comportamientos no verbales que se han empleado en diversos trabajos en el Estado del Arte.

Actualmente existen muchos trabajos que estudian las distintas formas de comunicación no verbal en interacción humano-robot. Dicho esto, esta revisión de literatura se enfoca en analizar principalmente los trabajos relacionados con el reconocimiento de lenguaje no verbal en IHR y cómo los robots sociales perciben a los humanos durante la interacción con este mismo. En estos trabajos se tienen en cuenta algunas de las características de los experimentos realizados en estas investigaciones como el tipo de robot que utilizan, el tipo de estudio y lenguaje no verbal que presentan las personas.

3.1. Robots sociales

Como se definió previamente, los robots sociales son agentes físicos capaces de interactuar socialmente con los humanos (Fong et al., 2003a). Los robots sociales emplean estrategias de interacción sin intervención humana, incluido el uso del habla, expresiones faciales y gestos comunicativos para promover interacciones sociales eficaces (Fasola & Mataric, 2012). También pueden tener la capacidad de utilizar movimientos y gestos físicos para expresar y percibir emociones, y primordialmente deben de estar en condiciones de entablar una conversación mediante diálogos de alto nivel, así como la capacidad de exhibir una personalidad (Li et al., 2011).

En un trabajo de investigación de (Breazeal & Scassellati, 1999) dan una buena visión de los requisitos para desarrollar una arquitectura de software para robots autónomos socialmente inteligentes (SIAR). Se centran principalmente en la arquitectura de comportamiento del robot Kismet como ejemplo. Se argumenta que el intercambio social entre personas está “mutuamente regulado”, lo que significa que los participantes se adaptan al comportamiento del otro y responden en consecuencia. (Hegel et al.,

2009) propusieron que un robot social necesita dos atributos clave: un robot y una interfaz social. Una interfaz social engloba todas las características diseñadas por las que un usuario juzga que el robot tiene cualidades sociales. En la Figura 7 se puede observar el concepto de robot social que se menciona anteriormente, en la cual se engloban el robot y la interfaz social:

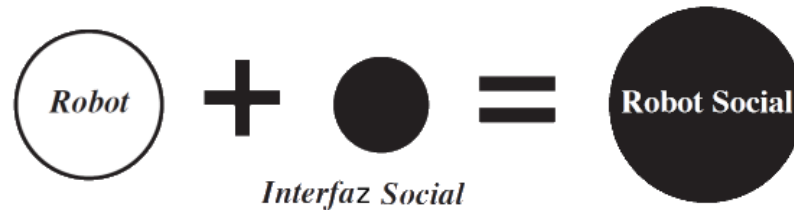


Figura 7. Definición de Robot Social por (Hegel et al., 2009)

Sin embargo, la definición anterior no cumple con los requerimientos para desarrollar una arquitectura de software adecuada para otorgarle a un robot social las herramientas de reconocer comportamientos no verbales en las personas. Es por eso que se ha tomado una alternativa para solucionar este problema. Este trabajo se enfoca principalmente en ofrecer una arquitectura capaz de integrar diversos modelos y dispositivos con el objetivo de reconocer la comunicación no verbal en los humanos.

3.2. Arquitecturas de robots sociales

En la revisión de la literatura, se observa que la mayoría de los robots sociales investigados carecen de una arquitectura de software que les permita incorporar diversos elementos para abordar el reconocimiento de la comunicación no verbal. Al examinar el trabajo relacionado de (Villa et al., 2022a)), se concluyó que para que el robot EVA pueda reconocer ciertos comportamientos no verbales en personas mayores con demencia, es crucial que el robot exhiba un comportamiento más natural, afectuoso y proactivo, lo cual genera una sensación de comodidad y aceptación por parte de la persona en su interacción con el robot.

Este trabajo cuenta con diferentes servicios, tales como: detección de presencia, *wakeface*, reconocimiento facial, entre otros más. Además, las consultas y respuestas de EVA se procesan mediante una plataforma de servicios cognitivos. En particular, el sistema de intención de diálogo *Watson Assistant* y el análisis de emociones de IBM se utilizan para respaldar las conversaciones EVA, así como texto a voz, voz a texto y traducción de Google Cloud (Villa et al., 2022b).

En el trabajo de (Rocha & Muchaluat-Saade, 2023), se propone un simulador en 3D para la plataforma robótica social FRED que puede ser utilizado para aplicaciones educativas y terapias de salud con niños autistas. El robot FRED tiene elementos de comunicación verbal y no verbal, a su vez, puede expresar emociones a través de los ojos y la boca. De hecho, este trabajo utiliza una parte de nuestra arquitectura para darle la capacidad de comunicarse con el simulador. Otro trabajo propuesto por él fue EvaML (un lenguaje basado en XML) y EvaSIM (un simulador para el robot EVA), capaz de ejecutar sus sesiones interactivas y ayudar en el desarrollo de programas para el robot (da Rocha & Muchaluat-Saade, 2023).

Es importante destacar que los trabajos previos mencionados no tienen el tipo de arquitectura que se desarrolló en este trabajo de tesis. Los robots y asistentes de última generación suelen incorporar mecanismos para mejorar el comportamiento natural similar al humano, pero siguen sin abordar el reconocimiento de interacciones no verbales en los humanos.

En los últimos años, ha habido un aumento en el interés y la investigación en asistentes robóticos que pueden realizar tareas centradas en la integración social, los vínculos afectivos y el entrenamiento cognitivo (Robinson et al., 2014). Estos aspectos son precisamente lo que ha motivado el desarrollo de una arquitectura abierta que nos brinde la posibilidad de mejorar EVA con la integración de diferentes dispositivos IoT y algoritmos de inteligencia artificial.

3.3. Comunicación no verbal

La comunicación es un proceso interactivo que involucra un emisor de señales, un canal de transmisión y un agente (o perceptor) que actúa sobre un estímulo. En el área de IHR, el agente o la fuente de los estímulos es un robot, pero también podría ser una persona. La forma que adopta este estímulo puede variar. En el caso de la comunicación no verbal, los estímulos no toman la forma del habla.

Existen dos formas primarias de comunicación en la robótica para establecer una interacción exacta entre un robot y un ser humano dentro de contextos físicos y psicológicos. La interacción verbal humana se expresa usando pensamientos, ideas y sentimientos a través del lenguaje hablado o escrito. Mientras tanto, en la kinesia, la comunicación no verbal se define a través de movimientos corporales, posicionamiento, expresiones faciales y gestos (Saunderson & Nejat, 2019). Sin embargo, la comunicación verbal no siempre nos aportará lo necesario para concluir con certeza lo que una persona piensa o siente en algún momento.

Es por esto que llevar a cabo un sensado fisiológico nos proporciona información valiosa para adaptar la interacción humano-robot de manera más efectiva y personalizada, mejorando así la calidad de la interacción y su utilidad en diversos contextos.

Como tal, definimos la comunicación no verbal en HRI como cualquier tipo de comunicación entre personas y robots que no involucra palabras, sino que incluye expresiones no verbales. Cabe aclarar que la comunicación es un proceso interactivo, donde los mensajes se intercambian continuamente entre el remitente y el receptor. En el trabajo de (Urakami & Seaborn, 2023), se toma un enfoque simple, donde se centran en escenarios donde el robot es el remitente de señales no verbales y una persona es el receptor, teniendo que decodificar y asignar significado a las señales transmitidas por el robot.

En los últimos años, se ha demostrado que la comunicación no verbal es un área vital de estudio ya que sus orígenes surgen en gran medida a través de comportamientos heredados o experiencias primarias comunes a todos los humanos, por ejemplo, el uso de las manos hacia la boca indicando comer (Saunderson & Nejat, 2019). Hay varios estudios que consideran la comunicación no verbal durante la IHR. (McColl et al., 2016) revisó el reconocimiento de la comunicación no verbal en humanos para la toma de decisiones de robots sociales en escenarios de IHR.

3.4. Tipos de datos

Dentro del ámbito de identificación de señales no verbales, la elección del sensor conlleva la asociación específica de la categoría de datos a considerar. Estos datos se dividen en dos clases: en primer lugar, se encuentran los datos 2D o bidimensionales, los cuales suelen ser derivados de imágenes, vídeos o dispositivos de captura (Hall et al., 2019).

Contrariamente, el segundo tipo abarca los datos 3D o tridimensionales (sensores de profundidad laser LIDAR, Kinect, etc), para cuya representación se requieren sensores más avanzados, permitiendo una captura desde diversas perspectivas y una construcción más eficaz de la escena (Astorga et al., 2023).

Una vez explorada la tipología de datos, se procede a abordar otro componente esencial: las estrategias y algoritmos empleados en el reconocimiento de comportamientos no verbales variados en el ámbito de IHR, basados en los datos obtenidos. En la siguiente sección se abordaran las técnicas que se han utilizado en la literatura para el reconocimiento de comportamientos no verbales.

3.5. Técnicas para el reconocimiento de comportamientos no verbales

Estudios recientes revelan que se utilizan técnicas estándar de reconocimiento de patrones para que los robots puedan percibir e identificar las señales no verbales de los humanos. El reconocimiento de posturas y gestos están bien estudiados. Los sistemas típicos utilizan cámaras RGB, cámaras de profundidad o sensores que lleva el usuario para registrar una serie temporal de datos. Aunque se podría escribir un software para reconocer un número limitado de gestos, lo habitual es utilizar el aprendizaje automático como sistema para reconocer gestos y otras señales no verbales.

Para ello, se recoge una base de datos de, por ejemplo, personas que muestran diferentes gestos. Por lo general, miles o incluso millones de datos son requeridos, y cada uno de ellos debe ser etiquetado. En el trabajo de (Mitra & Acharya, 2007), se entrenó un clasificador utilizando una red neuronal *Time Delay Neural Network* (TDNN) mediante el uso de datos etiquetados, con el propósito de reconocer gestos faciales y corporales.

Estas técnicas básicas de percepción permiten a los investigadores de IHR estimar si las personas están realmente involucradas en las interacciones con sus robots. A diferencia de la interacción humana típica, en la que se espera que el interlocutor humano esté atento y comprometido, en la IHR los usuarios a veces no atienden a lo que el robot dice y señala. Por eso, percibir el compromiso de los usuarios es un paso crucial para que los robots puedan crear una interacción satisfactoria.

En el trabajo realizado por (Rich et al., 2010) se desarrolló una técnica para integrar la detección de señales como el contacto visual y para identificar si un usuario está involucrado en la interacción. (Sanghvi et al., 2011) analizaron las posturas y los movimientos corporales para detectar el compromiso con un compañero de juego robótico.

Aunque el constante avance de la tecnología permite mejorar las capacidades de percepción de los robots, los investigadores también añaden equipamientos especiales al robot, como rastreadores oculares y sistemas de captura de movimiento, para proporcionar datos sobre señales no verbales relevantes para la interacción. Para la interacción táctil, se han realizado algunas investigaciones en el campo de la robótica en las que se insertaron sensores de polímero (Taichi et al., 2006).

En las siguientes subsecciones, exploraremos algunas de las principales técnicas utilizadas para el reconocimiento de comportamientos no verbales, brindando una visión general de cómo funcionan y cómo se aplican en diferentes contextos.

3.5.1. Seguimiento facial

Es una tecnología o técnica utilizada para detectar y rastrear automáticamente los movimientos y características faciales en tiempo real a través de cámaras u otros dispositivos de captura de imágenes (Kim et al., 2008). Consiste en la capacidad de un sistema informático para identificar y seguir la ubicación y las expresiones del rostro de una persona en un flujo de vídeo o imágenes.

Una variable perceptual importante en el análisis de las señales no verbales es el enfoque de atención, que indica que la persona está prestando atención a algo y a menudo se utiliza para transmitir intenciones interpersonales. A menudo sucede que las personas voltean hacia su enfoque de atención, expresando así físicamente su atención mediante la orientación de la cabeza. En la mayoría de los casos, la orientación de la cabeza está directamente relacionada con la estimación de la mirada visual, ya que la dirección de la mirada percibida está determinada por la orientación de la cabeza.

En el trabajo realizado por (Soomro et al., 2023), se emplea el método *Face mesh* implementado por MediaPipe para estimar en tiempo real 468 puntos clave del rostro humano. Este método es capaz de utilizar diseños de modelos ligeros y aceleración de CPU en todo el proceso para lograr un rendimiento en tiempo real. El método opera sobre el vídeo capturado utilizando la biblioteca OpenCV y calcula las posiciones de los rostros y un conjunto de puntos de referencia en 3D de los rostros que utiliza esas ubicaciones para predecir la geometría aproximada de la superficie a través de la regresión. Esto también tiene la ventaja de reducir en gran medida el requisito de aumentos de datos típicos, como transformaciones afines, que incluyen rotaciones, traslaciones y modificaciones de escala, ver Figura 8

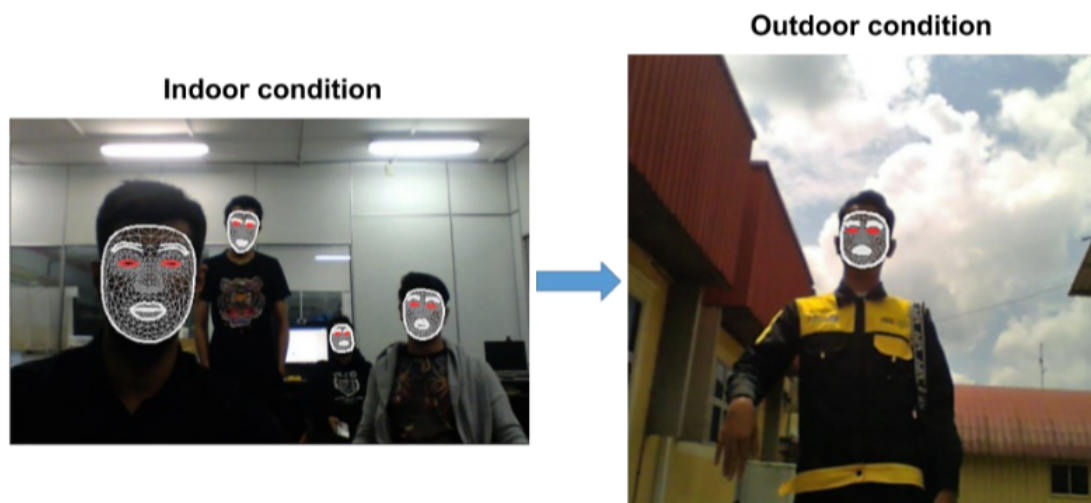


Figura 8. Resultados del modelo de seguimiento facial. Tomada de (Soomro et al., 2023).

3.5.2. Parpadeo del ojo

Aunque parpadear puede parecer un hábito inconsciente, una nueva investigación realizada por (Hömke et al., 2017) y sus colegas revela que cuando los seres humanos están conversando, reciben inconscientemente los parpadeos como indicadores no verbales (Królak & Strumillo, 2012). Además, la investigación ha revelado que los parpadeos ocurren con frecuencia durante las pausas naturales en el discurso. Hömke se preguntó si, al igual que asentir con la cabeza, un movimiento tan pequeño e inconsciente como parpadear podría funcionar como retroalimentación verbal.

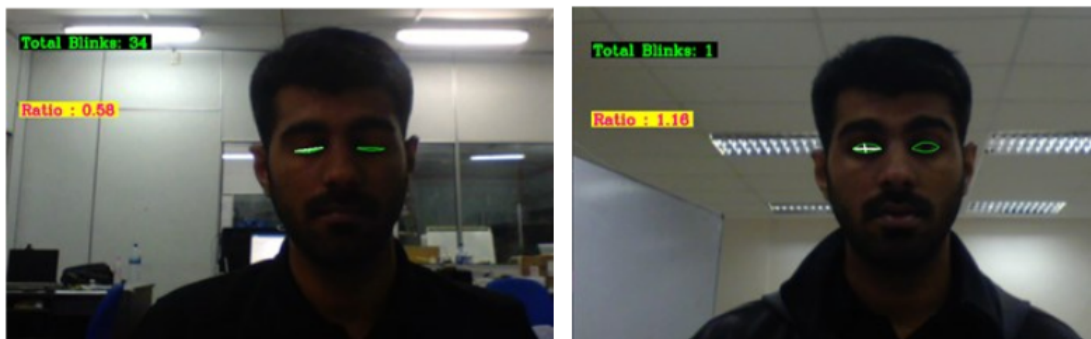


Figura 9. Resultados del modelo de parpadeo de ojos. Tomada de (Soomro et al., 2023).

Los oradores fueron capaces de detectar una pequeña diferencia entre los parpadeos cortos y largos, siendo que los parpadeos más largos provocaron respuestas significativamente más breves por parte de los voluntarios. En un sentido más amplio, este descubrimiento puede ayudarnos a comprender los orígenes de cómo los seres humanos comunican sus estados mentales, lo cual ha evolucionado hasta convertirse en un componente importante de las interacciones sociales cotidianas. En este algoritmo, se ha utilizado un enfoque diferente para calcular los parpadeos oculares, esto mediante el cálculo de los parpadeos de los ojos utilizando el algoritmo Eye Aspect Ratio (EAR), ver Figura 9.

Primero, se utilizan algoritmos de visión por computadora para detectar la cara y luego identificar puntos de referencia específicos alrededor de los ojos. Estos puntos de referencia suelen ser detectados por un modelo de aprendizaje profundo preentrenado, que puede identificar partes específicas de la cara, incluidos los contornos de los ojos.

El EAR es un valor escalar que refleja la proporción de la apertura del ojo. Se calcula utilizando la distancia vertical entre los puntos de referencia del párpado superior e inferior y la distancia horizontal entre los puntos de referencia del raballo del ojo.

3.5.3. Proxémica

La proxémica es el estudio del proceso dinámico mediante el cual las personas se posicionan en encuentros sociales cara a cara (Hall, 1973). Las personas utilizan señales proxémicas, como la distancia, la postura, la orientación de las caderas y los hombros, la posición de la cabeza y la mirada de los ojos, para comunicar un interés en iniciar, aceptar, mantener, terminar o evitar la interacción social (Mead et al., 2011). A lo largo del tiempo, se han desarrollado varias técnicas y enfoques en el estado del arte para reconocer y responder a la proxémica en estas interacciones.

En el trabajo de (Mead & Mataric, 2014) se realizó una recopilación de datos para modelar un controlador proxémico probabilístico basado en datos multimodales, esto con la finalidad de inferir información sobre variables latentes (por ejemplo, los niveles de entrada de gestos y de habla percibidos de una persona) basándose en datos detectados a partir de la proxémica, y utiliza estas inferencias para seleccionar parámetros para cambiar el comportamiento de un robot social de acuerdo a las condiciones en las que se encuentre, ver Figura 10.

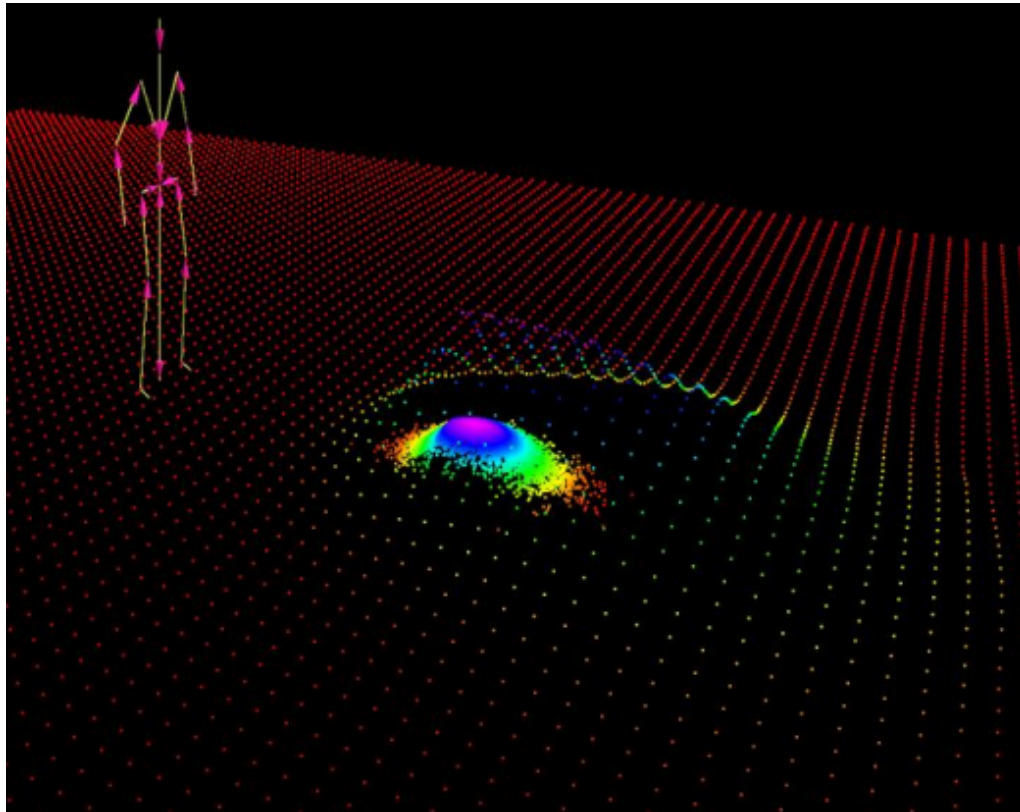


Figura 10. Controlador proxémico autónomo utilizando un enfoque basado en muestras para estimar cómo los agentes interactuantes experimentarían una interacción social basada en modelos en el marco proxémico socialmente situado; filtro de partículas (con remuestreo) utilizado para maximizar el desempeño social de todos los agentes que interactúan. Tomada de (Mead & Mataric, 2014).

En el trabajo de (Mead et al., 2013) se entrenaron Modelos Ocultos de Markov (HMM) para reconocer comportamientos espacio-temporales que indicaban transiciones hacia (iniciación) y fuera de (terminación) una interacción social. El comportamiento de iniciación intenta involucrar o reconocer a un posible compañero social en un discurso. El comportamiento de terminación propone el final de una interacción de manera socialmente apropiada. Estos comportamientos están dirigidos hacia un estímulo social (es decir, un objeto u otro agente) y ocurren secuencialmente o en paralelo con respecto a cada estímulo.

3.5.4. Detección y seguimiento de objetos

La detección y seguimiento de objetos es una característica básica pero fundamental en muchas tareas de robótica, incluyendo las interacciones entre humanos y robots. La visión por computadora y el procesamiento de imágenes están involucrados en esta tecnología informática y se utilizan ampliamente. En el camino para aprovechar el poder de la visión, se han desarrollado numerosos algoritmos, desde la detección de bordes simples hasta la detección de objetos a nivel de píxel.

Los robots deben ser capaces de detectar el objeto de interés, localizarlo y seguir sus movimientos para sincronizar temporal y espacialmente la fase de transición. En los últimos años, una variedad de arquitecturas basadas en redes neuronales convolucionales, especialmente los modelos R-CNN y YOLO (You Only Look Once), han contribuido en gran medida a mejorar la precisión y eficiencia en la detección.



Figura 11. Detección y seguimiento de objetos usando el robot NAO. Tomada de (Zhou et al., 2018).

En un trabajo realizado por (Zhou et al., 2018) se diseñó un sistema de detección visual para que el robot Nao detecte y siga algunos objetos basado en la arquitectura YOLO. En este trabajo se adaptó la red YOLO Tiny como un modelo preentrenado y se modificó la última capa completamente conectada para la detección de objetos de tres clases (taza, pluma, palo).

El modelo fue entrenado utilizando un conjunto de datos que constaba de 4322 imágenes de entrenamiento y logró obtener un promedio de precisión (mAP) del 44.3 % en el conjunto de pruebas. Posteriormente, este modelo se implementó en una función ampliamente reconocida del sistema de visión del robot Nao: la detección y localización de puntos de referencia.

Además, este sistema también se aplicó a objetos cotidianos, y los resultados indican que puede contribuir eficazmente a la localización y seguimiento de objetos de interés, como medicamentos o objetos peligrosos, en situaciones de interacción entre humanos y robots, como se ilustra en la Figura 11.

3.5.5. Señales fisiológicas y expresiones faciales

El reconocimiento de señales fisiológicas en la interacción humano-robot implica la identificación y el análisis de las respuestas fisiológicas del ser humano durante la interacción con un robot. Estas señales fisiológicas pueden proporcionar información valiosa sobre el estado emocional, cognitivo y físico del usuario, lo que permite al robot adaptar su comportamiento de manera más efectiva (Tapus & Mataric, 2008).

En un trabajo realizado por (Val-Calvo et al., 2020) se propone un algoritmo de aprendizaje profundo en el cual consiste en reconocer emociones a través de señales fisiológicas y expresiones faciales, utilizando una red neuronal convolucional (CNN).

El conjunto de datos empleado fue FER-2013 (Dumitru et al., 2013) el cual consistía de 3 subconjuntos que contienen imágenes de 48×48 píxeles: 28709 imágenes destinadas al entrenamiento, 3589 imágenes para la validación y 3589 imágenes para las pruebas. Todas las imágenes incluyen la siguiente etiqueta: 0 para enojo, 1 para disgusto, 2 para miedo, 3 para felicidad, 4 para tristeza, 5 para sorpresa y 6 para neutral.

Las señales fisiológicas que se recopilaron en el trabajo de (Val-Calvo et al., 2020) fueron: electroencefalograma (EEG), respuesta galvánica de la piel (GSR), y el pulso de volumen sanguíneo (BVP). Los

pasos de procesamiento involucran una serie de etapas para obtener intervalos entre latidos libres de ruido para codificar adecuadamente las propiedades de la señal. En primer lugar, se calcula el promedio móvil sobre los datos en bruto, donde se seleccionan regiones de interés cuando la amplitud de la señal es mayor que el promedio móvil.

Se marcan los picos R en el máximo de cada región de interés, lo que permite calcular la serie de intervalos entre latidos (intervalo de tiempo entre dos picos R sucesivos de latidos cardíacos). Posteriormente, se realiza la detección y el rechazo de valores atípicos, y derivado de esto, se procede a realizar la extracción de características de las señales preprocesadas para así poderlas meter a su modelo de clasificación.

La estimación de expresiones faciales se logra mediante una combinación de pasos en dos etapas (Ver Figura 12). En la primera etapa, se realiza la detección facial para simplificar la inferencia de la estimación de emociones. Esto se logra mediante el uso de un modelo de aprendizaje profundo convolucional (Vigil et al., 2019) que puede funcionar con restricciones en tiempo real.

La cara detectada se preprocesa de la siguiente manera: la imagen se recorta para extraer la región de interés, se convierte de RGB a escala de grises, se redimensiona a una resolución de 48×48 píxeles y, finalmente, se normaliza en un rango de $[0, 1]$.

En la segunda etapa, la imagen preprocesada se introduce en una capa de extracción de características de bajo nivel y en un conjunto de redes neuronales convolucionales profundas para obtener la clasificación de la emoción (Benamara et al., 2021). Este modelo de conjunto se entrenó en la base de datos FER-2013, logrando una precisión del 72,47 % en el conjunto de prueba.

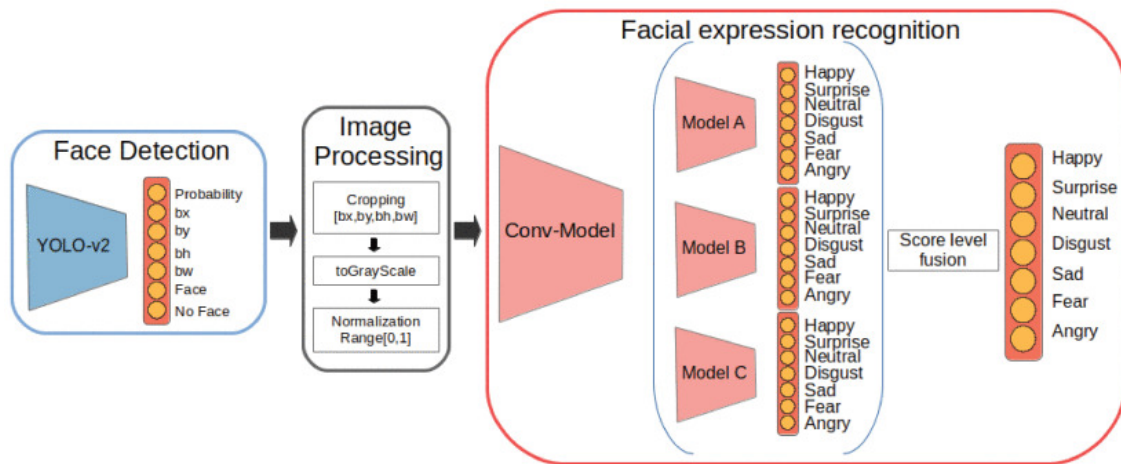


Figura 12. Estimación de expresiones faciales. En la primera etapa, detección de rostros mediante un modelo de aprendizaje profundo convolucional y preprocesamiento del rostro detectado. En la segunda etapa, extracción de características y clasificación mediante un conjunto de modelos de aprendizaje profundo convolucional. Tomada de (Val-Calvo et al., 2020).

Capítulo 4. Arquitectura de interacción no verbal SANI-HRI

En este capítulo se presenta la arquitectura de software **SANI-HRI** (Software Architecture for Nonverbal Interaction in Human-Robot Interaction) desarrollada durante el trabajo de tesis y se describen los componentes que la integran.

En la primera sección se introduce al robot social EVA con el que se trabajó durante el diseño y desarrollo de la arquitectura. En la segunda sección se presenta información sobre los sensores vestibulares y ambientales que fueron utilizados para recopilar los datos de los usuarios que posteriormente se utilizaron para detectar ciertos comportamientos no verbales. En la tercera y cuarta sección se describen el broker MQTT (Message Queuing Telemetry Transport) que se implementó para recibir y transferir los datos desde los dispositivos hacia el EVANotebook, y la plataforma Rhasspy ¹ diseñada para el procesamiento de lenguaje natural y reconocimiento de voz. En la quinta sección se presentan los modelos preentrenados y de lenguaje que se utilizaron para llevar a cabo las interacciones con los usuarios en los escenarios. Por último, se mencionan las conclusiones de la arquitectura desarrollada.

4.1. Robot social EVA

EVA ² (Embodied Voice Assistant) es una plataforma robótica de hardware abierto caracterizada por su diseño modular y sus componentes de bajo coste (Ver Figura. 13). Esta plataforma se desarrolló originalmente como robot asistente social para interactuar y asistir a adultos mayores que padecen demencia (Cruz-Sandoval et al., 2020). EVA se ha utilizado como plataforma para otros experimentos relacionados, como la asistencia a niños con autismo en la regulación de emociones (Rocha et al., 2021) y para apoyar en la detección de comportamientos alimentarios disruptivos de las personas con demencia (Astorga et al., 2023).

La plataforma robótica EVA engloba una estructura de hardware fundamental que facilita la implementación de la interacción y las funcionalidades descritas en este trabajo de investigación. Su arquitectura modular permite el desarrollo de diversas interacciones utilizando un número mínimo de sensores y dispositivos. Estos sensores y dispositivos son fundamentales para establecer una interacción fluida con el robot. Es importante enfatizar que todo el procesamiento que se realizaba anteriormente era *in situ*, y esto era un problema debido a que los componentes no tenían la potencia adecuada para soportar tanto

¹<https://rhasspy.readthedocs.io/en/latest/>

²<https://EVA-social-robot.github.io/>

nivel de computo. Sin embargo, una de las mejoras a partir de este trabajo de tesis es que se reduzca la carga de procesamiento del procesador Raspberry, ya que la arquitectura nos otorga la ventaja de usar una computadora externa para realizar ahí todo el procesamiento.

La versión estándar incluye un arreglo de micrófonos y un altavoz para la interacción por voz, y una pantalla táctil de 5 pulgadas que se utiliza para mostrar expresiones faciales, como atención o emociones, pero ocasionalmente animaciones o vídeos cortos. Además, un anillo LED RGB situado en el torso podría mostrar animaciones luminosas para indicar el estado del robot, como cuando se encuentra hablando o escuchando.

La versión intermedia (Ver Figura 13) agrega dos servomotores que proporcionan 2 grados de libertad (DOF) en la cabeza del robot y una cámara de profundidad para habilitar componentes de visión por computadora, como la detección de usuarios y el reconocimiento de actividades.

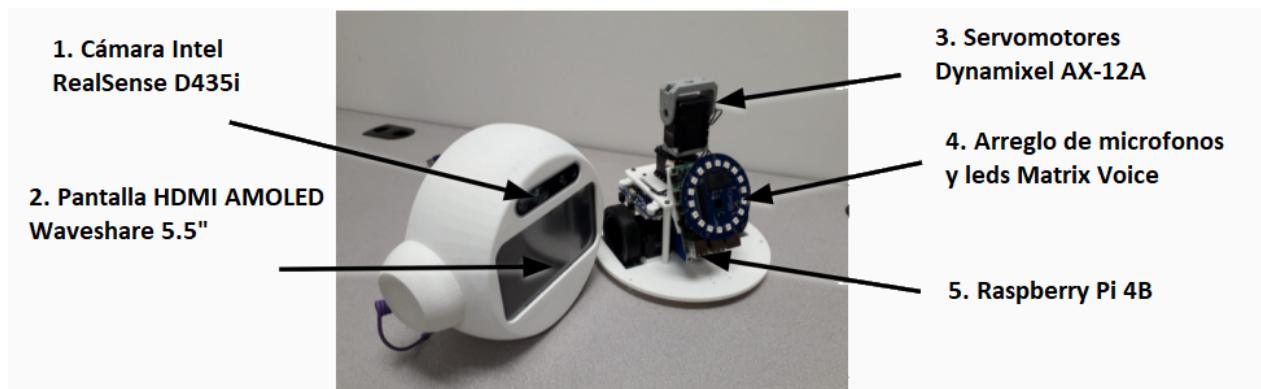


Figura 13. Componentes hardware del robot social EVA.

La cámara de profundidad Intel RealSense D435i ³, que se puede observar en la imagen anterior, se puede utilizar para reconocer al usuario y algunos comportamientos no verbales, como un gesto o expresión facial.

Además, se utiliza un arreglo de micrófonos (MatrixVoice) que permite capturar la voz y otros sonidos ambientales que ocurran durante una interacción dada. Con los motores Dynamixel AX-12A, EVA consigue dos grados de libertad, lo que le permite realizar movimientos de la cabeza hacia arriba y hacia abajo y girar de izquierda a derecha.

Finalmente, se utiliza una pantalla Waveshare para mostrar una expresión facial. Esta pantalla también puede ser utilizada para mostrar al usuario imágenes, vídeos o animaciones.

³<https://www.intelrealsense.com/depth-camera-d435i/>

4.2. Sensores vestibles y ambientales

Los sensores ambientales y portátiles integrados en el usuario y/o en el entorno, se utilizan para recopilar datos. Al captar y analizar los datos, estos sensores proporcionan información valiosa sobre las emociones humanas, las señales físicas y el contexto ambiental. Aprovechando esta información, EVA puede ofrecer interacciones personalizadas y adaptables, fomentando una serie de beneficios para las personas. Algunas de las ventajas que ofrece a las interacciones múltiples son:

- **Comprensión emocional.** Analizando las señales fisiológicas y las expresiones faciales, EVA puede percibir el estado emocional de una persona y responder en consecuencia.
- **Adaptación contextual.** Los sensores ambientales, incluidos micrófonos y cámaras, permiten al robot EVA percibir y responder al entorno que le rodea.
- **Capacidades de apoyo y asistencia.** Los datos captados por los sensores portátiles y ambientales pueden ser utilizados por EVA para ofrecer apoyo y asistencia a las personas. Por ejemplo, si EVA detecta un aumento de la frecuencia cardíaca o signos de estrés a través de los sensores, puede ofrecer técnicas de relajación, ejercicios de respiración o música relajante para aliviar la ansiedad.
- **Compartición e integración de datos.** La capacidad de EVA para entregar la información capturada por estos sensores a través de un protocolo de comunicación a otros dispositivos o algoritmos abre oportunidades para un mayor análisis e integración de servicios.

En la Figura 14 se presentan los diferentes dispositivos que se utilizaron para la captura de datos de los usuarios mientras se interactuaba con ellos en los diferentes escenarios:



Figura 14. Sensores vestibles y ambientales utilizados en el proceso de la captura de datos durante las interacciones dadas con el robot EVA. a) Sensor portátil Polar H10 para monitorear ritmo cardíaco. b) Cámara Intel RealSense D435i integrada en el robot EVA. c) Arreglo de micrófonos y leds Matrix Voice para el reconocimiento de voz integrados en el robot EVA.

A continuación, se proporciona una explicación detallada sobre el uso de tres sensores mencionados en la imagen anterior:

El sensor **Polar H10**⁴ es un dispositivo vestible diseñado principalmente para medir la frecuencia cardíaca de un usuario. Este fue utilizado para monitorear la respuesta fisiológica del usuario mientras interactuaba con el robot EVA. Esto puede ayudar a evaluar el nivel de estrés, la ansiedad o la emoción del usuario durante la interacción. Este sensor se colocó en el pecho del usuario y se conecta a través de Bluetooth a un dispositivo de registro de datos. Los datos de la frecuencia cardíaca se recopilan continuamente y se almacenan para su posterior análisis.

La cámara **Intel RealSense D435i**⁵ es un sensor ambiental 3D que incluye una unidad de medición inercial (IMU) para rastrear el movimiento y la orientación del sensor. Se utiliza para capturar información sobre la posición y el movimiento del usuario, donde en este caso, se utilizó para el seguimiento de gestos o movimientos del usuario durante alguna interacción.

Por último, el **Matrix Voice**⁶ es un sensor ambiental diseñado para la detección de voz y el procesamiento de audio. Se emplea para mejorar la comunicación entre el robot y el usuario a través del reconocimiento de voz y la generación de respuestas de audio. Esto permite que el robot entienda y responda a los comandos verbales del usuario. Este dispositivo está integrado en el robot para capturar señales de audio del entorno, para posteriormente procesarlas utilizando algoritmos de reconocimiento de voz.

4.3. Broker MQTT

Dado que nuestra plataforma sigue una arquitectura basada en eventos (sucesos o cambios significativos en el estado del hardware o el software de un sistema), necesitamos un intermediario y un protocolo para distribuir tales eventos.

Hoy en día, el protocolo estándar para internet de las cosas (IoT) y computación ubicua es MQTT (Message Queuing Telemetry Transport), un protocolo de mensajería ligero diseñado para la comunicación eficiente entre dispositivos, especialmente en entornos restringidos con ancho de banda y recursos limitados (Yassein et al., 2017).

⁴<https://www.polar.com/en/sensors/h10-heart-rate-sensor>

⁵<https://www.intelrealsense.com/depth-camera-d435i/>

⁶<https://matrix-io.github.io/matrix-documentation/matrix-voice/overview/>

Hemos elegido Eclipse Mosquitto⁷ un broker de mensajes de código abierto que implementa el protocolo MQTT. Un corredor de mensajes actúa como intermediario para la comunicación entre dispositivos o aplicaciones mediante la recepción y distribución de mensajes. Mosquitto actúa como broker para MQTT, permitiendo a los dispositivos publicar mensajes (enviar datos) y suscribirse a temas (recibir datos) dentro de un modelo de mensajería publicar-suscribir (Ver Figura 15).

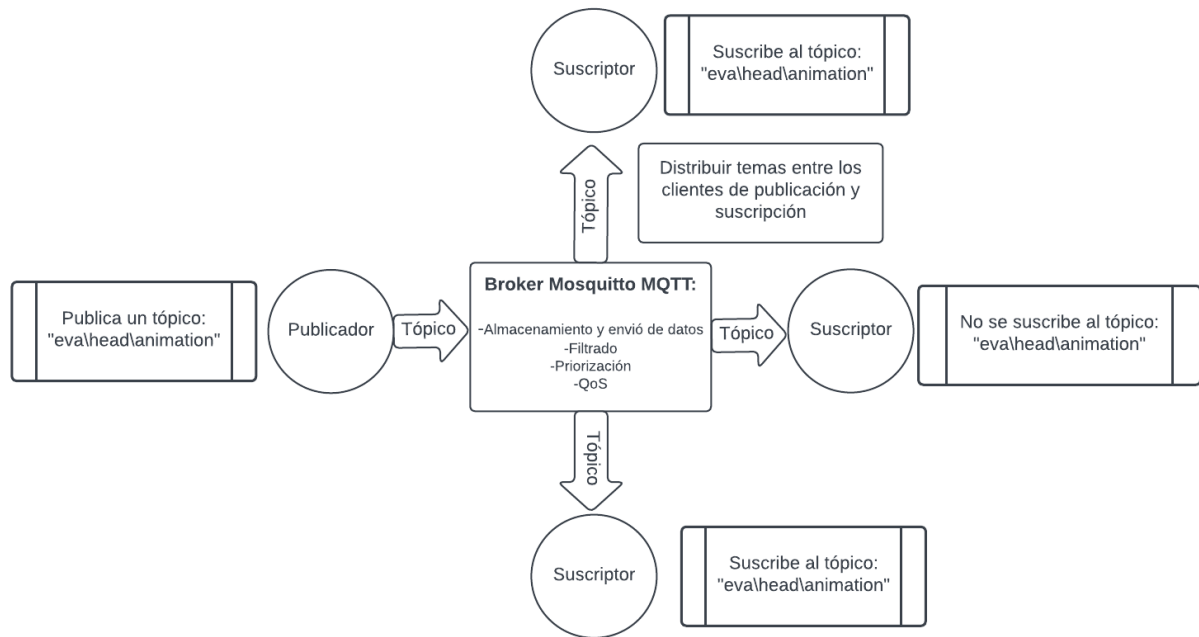


Figura 15. Arquitectura publicador/suscriptor del protocolo MQTT.

Para entender como funciona la comunicación entre dispositivos que estén conectados entre sí mediante el protocolo MQTT, se muestra en el diagrama anterior un breve ejemplo de su funcionamiento. En este caso, los clientes serán dispositivos o aplicaciones que desean intercambiar mensajes entre sí, los temas (tópicos) son cadenas de texto que actúan como canales o categorías de mensajes.

Esto habilita que los clientes pueden publicar mensajes en un tema en específico y suscribirse a uno o varios temas de su interés, como en este caso es **EVA\head\animation**. Así, cualquier cliente que este conectado a este tópico, tendrá acceso a los datos almacenados en esa categoría, de otra manera, no tendrá la posibilidad de acceder a ellos.

Para este caso, se eligió MQTT ya que es una elección sólida para este tipo de aplicaciones que requieren una comunicación eficiente, con baja latencia y capacidad de manejar condiciones de conectividad inestable.

⁷<https://mosquitto.org/>

4.4. Rhasspy

Rhasspy ⁸ es un conjunto de servicios de asistente de voz de código abierto, centrado en la privacidad, que permite a los usuarios crear sus propias aplicaciones controladas por voz. Proporciona una forma de crear interfaces de voz personalizadas que pueden utilizarse para interactuar con diversos dispositivos y servicios en un entorno local. Una de las principales características de Rhasspy es su énfasis en la privacidad. Por defecto, Rhasspy funciona totalmente offline, lo que significa que todos los datos de voz y su procesamiento permanecen dentro de la red local del usuario. Este enfoque de procesamiento local ayuda a proteger la privacidad del usuario al evitar la necesidad de enviar los datos de voz a servidores externos para su procesamiento (Filipe et al., 2021).

A continuación, se muestra en la Figura 16 la interfaz web que nos ofrece Rhasspy para manipular los servicios que tiene integrado:

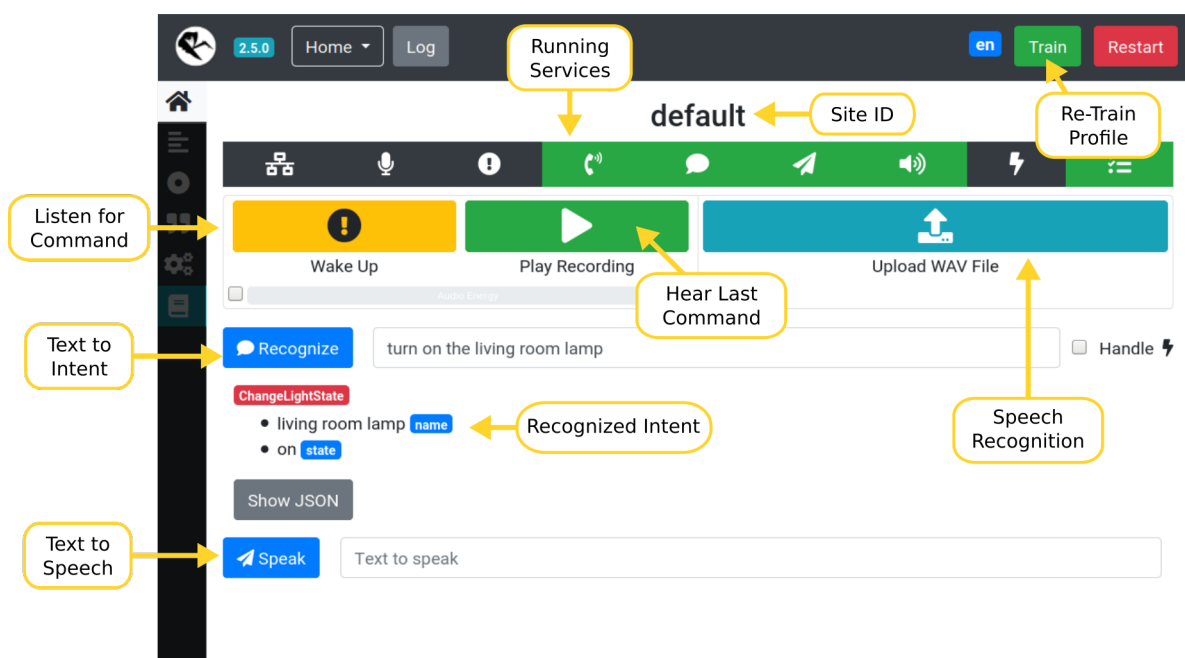


Figura 16. Interfaz web del sistema Rhasspy.

Esta plataforma admite varios protocolos de comunicación, incluidos MQTT y el protocolo de transferencia de hipertexto (HTTP), que se utilizaron en el desarrollo de este proyecto. Además, nos ofrece servicios de procesamiento de lenguaje natural (NLP), texto a voz (TTS), reconocimiento de voz, entre otros más. Estos servicios dotan al robot de capacidades para interactuar con un usuario de manera más natural y sofisticada.

⁸<https://rhasspy.readthedocs.io/en/latest/>

4.5. EVAnotebook

EVAnotebook⁹ es un cuaderno computacional o interfaz de cuaderno como Jupyter o Amazon SageMaker diseñado específicamente para operar únicamente dentro de un entorno de navegador y trabajar con diferentes kernels (JavaScript, Node, HTML, SQL y Python) de manera concurrente, sin necesidad de una arquitectura cliente-servidor pero sí aprovechando al máximo las API disponibles en el navegador. Al ser diseñado para desarrollar prototipos en arquitecturas dirigidas por eventos con la tradicional interactividad de escribir en solo documento código, resultados y gráficos, coincide con las características y necesidades de la arquitectura de EVA. Para lograr esta parte, EVAnotebook utiliza una base de datos descentralizada e incorpora varios protocolos de aplicación con un nivel mayor de abstracción, incluidos WebRTC, WebSockets y MQTT; además de incluir librerías que habilitan la programación reactiva como RxJS.

Así, EVAnotebook nos dio la habilidad de crear, la arquitectura de la presente tesis gracias a las características mencionadas, una explicación del código en Español y garantizar la reproducibilidad de la presente investigación. Por ejemplo, el desarrollador puede escribir una función que escucha el ambiente vía MQTT, importa su modelos de aprendizaje automático y ejecuta alguna acción con el mismo protocolo. El mismo puede colaborar con otros desarrolladores y compartirlo a través de internet. Cada una de las funciones corre de manera concurrente con WebWorkers, los cuales son subprocesos del navegador. A continuación, se muestra un ejemplo de un pseudocódigo para ejecutar alguna acción en el EVAnotebook:

```
begin
  // Este codigo es ejecutado en una celda .
  // Previamente importamos bibliotecas y creamos funciones personalizadas
  // en una celda
  // Escuchamos cambios del ambiente
  listen('Mqtt', 'stream').pipe(
    // Mapea los valores de la senal a otra
    map(x => llamarFuncionPersonalizada(x)),
    // Ignora o acepta valores que cumplen el predicado
    filter(x => predicado(x)),
    // Ejecuta alguna accion en el sistema publicando un topico
    tap(_ => mqtt.publish('topic'))))
end
```

⁹<https://notebook.sanchezcarlosjr.com>

Después de explicar los componentes de la arquitectura, pasaremos a explicar con más detalle su funcionamiento, abstrayendo los detalles de implementación para centrarnos en las características que proporciona desde la perspectiva de un discurso de **dominio**. En este sentido, la competencia de nuestra arquitectura se refiere al dominio (es decir, escenarios, comportamientos y tareas no verbales) sobre el que puede operar el sistema. Por su parte, el rendimiento depende de aspectos específicos de la aplicación, como la elección de la tecnología y los algoritmos utilizados.

Nuestro sistema se compone de microservicios que colaboran a través de flujos. Está inspirado en el trabajo de (Pineda et al., 2015) pero con énfasis en estrategias distribuidas y basadas en datos. La computación ubicua implica algoritmos distribuidos que se ejecutan en múltiples nodos interconectados sin un control centralizado rígido, y algoritmos en línea (así como algoritmos de flujo y dinámicos), que reciben su entrada de forma incremental a lo largo del tiempo en lugar de tener toda la entrada disponible desde el principio. Esto los asemeja a la lógica no monotónica.

Además, la programación reactiva introduce una semántica operativa que reacciona ante eventos externos y gestiona de forma transparente los flujos de datos relacionados con el tiempo, como el manejo de señales asíncronas del entorno.

La arquitectura de software propuesta para la interacción humano-robot se basa en eventos y emplea una combinación de modelos de publicación/suscripción a través de varios protocolos y capas de comunicación. En este contexto, un **event** se define como un objeto inmutable, denotado como la evaluación de la señal, $s(t_0)$, o el cambio en el estado del sistema, originado por un subsistema. En incrementos de tiempo discretos, este objeto produce un **stream**, la señal $s(t)$. Un productor genera un evento que múltiples consumidores pueden procesar potencialmente.

El sistema de mensajería puede funcionar de tres maneras: (i) los productores envían mensajes directos sin nodos intermediarios, (ii) el productor envía un mensaje al consumidor a través de un intermediario de mensajes, o broker, y (iii) a través de registros de eventos. A efectos de este trabajo, denominaremos microservicios tanto a los productores como a los consumidores. A diferencia del software monolítico, los microservicios pueden desplegarse de forma independiente y están débilmente acoplados. A grandes rasgos, se clasifican en sincronizadores de datos, tareas y etapas de canalización.

Los sincronizadores de datos tienen la responsabilidad de almacenar o presentar los datos en un sistema de almacenamiento. Las tareas envían eventos que afectan al entorno. Las etapas de canalización reciben eventos de múltiples flujos de entrada para producir uno o más flujos de salida que pasan por diferentes etapas hasta que llegan a una tarea o a sincronizadores de datos que aplican operaciones de canalización

y filtrado. Por lo tanto, un servicio es un grafo de dependencias dentro de microservicios unidos por diferentes protocolos, que está diseñado para lograr una competencia específica.

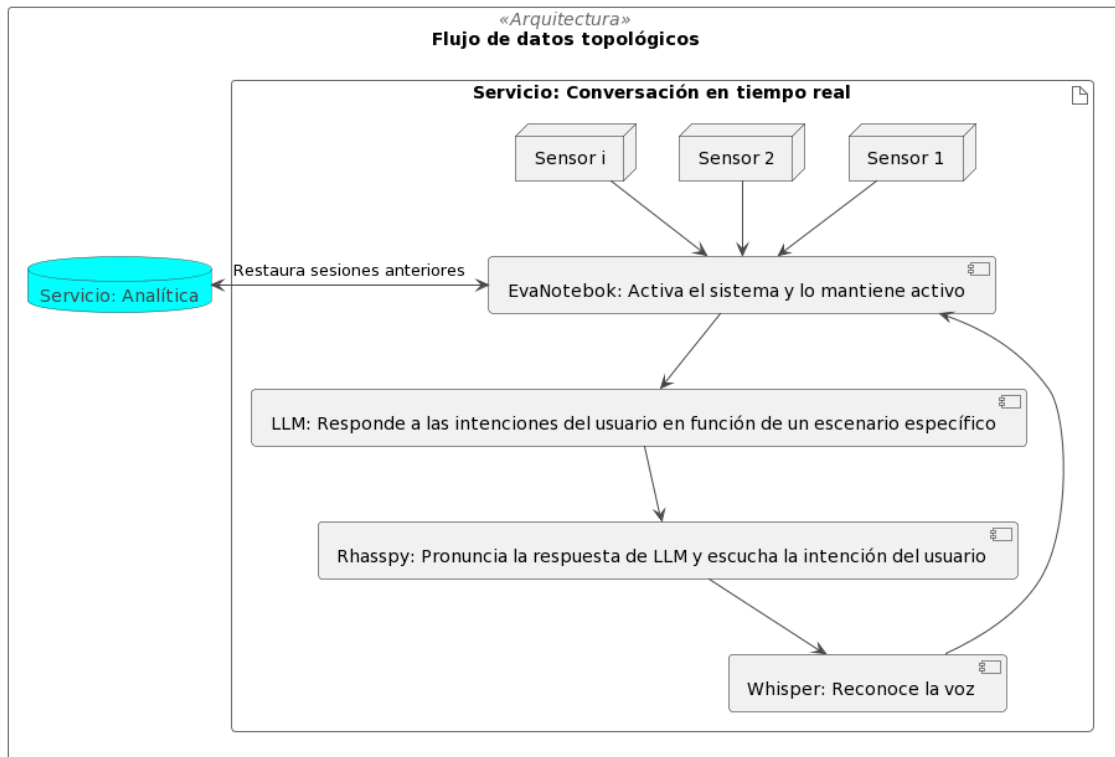


Figura 17. Ejemplo de flujo de datos en nuestra arquitectura basada en eventos

La Figura 17 ilustra el flujo de datos dentro de un ejemplo de servicio, donde los nodos simbolizan microservicios y las aristas representan protocolos de comunicación. Una conversación en tiempo real, que es una competencia específica, con un robot social implica entender la comunicación no verbal para despertar o dormir al sistema. En lugar de *wake words* tales como **"Hola EVA"** para inicializar el sistema, el resto del sistema permanece en gran medida inoperativo hasta que tenemos pruebas suficientes para tomar la decisión de volver a activarse, por ejemplo, al momento que se detecte alguna persona en el ambiente, el sistema inicializara al robot EVA. Además, una conversación implica que el usuario gestione sesiones, en las que el robot social puede recordar la voz, la cara y otras características del usuario para interactuar de la mejor manera adaptada a ese usuario.

Cuando EVAnotebook despierta el sistema basándose en sensores, análisis o la intención del usuario, personaliza el Large Language Model (LLM) para responder basándose en un escenario específico inyectando un prompt del sistema y restaurando sesiones anteriores. A continuación, Rhasspy vocaliza la respuesta del LLM y escucha la intención del usuario. Whisper es un modelo basado en redes neuronales desarrollado por OpenAI para resolver tareas de conversión de voz a texto, esto fue utilizado para

devolver al EVAnotebook las transcripciones de las palabras que se reconocieron de los usuarios. Por lo tanto, formamos un bucle de retroalimentación en el que diferentes consumidores emiten eventos que necesitamos inyectar en el LLM y reaccionar en consecuencia. En segundo plano, el sistema registra datos para su posterior análisis. En el capítulo 5 se presentarán otro tipo de demostraciones que puede ser capaz de realizar la arquitectura en los escenarios definidos anteriormente.

4.6. Modelos de análisis de datos

Para tener la capacidad de detectar señales no verbales, la arquitectura puede hacer uso de modelos de visión por computadora e incluso de algoritmos de aprendizaje máquina, algunos de ellos son módulos de reconocimiento de expresiones faciales, corporales y de emociones. En el presente trabajo se han desarrollado varios EVAnotebooks para atacar diversos problemas en cuestión de reconocer la dirección de la mirada, la proximidad, el reconocimiento de objetos, la postura, etc.

Estos pueden adaptarse o personalizarse para el problema en cuestión y ejecutarse de forma sincrónica con otras tareas, ocasionando que se aproveche al máximo el rendimiento y flexibilidad que cuenta la arquitectura al incluir diversos componentes sin necesidad de hacer mucho esfuerzo. En la siguiente figura se muestra un ejemplo de un EVAnotebook creado para habilitar el servicio WakeFace:

```

1  connect("MQTT", {
2      hostname: "localhost",
3      port: 8081,
4      path: '/',
5      topic: 'eva/camera/headtilting',
6      protocol: "ws",
7      username: "eva_cicese",
8      password: "NQz4esJX$"
9  }).pipe(
10     chat(message =>
11         of(message).pipe(
12             filter(message => message === "F"),
13             map(_ => ({
14                 topic: "hermes/hotword/default/detected",
15                 message: serialize({
16                     "siteId": "default",
17                     "model_id": "default",
18                     "model_version": "",
19                     "model_type": "personal",
20                     "current_sensitivity": 1.0
21                 })
22             }))
23         )
24     )
25 )

```

Figura 18. EVAnotebook con código para detectar la dirección de atención del usuario.

En el ejemplo mostrado en la imagen anterior (Ver Figura 18) se puede observar como en las líneas 1-8 la conexión al protocolo MQTT dentro del tópico *EVA/camera/headtilting*, en el cual se realiza la transmisión de los datos recopilados a través de la cámara del robot EVA. Posteriormente, las líneas 10-20 se encargan de analizar el seguimiento de la orientación de la mirada de la persona y después habilitar el servicio del robot para iniciar con la interacción deseada. Se debe tener en cuenta que en la línea 12 se realiza la filtración de los mensajes que nos llegan, debido a que solo nos interesa aquellos con la letra "F", que indica la palabra Front (hacia adelante).

Finalmente, cuando se tienen los mensajes correctos que indican que la persona está mirando hacia el robot, en la línea 14 se realiza la conexión al tópico *hermes/hotword/default/detected* y el protocolo MQTT, esto para detectar alguna palabra clave con el arreglo de micrófonos Matrix Voice, y poder enviar cualquier mensaje al usuario desde el robot EVA, por medio del servicio Text-To-Speech que nos proporciona la plataforma Rhasspy.

4.6.1. MediaPipe

Es un framework de código abierto desarrollado por Google que proporciona un conjunto de bibliotecas y herramientas para construir e implementar técnicas de aprendizaje automático (ML) para varias tareas de procesamiento de datos, como análisis de imágenes y video, procesamiento de audio y reconocimiento de gestos (Lugaresi et al., 2019). Su objetivo es simplificar el desarrollo de aplicaciones multimedia que requieren procesamiento y análisis en tiempo real de datos visuales y auditivos. MediaPipe está diseñado para ser flexible y personalizable, lo que permite a los desarrolladores integrar sus componentes en sus aplicaciones y adaptarlos a sus necesidades específicas. El marco proporciona tanto modelos pre-entrenados como herramientas para entrenar modelos personalizados sobre nuevos datos. Soporta varios lenguajes de programación, incluyendo Python, C++ y JavaScript.

Los desarrolladores pueden utilizar MediaPipe para una amplia gama de aplicaciones, incluyendo realidad aumentada, realidad virtual, interacción basada en gestos, análisis de vídeo y mucho más. La naturaleza modular y extensible del framework lo convierte en una poderosa herramienta para construir aplicaciones multimedia innovadoras que aprovechan técnicas de aprendizaje automático y visión por computadora. En el Apéndice B se muestran los diferentes modelos que ofrece Mediapipe y que fueron utilizados en este trabajo de tesis para hacer reconocimiento facial, seguimiento de gestos y movimientos que se realizaron en los escenarios de interacción.

4.6.2. OpenCV

OpenCV (Open Source Computer Vision Library), es una popular biblioteca de visión y procesamiento de imágenes de código abierto (Pulli et al., 2012). Proporciona una amplia gama de herramientas, funciones y algoritmos para varias tareas relacionadas con la visión por computadora, el análisis de imágenes y el aprendizaje automático. OpenCV está escrito en C++ y tiene interfaces para varios lenguajes de programación, incluyendo Python, Java y más. Esta biblioteca fue usada en el presente trabajo para procesar las imágenes y vídeos que se capturaron de la cámara del robot EVA, en conjunto con la herramienta TensorFlow y Mediapipe fueron los que hicieron posible el reconocimiento de expresiones faciales y el seguimiento de movimientos que realizaban los usuarios.

4.6.3. Análisis de procrustes

Es una técnica matemática y estadística utilizada en diversos campos como geometría, estadística y análisis de datos para comparar y alinear formas o conjuntos de datos que podrían haber sufrido transformaciones de rotación, escala o reflexión (Ross, 2004). El objetivo del análisis de procrustes es encontrar una transformación óptima que alinee las formas o conjuntos de datos de la mejor manera posible, minimizando las diferencias entre ellas, permitiendo comparaciones y análisis significativos.

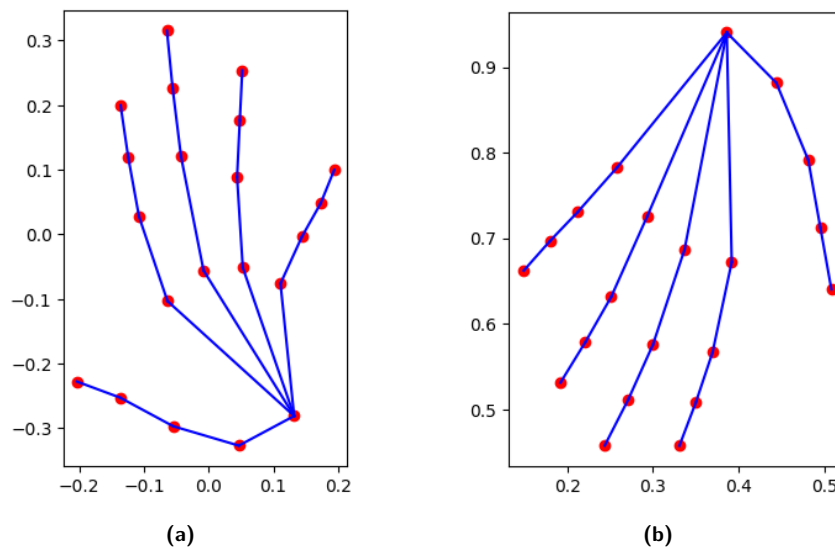


Figura 19. Representación de la forma de una mano a partir de la extracción de los puntos de referencia utilizando la técnica de análisis de procrustes. a) Imagen original de la forma de la mano. b) Análisis de procrustes aplicada a la imagen original para alterar sus características originales (rotación, traslación, escalado y normalización).

En la Figura 19 se puede observar el resultado que nos otorga esta técnica al momento de extraer los puntos más relevantes de la forma de la mano, para después aplicar el análisis de procrustes y plasmar la posición de la mano. En la Figura 19b se puede observar el resultado de aplicar esta técnica a la imagen original, dándonos una rotación y comparando la imagen modificada con la original.

En este trabajo se utilizó análisis de procrustes a partir de la librería **orthogonal_procrustes**¹⁰ de *scipy.linalg*, para diseñar posiciones ideales a partir de la extracción de puntos clave de una pose en específico, para así comparar la similitud o congruencia que presentaba con otros conjuntos de datos.

En comparación con utilizar algoritmos de aprendizaje máquina, esta técnica es mucho más rápida y fácil de usar, debido a que no se tiene que entrenar un modelo para realizar la clasificación de una imagen tomada del robot EVA. Esta técnica fue empleada para reconocer cuando una persona estuviera haciendo una seña con el brazo, como es el caso del escenario 3.

4.6.4. Whisper

Es un modelo de reconocimiento de voz de uso general. Está entrenado con un gran conjunto de datos a partir de diversos audios y es también un modelo multitarea que puede realizar reconocimiento de voz en varios idiomas, traducir las palabras capturadas por voz e identificar distintos idiomas (Radford et al., 2023). Este modelo fue utilizado en la arquitectura debido a su gran potencial para reconocer una variedad de palabras que son complejas de entender con otros tipos de modelos. Además, Whisper nos otorga la oportunidad de descargar diferentes tipos de modelos.

Existen diversos modelos que se pueden utilizar para el reconocimiento de voz, sin embargo, el que mejores resultados nos dio fue el *Medium*, debido a que está entrenado con 769 millones de parámetros y la velocidad de respuesta era rápida con respecto a la cantidad de segundos que se demoraba para transcribir las palabras reconocidas, en comparación con los demás modelos que nos daban muchos errores en la traducción de las palabras.

Es importante mencionar que, si bien se tuvieron algunos problemas con el tiempo de respuesta y la exactitud con la que nos daba las transcripciones, este tipo de modelo tiene un rendimiento competitivo en cuanto a su adaptabilidad a cualquier tipo de escenario en el que se utilizó.

¹⁰https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.orthogonal_procrustes.html

4.6.5. ChatGPT

Es un modelo de lenguaje desarrollado por OpenAI basado en la arquitectura GPT (Generative Pre-trained Transformer) (Zhu & Luo, 2022). Es una herramienta de procesamiento del lenguaje natural impulsada por que te permite mantener conversaciones similares a las humanas con la ayuda de un chatbot.

El propósito principal de ChatGPT es interactuar con los usuarios a través de conversaciones escritas. Puedes ingresar preguntas, comentarios o cualquier otro tipo de texto, y el modelo generará respuestas en función de lo que ha aprendido durante su entrenamiento. Aunque las respuestas pueden ser coherentes y contextualmente relevantes, es importante recordar que ChatGPT no tiene comprensión real ni emociones, solo simula respuestas basadas en patrones.

Durante el desarrollo del trabajo de tesis, se adaptó este modelo de lenguaje para mejorar la calidad de respuestas del robot EVA mientras está teniendo una interacción con una persona. Sin embargo, tiene sus desventajas debido a que no siempre nos arrojaba respuestas coherentes respecto al tema de conversación.

Se utilizó la API de ChatGPT-3.5 para tener acceso gratuito a las características que nos ofrece este modelo, teniendo en cuenta que muchas veces se tenían problemas con el tiempo que se tardaba en procesar el texto que se le mandaba al prompt para que nos regresara una respuesta, aunque hay que tener en cuenta que al ser una versión gratuita este tipo de problemas puede presentarse por la alta demanda que tiene, sin embargo, al utilizar una versión de paga es muy probable que estos errores disminuyan considerablemente.

4.6.6. LangChain

LangChain es un framework para desarrollar aplicaciones utilizando modelos de LLM's, y su objetivo es permitir a los desarrolladores utilizar convenientemente otros orígenes de datos e interactuar con otras aplicaciones (Topsakal & Akinci, 2023). Para activarlo, LangChain proporciona componentes (abstracciones modulares) y cadenas (pipelines personalizables para cada caso). Los principales componentes de LangChain, tales como mensajes, memoria, cadenas y agentes son explicados a continuación:

4.6.6.1. Prompts

Un «prompt» es la entrada a un LLM. Generalmente se generan dinámicamente cuando se utilizan en una aplicación LLM e incluye la entrada del usuario (pregunta), un conjunto de pocos ejemplos de disparos para ayudar al modelo de lenguaje a generar una mejor respuesta, e instrucciones para el LLM sobre cómo procesar la entrada que proviene del usuario. LangChain proporciona varias clases para crear mensajes utilizando varias plantillas de solicitud especializadas. Una plantilla prompt hace referencia a una forma reproducible de generar un mensaje. Contiene una cadena de texto («la plantilla») que puede tomar un conjunto de parámetros del usuario final y genera un mensaje.

4.6.6.2. Cadenas

El bloque de construcción clave más importante de LangChain es la cadena. La cadena generalmente combina un LLM junto con un prompt, y con este bloque de creación, también puede combinar los bloques de construcción para llevar a cabo una secuencia de operaciones en el texto o en los demás datos. Una cadena simple toma un indicador de entrada y produce una salida. Varias cadenas se pueden ejecutar una tras otra, donde la salida de la primera cadena se convierte en la entrada de la siguiente cadena.

4.6.6.3. Memoria

El modelo de lenguaje en sí no tiene estado y no recuerda las conversaciones. Cada transacción (llamada) al punto final API del LLM es independiente. La ilusión de la memoria en los sistemas de chatbot se ve facilitada por un código complementario, que incorpora el contexto de los diálogos anteriores al interactuar con el LLM. Se necesita un componente de memoria que pueda almacenar las conversaciones anteriores y pasarlas al LLM con el siguiente mensaje.

Las interacciones de un usuario con un modelo de lenguaje se capturan en el concepto de *ChatMessages*, durante este proceso se emplean funciones que extraen información de una secuencia de mensajes, para después guardarla en memoria, donde al final nos puede devolver múltiples datos (los N mensajes más

recientes y un resumen de todos los mensajes anteriores). La información devuelta puede ser una cadena o una lista de mensajes. En este trabajo, se integró este método en EVAnotebook para mantener un historial de las conversaciones que han tenido los usuarios con el robot EVA durante una interacción.

4.7. Conclusiones de la arquitectura

Hasta el momento, a lo largo del capítulo hemos presentado y descrito los componentes de la arquitectura de software, al igual que los modelos preentrenados utilizados para reconocer los diferentes tipos de comportamientos no verbales en IHR. La arquitectura debe ser diseñada considerando factores clave como la precisión en la detección de comportamientos no verbales, la adaptabilidad a las necesidades cambiantes de los usuarios y la privacidad en el manejo de datos sensibles.

En última instancia, la arquitectura de software debe ser escalable y actualizable para afrontar los desafíos emergentes en la interacción humano-robot, manteniendo un enfoque centrado en el usuario y en la mejora de la experiencia global. Con avances continuos en tecnología y la intersección de disciplinas, la arquitectura de software en este contexto promete ampliar los horizontes de la interacción humana con robots, facilitando la coexistencia y colaboración entre ambas entidades de manera más natural y fluida.

En resumen, el diseño de una arquitectura de software para detectar comportamientos no verbales en interacciones humano-robot requiere la convergencia de tecnologías avanzadas de visión, sensores biométricos, procesamiento de lenguaje natural y aprendizaje automático. La combinación de estos elementos permitirá al robot social EVA comprender e interpretar las señales no verbales de los usuarios, adaptando sus respuestas de manera empática y eficiente en escenarios diversos y en constante evolución.

En el siguiente capítulo describimos cómo son utilizados los componentes de esta arquitectura para implementar los escenarios descritos en el capítulo 2.

Capítulo 5. Implementación de escenarios

En este capítulo se presentan los diversos escenarios de interacción no verbal que se implementaron para evaluar la arquitectura desarrollada durante el trabajo de investigación. También se presentan los escenarios adicionales que surgieron a medida que se fue desarrollando la arquitectura, esto con el objetivo de evaluar su funcionamiento en experimentos con usuarios.

5.1. Escenarios principales propuestos para la arquitectura

En esta sección se describen los escenarios presentados en la sección 2.4 implementados usando los componentes de la arquitectura propuesta:

5.1.1. Escenario 1: Mirada y proxémica para iniciar una interacción

Este escenario se implementó en un EVAnotebook que extiende el microservicio "WakeFace" propuesto en (Villa et al., 2022a) como una alternativa al método *wakeword* para iniciar una interacción. La extensión consiste en utilizar la proximidad además de la dirección de la mirada para adaptar la interacción según los siguientes criterios:

1. **Distancia mayor a tres metros:** Cuando EVA detecta a una persona caminando a una distancia de más de tres metros (zona social/pública), el robot activa su función de seguimiento visual. EVA gira el cuello y dirige su mirada hacia la persona mientras ésta sigue moviéndose. El robot ajusta la velocidad y el rango de movimiento de su cuello para seguir suave y naturalmente el movimiento de la persona. Sin embargo, a esta distancia, EVA no inicia una interacción verbal ni hace preguntas.
2. **Distancia inferior a tres metros:** cuando la persona se acerca a aproximadamente tres metros de EVA (zona personal/social), el robot emite una señal auditiva suave para captar la atención de la persona y luego inicia una interacción verbal. Por ejemplo, podría decir "¡Hola! ¿Cómo estás hoy?". o "¡Bienvenido! ¿Puedo ayudarte en algo?".
3. **Servicio de una conversación en tiempo real:** Una vez que EVA ha iniciado la conversación, espera la respuesta del usuario. Si la persona responde positivamente o muestra interés, EVA

continúa el diálogo con fluidez y naturalidad, y en caso de que el usuario no muestre interés, se espera alrededor de unos 5 segundos para despedirse de él. También puede hacer preguntas de seguimiento para conocer los intereses o necesidades de la persona.

La Figura 20 muestra un diagrama de secuencia que describe la interacción del Escenario 1 presentado en la Sección 2.4.1. El robot social EVA inicia una interacción cuando detecta la presencia de un individuo a cierta distancia. Utiliza su cámara y un algoritmo de reconocimiento facial para inferir la dirección y la proximidad de la mirada.

Cabe destacar que el EVANotebook se conecta a diferentes modelos o librerías para realizar los cálculos necesarios para estimar la dirección y distancia de la persona. En este escenario en particular, la librería de TensorFlow se utilizó en combinación con OpenCV y Mediapipe para mejorar la precisión y estimación del rostro facial.

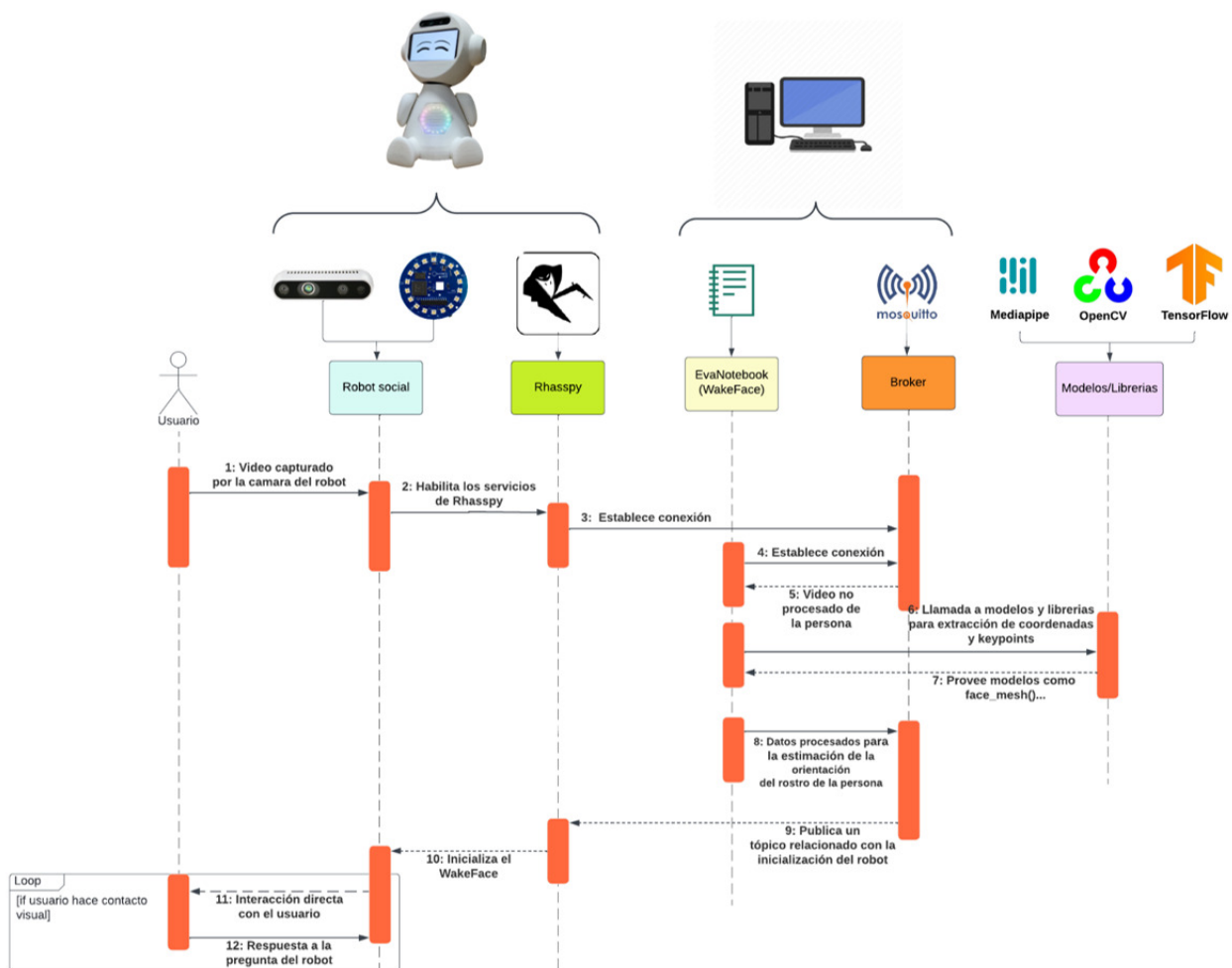


Figura 20. Robot EVA inicia proactivamente una interacción basada en la proximidad del usuario y la dirección de la mirada.

El siguiente código es un extracto del microservicio WakeFace escrito en EVAnotebook, específicamente en un WebWorker de JavaScript. Ilustra cómo el microservicio se suscribe a eventos MQTT relevantes. Los datos de la cámara obtenidos se envían luego a los módulos ML responsables de evaluar la dirección y la proximidad de la mirada.

Además, utiliza la semántica de RxJS y utiliza una arquitectura de tubería y filtro para manejar eventos asíncronos, lo que permite el procesamiento de transmisión a través de operadores.

```
begin
  // import mediapipe
  zip(
    listen('Mqtt', 'camera stream').pipe(
      map(frame => mediapipe.faceDetector.detect(frame).detections[0]),
      // Ignoramos los frames si no hay personas visibles.
      filter(landmarks => landmarks !== undefined),
      // Ejecutamos alguna prueba de estimacion de pose
      map(landmarks => testIfUserLooksAtCamera(landmarks))
    ),
    listen('Mqtt', 'camera stream').pipe(
      map(frame => mediapipe.faceLandmarks.detect(frame).detections[0]),
      filter(landmarks => landmarks !== undefined),
      map(landmarks => testIfUserIsADistanceLessThan3Meters(landmarks))
    )
  ).pipe(
    filter(
      ([userLooksAtCamera, userIsADistanceLessThan3Meters]) =>
        userLooksAtCamera && userIsADistanceLessThan3Meters),
    tap(_ => mqtt.publish('wakeup'))
  )
end
```

Mediapipe se utiliza para detectar puntos de referencia faciales. Posteriormente, el código estima si la persona está mirando al robot EVA. Por otro lado, calculamos la distancia del usuario con respecto a la cámara utilizando, por ejemplo, el modelo de cámara estenopeica. Luego combinamos ambos flujos para determinar si activar el sistema o no. Es importante mencionar que el algoritmo de reconocimiento facial puede funcionar correctamente o no, dependiendo de las condiciones del ambiente al momento de estar capturando los datos.



Figura 21. Estimación de la orientación del rostro a partir del modelo de Mediapipe face mesh.

La Figura 21 muestra los resultados de los cálculos para descubrir en qué dirección mira una persona frente al robot EVA. En este contexto, lo importante es cuando la persona está mirando directamente hacia el robot, que llamamos "Forward". Esto es clave para que el robot pueda comenzar a interactuar de manera efectiva con la persona, lo que hace que la comunicación entre ellos sea más fluida y natural.

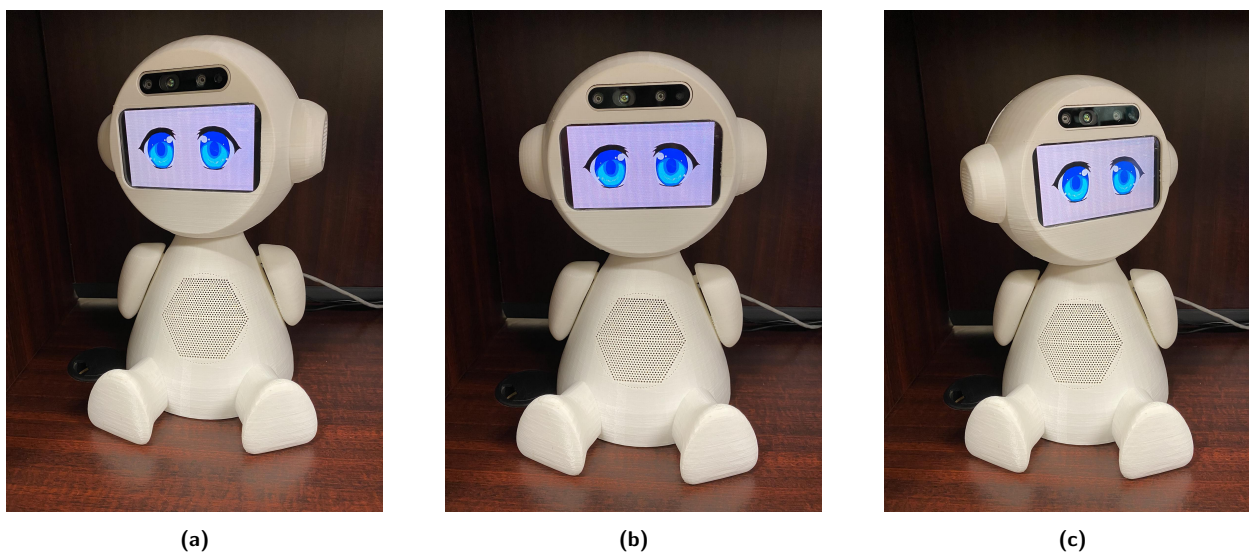


Figura 22. EVA realiza el seguimiento facial de una persona a través del movimiento de su cabeza para mantener un ángulo visible de la cámara. a) Girando su cuello hacia la derecha. b) Manteniendo su cuello en la posición inicial. c) Girando su cuello hacia la izquierda.

La Figura 22 ilustra la ejecución del movimiento del cuello del robot EVA en el proceso de seguimiento facial de una persona. Es relevante señalar que los servomotores empleados para controlar la inclinación de la cabeza no tienen restricciones de rotación, pero se ha establecido un límite en el rango de movimiento de 0 a 300 grados. Esta limitación se implementó con el propósito de evitar que el robot realice una rotación completa de la cabeza, ya que esto podría resultar incómodo o inapropiado para el usuario.

5.1.2. Escenario 2: Rutina de ejercicio

En la Figura 23 se muestra el diagrama de secuencia que describe la interacción del escenario 2 presentado en la sección 2.4.2. Cabe mencionar que en este escenario se implementa el uso de API's externas las cuales van a conectarse a los dispositivos vestibles que lleva el usuario consigo para la recopilación de sus datos.

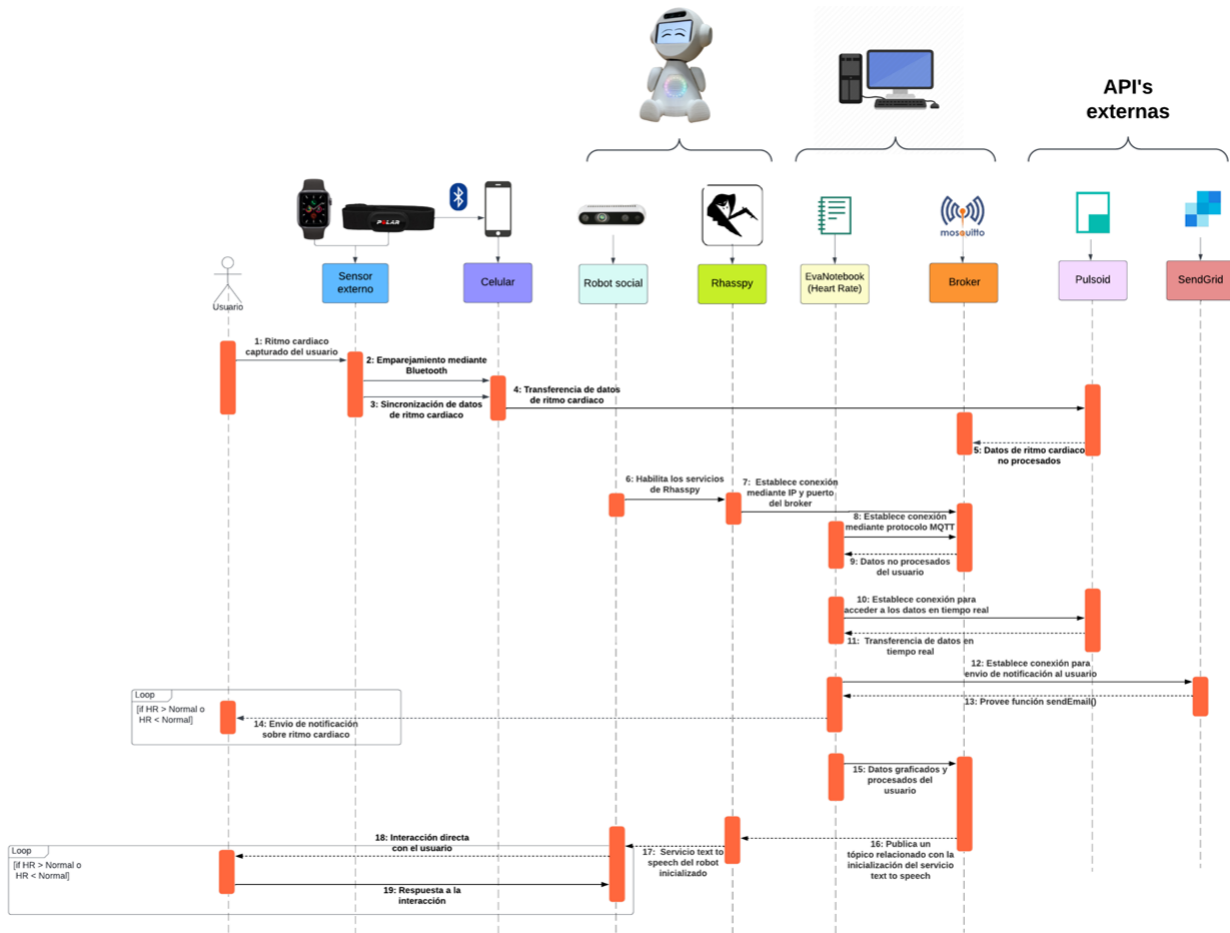


Figura 23. Diagrama de secuencia de un usuario realizando ejercicio y siendo monitoreado por el robot EVA.

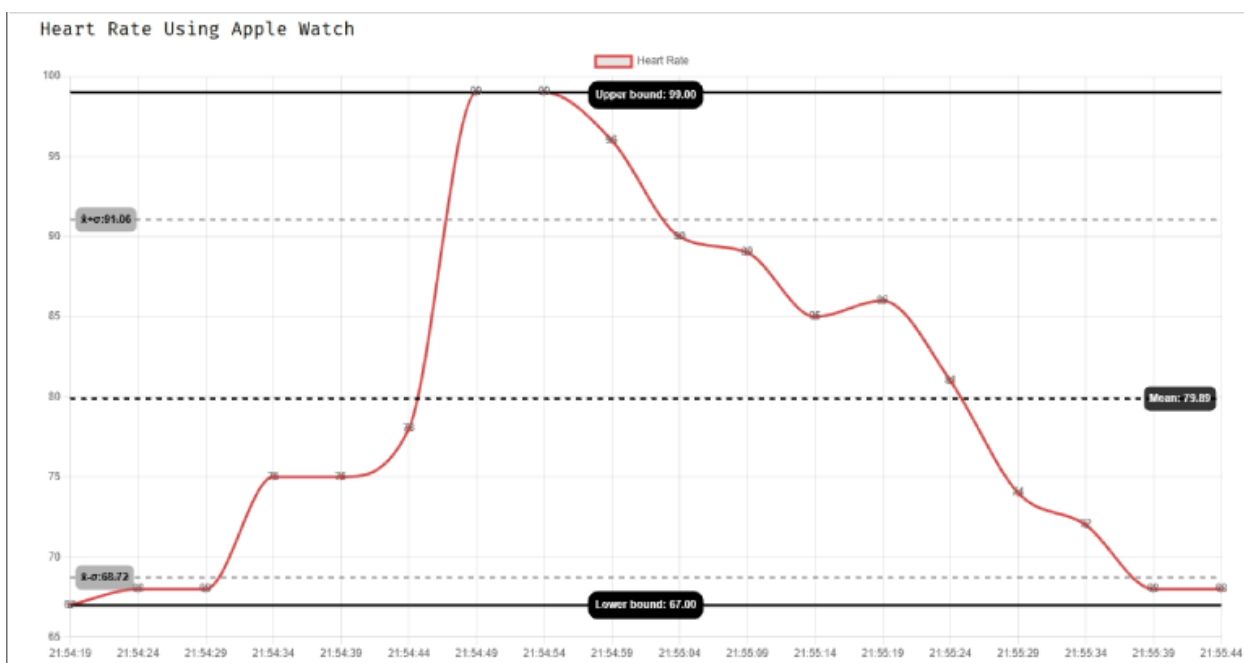
El robot social EVA iniciará con la interacción indicándole al usuario que rutina de ejercicio debe realizar. Posteriormente, al momento que el robot detecte que el rango establecido del ritmo cardíaco sobrepasa o disminuye de lo normal, tomará una decisión acerca de si cambiar la rutina o aconsejarle al usuario que tome un descanso. El usuario llevará puesto un sensor fisiológico que estará monitorizando su ritmo cardíaco durante la realización de una rutina de ejercicio, esto con la ayuda de la API Pulsoid que estará conectada a un celular mediante bluetooth y que a su vez, estos datos se estarán transfiriendo al EVAnotebook para realizar su debido procesamiento.

En el diagrama de secuencia de la figura anterior, se puede observar como es que el EVAnotebook también se puede conectar a la API llamada SendGrid, que se encargará de proporcionar las herramientas necesarias para que el notebook pueda enviar notificaciones por correo electrónico y por mensajes de texto, todo esto con la finalidad de mandar una alerta de notificación a algún cuidador o persona que tenga relación con el usuario.

Conectamos el notebook a los sensores fisiológicos Polar H10 y el Apple Watch mediante la API de Pulsoid, la cual nos entregará el ritmo cardíaco (BPM) medido en latidos por minuto del usuario. Asimismo, podemos graficarlo para que el personal que se encuentre cuidando a las personas pueda visualizar el ritmo cardíaco de los distintos pacientes y tome también sus propias conclusiones. El siguiente código muestra este último requerimiento:

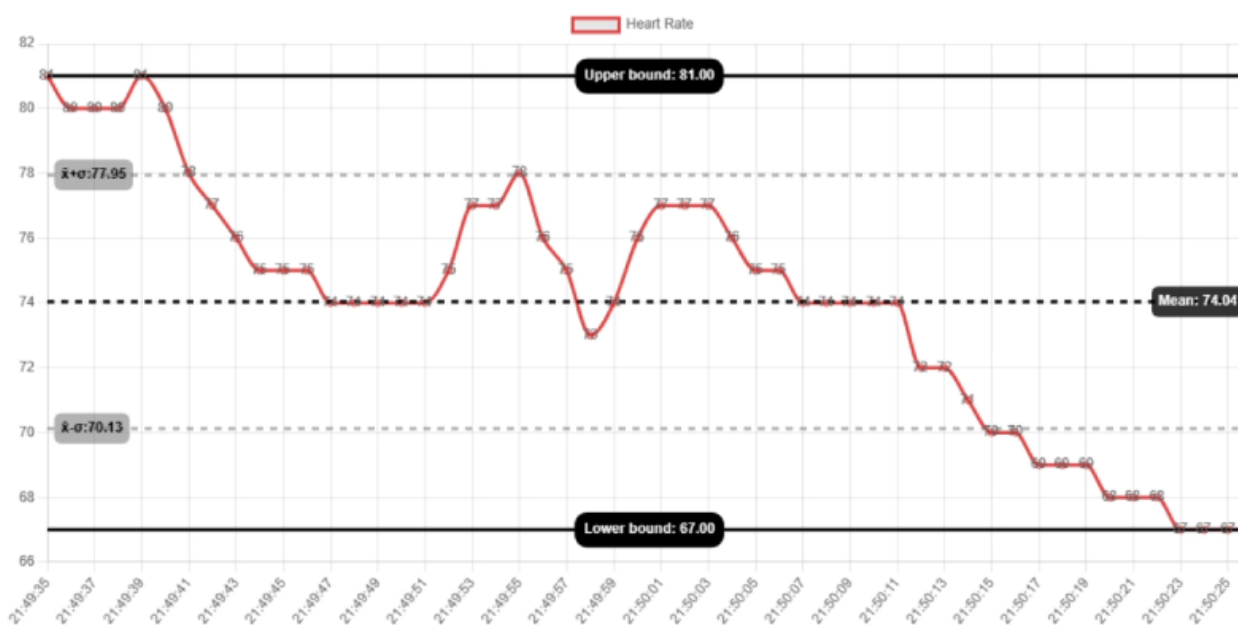
```
connect(" WebSocket", " wss://dev.pulsoid.net/api/v1/data/real_time")
  .pipe(
    plot({
      type: 'line',
      scan: ({
        dataset,
        next
      }) => dataset.data.push(next.data.heart_rate),
      data: {
        datasets: [{
          label: 'Heart Rate',
          cubicInterpolationMode: 'monotone',
          data: [-100],
        }]
      })
    })
  )
  )
  )
```

En la Figura 24 se muestran los resultados de las gráficas obtenidas del EVAnotebook a partir de los datos de ritmo cardíaco obtenidos de dos dispositivos diferentes, el Apple Watch y Polar H10. Cabe mencionar que el dispositivo Apple Watch es menos preciso en comparación al Polar H10, esto debido a que uno está diseñado para monitorear exclusivamente el ritmo cardíaco y el otro no, por lo tanto, la transmisión de los datos es mejor usando el sensor Polar H10 ya que tarda 1 seg, en comparación con el Apple Watch que tarda 5 segundos en enviar cada dato. A continuación se presentan las dos gráficas:



(a)

Heart Rate Using Polar H10



(b)

Figura 24. Resultados de las gráficas obtenidas de la frecuencia cardiaca transmitida de una persona realizando ejercicio, a partir de dos sensores fisiológicos. a) Apple Watch. b) Polar H10

5.1.3. Escenario 3: Recetas de comida con ChatGPT

En este escenario, se muestra la interacción presentada en la sección 2.4.3, donde el usuario le señala al robot EVA una mesa que contiene varios ingredientes para cocinar. EVA al reconocer la seña que le hace el usuario, girará la cabeza en la dirección a la que apunta con su mano, capturará una imagen del plato de comida que está puesto sobre la mesa, y llamará a un algoritmo de clasificación para identificar los ingredientes y posteriormente proporcionarle una variedad de recetas al usuario, ver Figura 25.

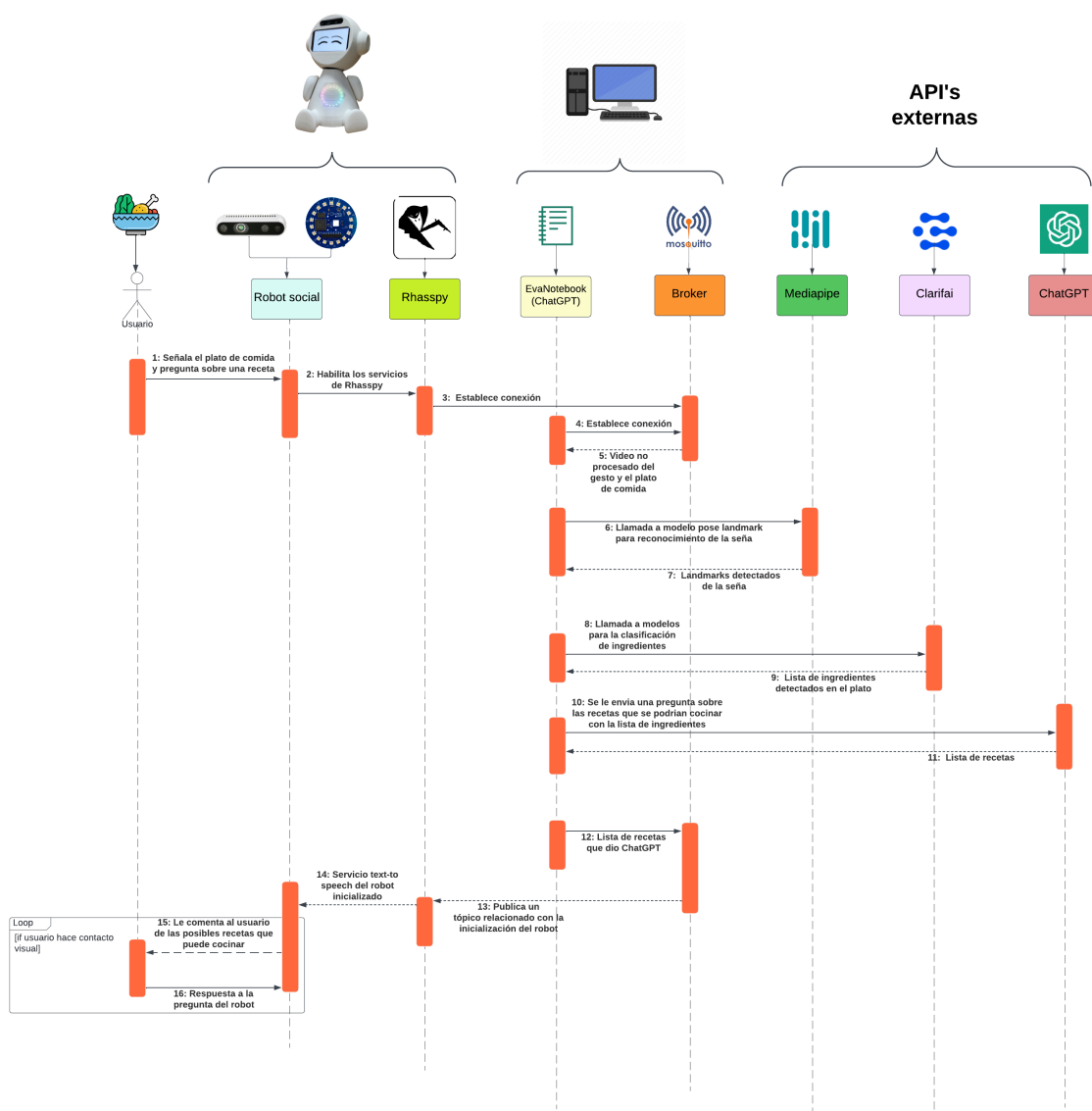


Figura 25. Diagrama de secuencia de un usuario pidiendo a EVA una receta con los ingredientes que tiene a mano.

A continuación se muestra una breve descripción del flujo de interacción entre el robot EVA y la persona:

1. La persona coloca su plato con ingredientes frente a EVA y le indica con un gesto corporal que voltee a ver los ingredientes de la comida, para posteriormente preguntarle "¿qué puedo cocinar con esto?", mientras señala a la comida. El modelo que utiliza EVA para reconocer este gesto corporal es *pose landmark* de Mediapipe.
2. EVA reconoce el gesto que realiza la persona para indicarle que vea el plato de comida y voltea en la dirección que le señala, y usa sus capacidades de visión artificial a través del algoritmo *food-item-recognition* de Clarifai, para identificar los ingredientes presentes en el plato. El algoritmo clasifica y etiqueta los ingredientes.
3. Una vez identificados los ingredientes, se enviará la lista de ingredientes agrupados y clasificados.
4. EVAnotebook transmite un prompt preguntando por recetas candidatas a partir de la lista de ingredientes agrupados a ChatGPT.
5. ChatGPT analiza la lista de ingredientes y genera una respuesta con los nombres de las recetas y sus métodos de preparación.
6. EVA lee la respuesta generada y se la transmite a la persona de forma verbal.

A continuación se muestra el pseudocódigo que representa el funcionamiento del flujo de interacción descrito anteriormente:

```
// Escuchamos la solicitud del usuario al escuchar el topico con MQTT
connect('mqtt', 'hermes/intent/food').pipe(
// Obtiene un frame de la camera para luego procesarlo
// concatMap es un operator que vuelve transparente el asincronismo.
concatMap(_ => getOneFrameImage()),
// Usa un modelo para detectar la comida en el frame
concatMap(frame => clarifai.recognizeFoodIn(frame)),
// Llama al modelo del lenguaje para resolver la tarea entre manos, esto es dado los
ingredientes en la imagen y el contexto del usuario, responde a que podemos cocinar
concatMap(food => openai.reply(food)),
// El robot dice la respuesta
tap(response => mqtt.publish('hermes/tts/say', response))
)
```

En la Figura 26, se puede observar a un usuario realizando un gesto corporal para señalar un plato que contiene varios ingredientes. El objetivo de este gesto es permitir que el robot EVA capture una imagen de los ingredientes utilizando su cámara integrada y, en conjunto con un modelo preentrenado que posee, buscar sugerencias específicas de comidas que se pueden preparar con esos ingredientes. Este proceso busca brindar recomendaciones culinarias basadas en los elementos visibles en la imagen del plato.

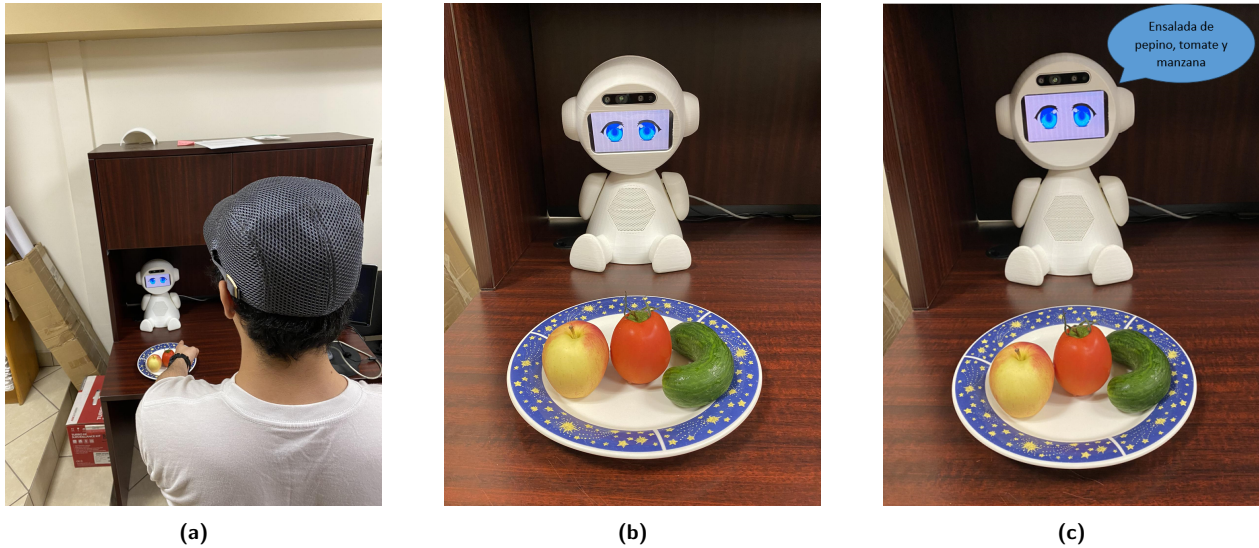


Figura 26. Usuario interactuando con el robot EVA para preguntarle sobre recetas de comida con los ingredientes que se encuentran en una mesa. a) Usuario señalando con la mano el plato con los ingredientes. b) EVA agachando la cabeza para ver los objetos sobre la mesa. c) EVA sugiriendo las recetas que se pueden cocinar con los ingredientes que reconoció.

5.2. Escenarios adicionales

Dentro de las posibilidades que esta arquitectura nos ofrece, algunas de ellas son implementar diversos escenarios de comunicación no verbal que estén acompañados con distintos tipos de comunicación verbal que complementen la interacción.

A continuación, se presentan tres escenarios que fueron desarrollados con el objetivo de experimentar y evaluar la flexibilidad que tiene la arquitectura para integrar comportamientos que no solo comprendan el lenguaje no verbal.

Cabe mencionar que estos escenarios fueron interpretados por estudiantes y personal del departamento de Ciencias de la Computación en un ambiente controlado. Se buscaba analizar cómo la arquitectura podría adaptarse y responder en situaciones que involucraran más que simplemente el lenguaje corporal.

5.2.1. Escenario adicional 1: Degustación del té

Este escenario fue propuesto con el propósito de verificar la flexibilidad, usabilidad y facilidad para que personas que no tengan ningún conocimiento sobre cómo utilizar la arquitectura, puedan integrar nuevos modelos o dispositivos a la arquitectura para detectar algún comportamiento no verbal. En este caso se presenta un juego de adivinanza, en el cual una persona tiene que adivinar qué ingredientes contiene el té que se les proporcionará. El robot EVA, con ayuda de sus sensores y modelos que tiene integrados, estará guiando al usuario en todo el proceso de la interacción.

En esta ocasión EVA será un experto en bebidas, específicamente el té. Para ello se diseñó un experimento en el cual el robot se comportará de diferente manera dependiendo de qué estado de ánimo se le especifique. En una mesa se encontrarán tres diferentes tipos de té. El usuario dirá el número que quiere adivinar, ya que solo adivinará uno. El robot sabiendo ya que té es cada número, debe guiar la conversación con pistas y preguntas hasta que la persona logre adivinar el té, sin mencionar nunca estos ingredientes al dar las pistas, ver Figura 27.

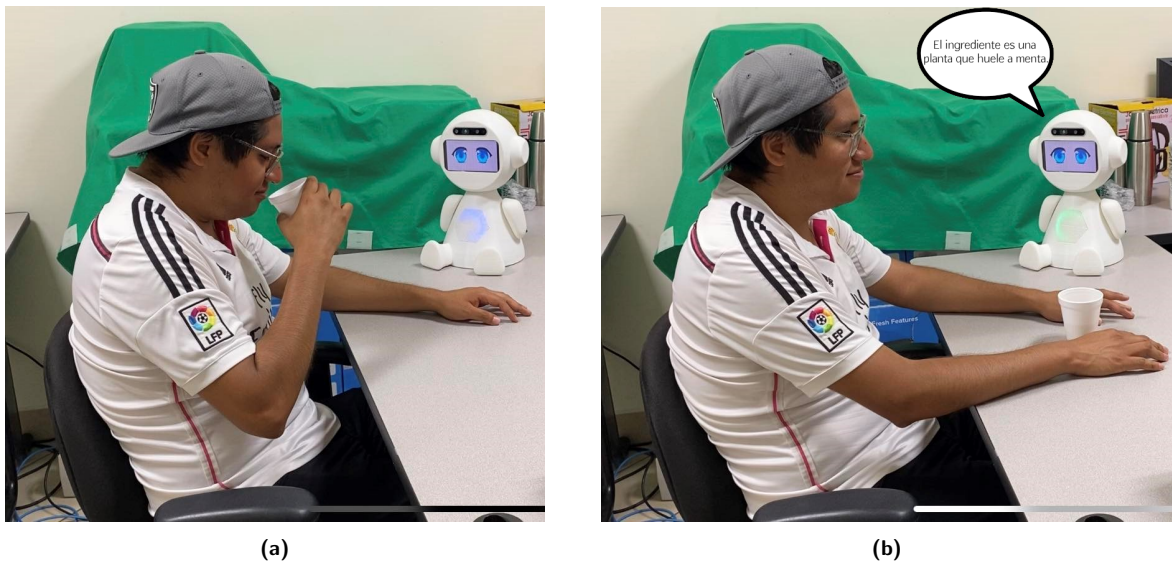


Figura 27. Interacción con el robot EVA de un usuario adivinando los ingredientes de un té utilizando el olfato. a) EVA reconoce que el usuario está oliendo el té. b) EVA indicándole al usuario una pista de algún ingrediente que contiene el té.

La interacción termina cuando el usuario adivina correctamente todos los ingredientes que contiene el té o cuando se le acaban las oportunidades que le dio el robot de adivinar los ingredientes. Cabe mencionar que cualquier persona puede ser capaz de manipular las características que el robot presenta, todo esto desde un EVAnotebook el cual tiene integrado *LangChain* para indicarle en un prompt el contexto que nosotros queramos darle al robot.

5.2.2. Escenario adicional 2: Terapia de reminiscencia

El escenario es una terapia de reminiscencia entre una persona con demencia llamada Antonio y el robot de asistencia social EVA. Antonio vivió la mayor parte de su vida en Cuernavaca y el robot le hace conversación hablando sobre lugares que recuerda de Cuernavaca y alimentos que le gustan que son típicos de esa ciudad. El robot debe guiar la conversación con preguntas sin presionar a Antonio para que recuerde cosas específicas, buscando generar una conversación interesante, agradable y con cierto humor, ver Figura 28.

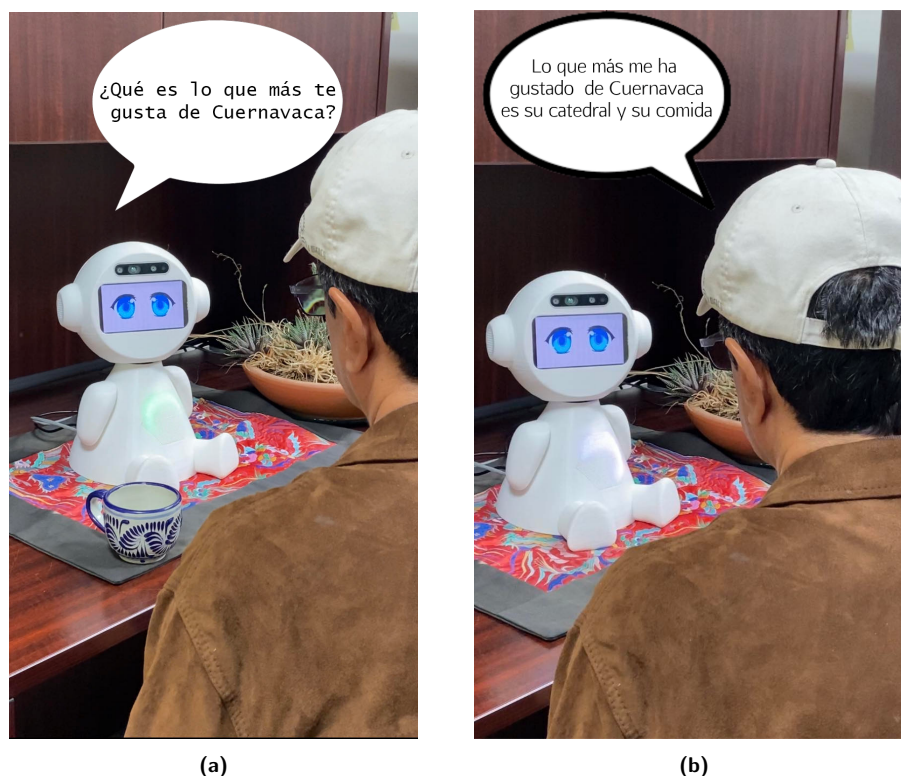


Figura 28. Interacción con el robot EVA de un usuario conversando sobre sus acontecimientos relevantes en Cuernavaca. a) EVA realizando preguntas al usuario sobre su pasado en Cuernavaca. b) Usuario respondiendo ante las preguntas del robot.

En la figura anterior, se puede observar al usuario conversando con el robot EVA sobre acontecimientos que recuerda sobre su pasado, teniendo en cuenta que se tiene la información necesaria para que el robot pueda preguntarle cosas que no llegaran a ocasionar algún problema relacionado con un tema que sea delicado para la persona y la llegue a incomodar. Para que lo anterior se pueda cumplir, es necesario escribir en un *prompt* toda la información que tengamos del usuario, como por ejemplo, sus intereses, gustos personales, fechas importantes, disgustos, etc. Esto para tener un contexto más amplio sobre los acontecimientos importantes de los cuales se podrían mencionar durante la conversación con el robot.

5.2.3. Escenario 3: Lengua de señas mexicana

La interpretación del lenguaje gesto-espacial emerge como un medio significativo para facilitar una comunicación más fluida entre las personas. Además, desde una perspectiva económica, el uso de cámaras se presenta como una opción más accesible y menos costosa en comparación con otros sensores especializados. En este contexto, proponemos un marco de trabajo diseñado específicamente para interpretar gestos, expresiones faciales y otros signos no verbales. Este enfoque puede desempeñar un papel crucial en la comprensión de las necesidades, deseos y emociones de los pacientes, especialmente aquellos con demencia, quienes a menudo enfrentan desafíos significativos para comunicarse mediante el lenguaje convencional.

Por ende, nuestra herramienta propuesta ofrece una alternativa valiosa y efectiva para captar y transmitir sus mensajes de manera más clara y precisa. En particular, proponemos interpretar la lengua de señas mexicana para facilitar la comunicación entre las personas sordas o con problemas auditivos y aquellas que no conocen la lengua de señas mexicana. Esta herramienta no solo eliminará barreras de comunicación, sino que también promoverá la inclusión social de las personas sordas, permitiéndoles participar de manera más activa y efectiva en la vida cotidiana y social. El objetivo principal de este escenario es describir las etapas para la interpretación de lenguaje gesto-espacial y el software asociado ¹, así como presentar como caso de estudio la lengua de señas mexicana. Esto implica el uso aprendizaje de máquina y el procesamiento de imágenes, para mejorar la precisión y velocidad de la interpretación.

La primera etapa del proceso implica la captura de un flujo continuo de imágenes o vídeo en tiempo real. Luego, mediante Mediapipe (aunque también se pueden emplear alternativas como OpenPose o una versión de YOLO), se identificarán los puntos de referencia cruciales en las manos y el cuerpo, elementos esenciales para la interpretación de la lengua de señas. En la etapa subsiguiente, los gestos identificados se clasificarán utilizando un algoritmo basado en el método kNN (k-Nearest Neighbors o el vecino más cercano), que puede ser de fuerza bruta, K-d Tree o Ball Tree. Este algoritmo empleará una distancia métrica de Procrustes y un espacio métrico derivado del análisis de Procrustes. Este último se centra en el espacio de matrices (o formas de objetos) después de eliminar factores de translación, rotación y escala, permitiendo identificar la clase equivalente más cercana.

Elegimos este enfoque debido a que podemos crear un modelo eficiente con una mínima cantidad de imágenes sin entrenamiento; pero es posible clasificar mediante otros métodos, como la Máquinas de

¹https://github.com/sanchezcarlosjr/mexican_sign_language_toolkit/

Soporte Vectorial, redes neuronales (convoluciones, recurrentes o transformadores), o una combinación de estos con Procrustes. Una vez que los gestos sean reconocidos, estos se transformarán en tokens que representarán palabras en un lenguaje verbal. Posteriormente, se realizará un análisis sintáctico y semántico de las expresiones verbales del usuario. Dicho análisis puede abarcar desde la implementación de condicionales simples hasta el empleo de gramáticas complejas, inferencia de tipos o la utilización de modelos avanzados de lenguaje, como GPT4 o BERT.

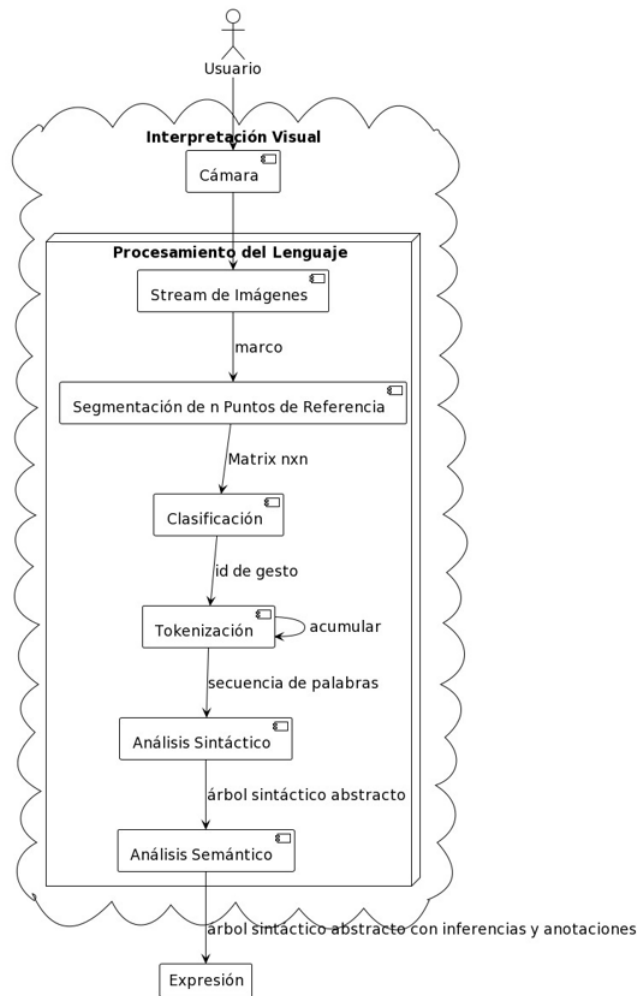


Figura 29. Etapas de lenguajes gesto-espaciales.

Estos modelos pueden aplicarse mediante técnicas como few-shot learning ², predicción de máscaras, fine-tuning o algún otro método más sofisticado. El objetivo final es transformar estos tokens en texto significativo y coherente en el idioma de destino, garantizando así una comunicación efectiva y precisa. El proceso general puede visualizarse en la Figura 29. La expresión final no solo tiene como propósito ser traducida, sino que también cumple con una función operacional, es decir, sirve para ejecutar acciones

²<https://chat.openai.com/share/05ce1523-df42-414a-89c1-7c39107e1dec>

en el sistema.

El enfoque delineado no solo es prometedor para la interpretación de la lengua de señas, sino que también ostenta el potencial de erigirse como un método universal para identificar el lenguaje gesto-espacial rico y diverso que las personas transmiten a través de sus cuerpos. Este lenguaje gesto-espacial no se limita a gestos simples como la afirmación o negación (comunicar «sí» o «no»), sino que se extiende a acciones más complejas y sutiles. Por ejemplo, puede ser útil para interpretar la revisión de una rutina de ejercicios, donde cada movimiento tiene un significado específico. Además, puede aplicarse en el estudio de la proxémica, en la identificación de posturas y gestos que denoten emociones y actitudes particulares, y también en el análisis del baile.

Conforme a (Sainos Vizuet, 2022), es posible categorizar nuestro enfoque, el cual está guiado por Visión por Computadora, como un método dinámico y semántico en un lenguaje objetivo. En este enfoque dinámico, cada marco de lenguaje contiene un gesto o segmento, y un conjunto de n gestos conforma una palabra. Para ilustrar las etapas de nuestro método, presentamos un ejemplo utilizando la lengua de señas mexicana como caso de estudio³.

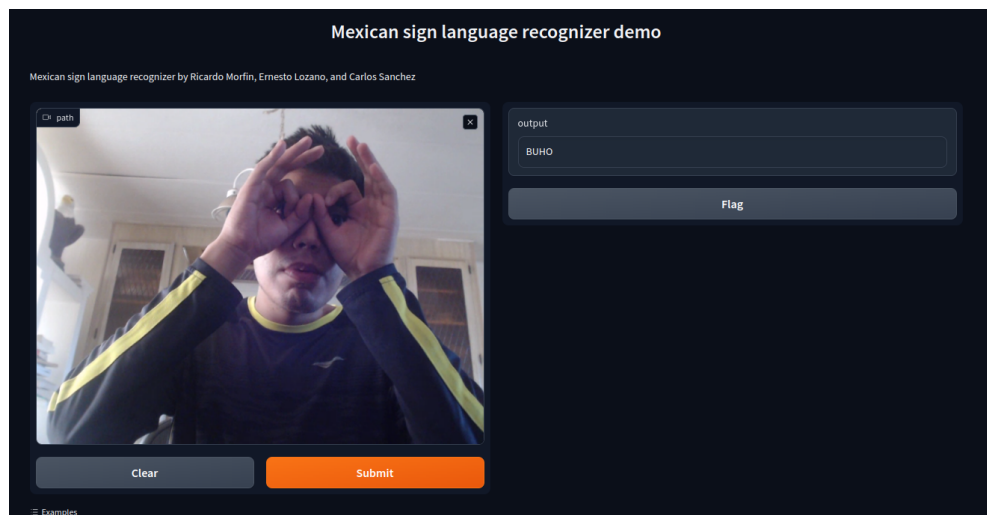


Figura 30. Interpretación de un lenguaje gesto-espacial.

En el ejemplo ilustrado en la Figura 30 se observa como un usuario al momento de realizar una seña, el sistema se encarga de reconocerla y arrojar como salida el nombre de la seña realizada. Este framework aún está en proceso de desarrollo, sin embargo, se podría incorporar en algún robot social para resolver los problemas que se presentan con otros modelos de reconocimiento de comportamientos no verbales.

³<https://msl.sanchezcarlosjr.com/>

Capítulo 6. Conclusiones y trabajo a futuro

En esta sección se presentan las conclusiones de este trabajo de tesis. Se describen las principales contribuciones a partir de la investigación. Además, se discuten las limitaciones que se presentaron en el desarrollo del mismo y para finalizar, el trabajo a futuro que se propone realizar.

6.1. Conclusiones

En este trabajo de tesis se describe el proceso para diseñar, implementar y evaluar una arquitectura la cual ofrece mecanismos para detectar comportamientos no verbales a partir de escenarios de interacción humano-robot, utilizando el robot social EVA, con el objetivo de evaluar la facilidad y flexibilidad con la que se pueden integrar nuevos componentes (sensores, modelos, etc) para solucionar algún problema en específico.

Dos componentes clave de la arquitectura son un servidor intermediario que distribuye mensajes de sensores, actuadores y módulos de análisis, y una red computacional P2P utilizada para desarrollar y probar rápidamente prototipos que pueden convertirse en componentes de escenarios más complejos. Además, se ilustró el uso de la arquitectura con tres escenarios unimodales y multimodales que hacen uso de diversos sensores y modelos preentrenados para reconocer distintos tipos de señales no verbales.

Durante la realización del trabajo, fue necesario tomar en cuenta los requisitos que se debían cubrir antes de diseñar y desarrollar la arquitectura, tomando en cuenta los escenarios establecidos y qué tipo de tecnologías se iban a necesitar al momento de capturar datos multimodales. También, se optó por utilizar un protocolo de comunicación MQTT debido a su gran capacidad de transmitir mensajes a través de dispositivos IoT, ya que la capacidad de cómputo interna del robot imposibilitaba realizar interacciones humano-robot complejas (Raspberry Pi 4, sensores de ritmo cardíaco, etc).

Los escenarios desarrollados durante este trabajo de tesis no fueron evaluados por usuarios finales, esto se debió a varios motivos: el objetivo principal de esta tesis es proporcionar una herramienta a desarrolladores e investigadores para solucionar diversos problemas de reconocimiento de comportamientos no verbales en el campo de IHR utilizando modelos y dispositivos. Por otro lado, la arquitectura desarrollada no está destinada a ser una aplicación o sistema final con los que los usuarios finales interactúen directamente. En cambio, es una herramienta que permitirá a otros desarrollar sistemas más complejos y aplicaciones basadas en ella.

A diferencia de los trabajos reportados en el Estado del Arte se decidió abordar el problema con una estrategia que aprovechara al máximo la relación entre los componentes integrados en el robot EVA y la integración de una nueva arquitectura que pudiera aprovechar esos mismos componentes pero ahora, en lugar de realizar todo el procesamiento dentro de la Raspberry, se pudiera realizar en otra computadora, con el objetivo de reducir algún riesgo de fallo en el sistema por procesar muchos datos y mejorar el tiempo de respuesta.

6.2. Contribuciones

Las principales contribuciones de este trabajo de tesis son las siguientes:

Desarrollo e implementación de una arquitectura flexible la cual tiene el potencial de llevar a cabo todo el procesamiento de los datos recopilados fuera del robot EVA, así como la capacidad de integrar diferentes dispositivos IoT y modelos para el reconocimiento de comportamientos no verbales.

Modificaciones al robot EVA en cuestión de software, para mejorarla en cuanto a capacidad de procesamiento, almacenamiento de datos y capacidad afectiva mediante el uso de diversas herramientas y protocolos de comunicación.

Notebook computacional que permite integrar diversas tecnologías de sensores y modelos para reconocer una amplia gama de comportamientos no verbales. Mediante la incorporación de distintos modelos, como algoritmos de visión por computadora y técnicas de procesamiento del lenguaje natural, EVA puede analizar e interpretar los datos recopilados, obteniendo una comprensión exhaustiva de las acciones e intenciones humanas.

Publicación del artículo *An Open Framework for Nonverbal Communication in Human-Robot Interaction* en la 15va Conferencia Internacional sobre Computación Ubicua e Inteligencia Ambiental (UCAmI2023).

6.3. Limitaciones

A medida que se fue integrando en la arquitectura los modelos y dispositivos utilizados en el transcurso de este trabajo de tesis, se fueron presentando diversos retos, los cuales nos impedían progresar. La

rápida evolución de APIs que pueden ser usadas en la detección de comportamientos no verbales puede ser un desafío, debido a que cada vez existen nuevas técnicas y enfoques que conducen a incluir mejores algoritmos de procesamiento de imágenes, métodos de fusión de datos multimodales o enfoques más efectivos que requieran un mayor nivel de cómputo.

Teniendo en cuenta el hardware que tiene integrado el robot EVA, nos limita la posibilidad de trabajar con algoritmos de aprendizaje profundo, debido que con el simple hecho de requerir una cantidad de recursos tan grandes, la Raspberry Pi 4 no será capaz de soportar esto. Existen modelos de visión que también requieren de un nivel de cómputo a grandes escalas, como una tarjeta gráfica dedicada con cierta VRAM, incluso un procesador de alto nivel que pueda soportar entrenamientos de modelos de muchas horas de trabajo.

Aunque la mayor parte del procesamiento de los datos recopilados por los sensores del robot se realizó externamente, si lo que se desea es procesar una gran cantidad de datos multimodales simultáneamente, pudiera reflejarse en un problema con la respuesta de interacción del robot EVA, debido a que existen diversos factores los cuales puedan interferir durante este proceso, tales como las llamadas a los servicios, modelos e incluso la red de internet.

Conforme a los escenarios desarrollados, cada uno de ellos presentó un reto en particular, siendo el más complicado de todos el escenario 2.4.3. Este escenario fue el que más problemas tuvo, en cuestión de la sincronización de la captura de datos, procesamiento en tiempo real de las imágenes de los ingredientes y la entrega inmediata de la respuesta del modelo de GPT-3.5.

6.4. Trabajo a futuro

Conforme se desarrolló este trabajo surgieron nuevas vertientes que podrían enriquecer nuestra arquitectura, a pesar de que no se desarrollaron quedan como propuesta para trabajo a futuro y se mencionan a continuación.

Se propone replicar la arquitectura implementada en algún otro tipo de robot social, esto con el objetivo de corroborar la capacidad de adaptabilidad y flexibilidad que se desea alcanzar.

Además, se propone continuar integrando nuevas funcionalidades al EVAnotebook para que funcione con otros protocolos de comunicación, y no solamente con los que cuenta por el momento (MQTT,

WebSockets, WebRTC, HTTP). Esto podría habilitar que el robot interactúe con múltiples dispositivos en el ambiente, y no solamente con los que tiene integrado internamente EVA.

Explorar el uso de modelos de aprendizaje profundo los cuales requieran de una recopilación de datos para detectar y clasificar otros tipos de comportamientos no verbales que sean complejos. Sin embargo, para lograr lo anterior se requiere una gran cantidad de datos para entrenar los modelos y que clasifiquen mejor las señales no verbales. Esto se puede obtener ya sea sintetizando datos o adquiriendo nuevos conjuntos de datos con las características estudiadas, incluso con otros dispositivos de adquisición de datos, por ejemplo, dispositivos portátiles, sensores ambientales, redes WiFi o Bluetooth, etc.

Para mejorar el rendimiento del robot y que tenga un mejor desempeño, se requiere mejorar el hardware que tiene integrado, es por ello que se propone realizar las adecuaciones necesarias en el robot EVA para integrar un hardware más avanzado, esto para tener la capacidad necesaria para hacer uso de algoritmos de visión por computadora e incluso de aprendizaje profundo para que tengan un mejor desempeño en el análisis y reconocimiento de comportamientos no verbales complejos.

Colaborar con otras instituciones que hayan trabajado con algún robot social, como es el caso del robot Fred. Este robot tiene la capacidad de realizar movimientos con las piernas, hablar y realizar gestos faciales mediante una pantalla. Un escenario que se podría realizar con el robot EVA y Fred que hagan uso de comportamientos no verbales podría ser un juego de baile con niños. EVA tendría la capacidad de hacer el reconocimiento de los movimientos que realicen durante el juego, mientras que Fred sería el que los vaya guiando en todo el proceso del juego.

Una cuestión muy importante la cual se debe de trabajar es la seguridad y manejo de los datos personales que se utilizaron en este trabajo de tesis, por lo que se deberían de agregar aspectos de seguridad a la plataforma para evitar que estos datos puedan ser explotados por terceros no autorizados.

Literatura citada

- Alaerts, K., Nackaerts, E., Meyns, P., Swinnen, S. P., & Wenderoth, N. (2011). Action and emotion recognition from point light displays: an investigation of gender differences. *PloS one*, 6(6), e20989. <https://doi.org/10.1371/journal.pone.0020989>.
- Andrist, S., Tan, X. Z., Gleicher, M., & Mutlu, B. (2014). Conversational gaze aversion for humanlike robots. 25–32. <https://doi.org/10.1145/2559636.2559666>.
- Astorga, M., Cruz-Sandoval, D., & Favela, J. (2023). A social robot to assist in addressing disruptive eating behaviors by people with dementia. *Robotics*, 12(1), 29. <https://doi.org/10.3390/robotics12010029>.
- Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice: Software Architect Practice_c3*. Addison-Wesley. <https://tinyurl.com/3pxkmnwe>.
- Beck, A., Hiolle, A., Mazel, A., & Cañamero, L. (2010). Interpretation of emotional body language displayed by robots. 37–42. <https://doi.org/10.1145/1877826.1877837>.
- Benamara, N. K., Val-Calvo, M., Álvarez-Sánchez, J. R., Díaz-Morcillo, A., Ferrández Vicente, J. M., Fernández-Jover, E., & Stambouli, T. B. (2019). Real-time emotional recognition for sociable robotics based on deep neural networks ensemble. 171–180. https://doi.org/10.1007/978-3-030-19591-5_18.
- Benamara, N. K., Val-Calvo, M., Alvarez-Sanchez, J. R., Diaz-Morcillo, A., Ferrandez-Vicente, J. M., Fernandez-Jover, E., & Stambouli, T. B. (2021). Real-time facial expression recognition using smoothed deep neural network ensemble. *Integrated Computer-Aided Engineering*, 28(1), 97–111. <https://doi.org/10.3233/ICA-200643>.
- Breazeal, C., Dautenhahn, K., & Kanda, T. (2016). Social robotics. *Springer handbook of robotics*, 1935–1972. https://doi.org/10.1007/978-3-319-32552-1_72.
- Breazeal, C. & Scassellati, B. (1999). How to build robots that make friends and influence people. 2, 858–863. <https://doi.org/10.1109/IR0S.1999.812787>.
- Cruz-Sandoval, D., Morales-Tellez, A., Sandoval, E. B., & Favela, J. (2020). A social robot as therapy facilitator in interventions to deal with dementia-related behavioral symptoms. 161–169. <https://doi.org/10.1145/3319502.3374840>.
- da Rocha, M. M. & Muchaluat-Saade, D. C. (2023). Evaml e evasim: Proposta de linguagem baseada em xml e simulador para o robô eva. 98–107. <https://doi.org/10.5753/ctd.2023.230080>.
- Dumitru, R., Goodfellow, I., Cukierski, W., & Bengio, Y. (2013). Challenges in representation learning: Facial expression recognition challenge. <https://www.kaggle.com/competitions/challenges-in-representation-learning-facial-expression-recognition-challenge>.
- Faria, D. R., Vieira, M., Faria, F. C., & Premevida, C. (2017). Affective facial expressions recognition for human-robot interaction. 805–810. <https://doi.org/10.1109/ROMAN.2017.8172395>.
- Fasola, J. & Mataric, M. J. (2012). Using socially assistive human–robot interaction to motivate physical exercise for older adults. *Proceedings of the IEEE*, 100(8), 2512–2526. <https://doi.org/10.1109/JPROC.2012.2200539>.
- Filipe, L., Peres, R. S., & Tavares, R. M. (2021). Voice-activated smart home controller using machine learning. *IEEE Access*, 9, 66852–66863. <https://doi.org/10.1109/ACCESS.2021.3076750>.

- Fong, T., Nourbakhsh, I., & Dautenhahn, K. (2003a). A survey of socially interactive robots. *Robotics and autonomous systems*, 42(3-4), 143–166. [https://doi.org/10.1016/S0921-8890\(02\)00372-X](https://doi.org/10.1016/S0921-8890(02)00372-X).
- Fong, T., Thorpe, C., & Baur, C. (2003b). Collaboration, dialogue, human-robot interaction. 255–266. https://doi.org/10.1007/3-540-36460-9_17.
- Grishchenko, I., Ablavatski, A., Kartynnik, Y., Raveendran, K., & Grundmann, M. (2020). Attention mesh: High-fidelity face mesh prediction in real-time. *arXiv preprint arXiv:2006.10962*. <https://doi.org/10.48550/arXiv.2006.10962>.
- Hall, E. T. (1973). *The silent language*. Anchor. https://monoskop.org/images/5/57/Hall_Edward_T_The_Silent_Language.pdf.
- Hall, J. A., Horgan, T. G., & Murphy, N. A. (2019). Nonverbal communication. *Annual review of psychology*, 70, 271–294. <https://doi.org/10.1146/annurev-psych-010418-103145>.
- Hans, A. & Hans, E. (2015). Kinesics, haptics and proxemics: Aspects of non-verbal communication. *IOSR Journal of Humanities and Social Science (IOSR-JHSS)*, 20(2), 47–52. <https://www.iosrjournals.org/iosr-jhss/papers/Vol20-issue2/Version-4/H020244752.pdf>.
- Hegel, F., Muhl, C., Wrede, B., Hielscher-Fastabend, M., & Sagerer, G. (2009). Understanding social robots. 169–174. https://doi.org/10.1007/978-3-030-34964-6_8.
- Hindemith, L., Vollmer, A.-L., Wiebel-Herboth, C. B., & Wrede, B. (2021). Improving hri through robot architecture transparency. *arXiv preprint arXiv:2108.11608*. <https://doi.org/10.48550/arXiv.2108.11608>.
- Hömke, P., Holler, J., & Levinson, S. C. (2017). Eye blinking as addressee feedback in face-to-face conversation. *Research on Language and Social Interaction*, 50(1), 54–70. <https://doi.org/10.1080/08351813.2017.1262143>.
- Imai, M., Ono, T., & Ishiguro, H. (2003). Physical relation and expression: Joint attention for human-robot interaction. *IEEE Transactions on Industrial Electronics*, 50(4), 636–643. <https://doi.org/10.1109/TIE.2003.814769>.
- Jang, M. H. (2006). *Linux Annoyances for Geeks: Includes Index*. O'Reilly. <https://books.google.mu/books?id=U6abAgAAQBAJ&printsec=frontcover#v=onepage&q&f=false>.
- Juan, E., Frum, C., Bianchi-Demicheli, F., Yi-wen, W., Lewis, J., & Cacioppo, S. (2013). Beyond human intentions and emotions. *Frontiers in Human Neuroscience*, 7. <https://doi.org/10.3389/fnhum.2013.00099>.
- Kim, M., Kumar, S., Pavlovic, V., & Rowley, H. (2008). Face tracking and recognition with visual constraints in real-world videos. 1–8. <https://doi.org/10.1109/CVPR.2008.4587572>.
- Kleppmann, M., Wiggins, A., Van Hardenberg, P., & McGranaghan, M. (2019). Local-first software: you own your data, in spite of the cloud. 154–178. <https://doi.org/10.1145/3359591.3359737>.
- Kozima, H., Michalowski, M. P., & Nakagawa, C. (2009). Keepon: A playful robot for research, therapy, and entertainment. *International Journal of social robotics*, 1, 3–18. <https://doi.org/10.1007/s12369-008-0009-8>.
- Królak, A. & Strumillo, P. (2012). Eye-blink detection system for human–computer interaction. *Universal Access in the Information Society*, 11, 409–419. <https://doi.org/10.1007/s10209-011-0256-6>.

- Kumar, A. (2023). What is nonverbal communication? principles, functions, types. <https://getup1earn.com/blog/nonverbal-communication/>.
- Kusumota, V. L. P., Aroca, R. V., & Martins, F. N. (2018). An open source framework for educational applications using cozmo mobile robot. 569–576. <https://doi.org/10.1109/LARS/SBR/WRE.2018.00104>.
- Laghrissi, F., Douzi, S., Douzi, K., & Hssina, B. (2021). Intrusion detection systems using long short-term memory (lstm). *Journal of Big Data*, 8(1), 65. <https://doi.org/10.1186/s40537-021-00448-4>.
- Lane, G. W., Noronha, D., Rivera, A., Craig, K., Yee, C., Mills, B., & Villanueva, E. (2016). Effectiveness of a social robot, “paro,” in a va long-term care setting. *Psychological services*, 13(3), 292. <https://doi.org/10.1037/ser0000080>.
- Leite, I., Martinho, C., & Paiva, A. (2013). Social robots for long-term interaction: a survey. *International Journal of Social Robotics*, 5, 291–308. <https://doi.org/10.1007/s12369-013-0178-y>.
- Li, H., John-John, C., & Tan, Y. K. (2011). Towards an effective design of social robots. *International Journal of Social Robotics*, 3(4), 333–335. <https://doi.org/10.1007/s12369-011-0121-z>.
- Li, X. et al. (2021). Design and implementation of human motion recognition information processing system based on lstm recurrent neural network algorithm. *Computational Intelligence and Neuroscience*, 2021. <https://doi.org/10.1155/2021/3669204>.
- Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., et al. (2019). Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*. <https://doi.org/10.48550/arXiv.1906.08172>.
- Maglie, A. & Maglie, A. (2016). Reactivex and rxjava. *Reactive Java Programming*, 1–9. https://doi.org/10.1007/978-1-4842-1428-2_1.
- McColl, D., Hong, A., Hatakeyama, N., Nejat, G., & Benhabib, B. (2016). A survey of autonomous human affect detection methods for social robots engaged in natural hri. *Journal of Intelligent & Robotic Systems*, 82, 101–133. <https://doi.org/10.1007/s10846-015-0259-2>.
- Mead, R., Atrash, A., & Mataric, M. J. (2011). Recognition of spatial dynamics for predicting social interaction. 201–202. <https://doi.org/10.1145/1957656.1957731>.
- Mead, R., Atrash, A., & Matarić, M. J. (2013). Automated proxemic feature extraction and behavior recognition: Applications in human-robot interaction. *International Journal of Social Robotics*, 5, 367–378. <https://doi.org/10.1007/s12369-013-0189-8>.
- Mead, R. & Mataric, M. J. (2014). Probabilistic models of proxemics for spatially situated communication in hri. https://robotics.usc.edu/publications/media/uploads/pubs/2014MeadMataric_HRI2014.pdf.
- Mitra, S. & Acharya, T. (2007). Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(3), 311–324. <https://doi.org/10.1109/TSMCC.2007.893280>.
- Mutlu, B. & Forlizzi, J. (2008). Robots in organizations: the role of workflow, social, and environmental factors in human-robot interaction. 287–294. <https://doi.org/10.1145/1349822.1349860>.
- Mutlu, B., Shiwa, T., Kanda, T., Ishiguro, H., & Hagita, N. (2009). Footing in human-robot conversations: how robots might shape participant roles using gaze cues. 61–68. <https://doi.org/10.1145/1514095.1514109>.

- Picard, R. W. (2000). *Affective computing*. MIT press. <https://affect.media.mit.edu/pdfs/95.picard.pdf>.
- Pineda, L. A., Rodríguez, A., Fuentes, G., Rascon, C., & Meza, I. V. (2015). Concept and functional structure of a service robot. *International Journal of Advanced Robotic Systems*, 12(2), 6. <https://doi.org/10.5772/60026>.
- Pritchett, D. (2008). Base: An acid alternative: In partitioned databases, trading some consistency for availability can lead to dramatic improvements in scalability. *Queue*, 6(3), 48–55. <https://doi.org/10.1145/1394127.1394128>.
- Pulli, K., Baksheev, A., Korniyakov, K., & Eruhimov, V. (2012). Real-time computer vision with opencv. *Communications of the ACM*, 55(6), 61–69. <https://doi.org/10.1145/2184319.2184337>.
- Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. 28492–28518. <https://proceedings.mlr.press/v202/radford23a.html>.
- Rane, P., Mhatre, V., & Kurup, L. (2014). Study of a home robot: Jibo. *International journal of engineering research and technology*, 3(10), 490–493. <https://www.ijert.org/research/study-of-a-home-robot-jibo-IJERTV3IS100361.pdf>.
- Rich, C., Ponsler, B., Holroyd, A., & Sidner, C. L. (2010). Recognizing engagement in human-robot interaction. 375–382. <https://doi.org/10.1109/HRI.2010.5453163>.
- Robaczewski, A., Bouchard, J., Bouchard, K., & Gaboury, S. (2021). Socially assistive robots: The specific case of the nao. *International Journal of Social Robotics*, 13, 795–831. <https://doi.org/10.1007/s12369-020-00664-7>.
- Robinson, H., MacDonald, B., & Broadbent, E. (2014). The role of healthcare robots for older people at home: A review. *International Journal of Social Robotics*, 6, 575–591. <https://doi.org/10.1007/s12369-014-0242-2>.
- Rocha, M., Valentim, P., Barreto, F., Mitjans, A., Cruz-Sandoval, D., Favela, J., & Muchaluat-Saade, D. (2021). Towards enhancing the multimodal interaction of a social robot to assist children with autism in emotion regulation. 398–415. https://doi.org/10.1007/978-3-030-99194-4_25.
- Rocha, M. M. D. & Muchaluat-Saade, D. C. (2023). Friendly robot for education and healthcare: Fred. 19–22. <https://doi.org/10.1145/3604321.3604342>.
- Ross, A. (2004). Procrustes analysis. *Course report, Department of Computer Science and Engineering, University of South Carolina*, 26, 1–8. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ca34cf18bc8d243213adf6b39293f4b31684b344>.
- Rossi, S., Staffa, M., Bove, L., Capasso, R., & Ercolano, G. (2017). User's personality and activity influence on hri comfortable distances. 167–177. https://doi.org/10.1007/978-3-319-70022-9_17.
- Sainos Vizuet, M. (2022). Traducción de lenguaje de señas mexicano a texto mediante aprendizaje profundo. [Tesis de Maestría en Ciencias, Centro de Investigación Científica y de Educación Superior de Ensenada, B.C.].
- Sanghvi, J., Castellano, G., Leite, I., Pereira, A., McOwan, P. W., & Paiva, A. (2011). Automatic analysis of affective postures and body motion to detect engagement with a game companion. 305–312. <https://doi.org/10.1145/1957656.1957781>.

- Saunderson, S. & Nejat, G. (2019). How robots influence humans: A survey of nonverbal communication in social human–robot interaction. *International Journal of Social Robotics*, 11, 575–608. <https://doi.org/10.1007/s12369-019-00523-0>.
- Shamsuddin, S., Ismail, L. I., Yussof, H., Zahari, N. I., Bahari, S., Hashim, H., & Jaffar, A. (2011). Humanoid robot nao: Review of control and motion exploration. 511–516. <https://doi.org/10.1109/ICCSC.2011.6190579>.
- Silva, M. P., do Nascimento Silva, V., & Chaimowicz, L. (2015). Dynamic difficulty adjustment through an adaptive ai. 173–182. <https://doi.org/10.1109/SBGames.2015.16>.
- Singh, A. K., Kumbhare, V. A., & Arthi, K. (2021). Real-time human pose detection and recognition using mediapipe. 145–154. https://doi.org/10.1007/978-981-16-7088-6_12.
- Soomro, Z. A., Shamsudin, A. U. B., Rahim, R. A., Adrianshah, A., & Hazeli, M. (2023). Non-verbal human-robot interaction using neural network for the application of service robot. *IJUM Engineering Journal*, 24(1), 301–318. <https://doi.org/10.31436/ijumej.v24i1.2577>.
- Taichi, T., Takahiro, M., Hiroshi, I., & Norihiro, H. (2006). Automatic categorization of haptic interactions-what are the typical haptic interactions between a human and a robot? 490–496. <https://doi.org/10.1109/ICHR.2006.321318>.
- Takahashi, Y., Hasegawa, N., Takahashi, K., & Hatakeyama, T. (2001). Human interface using pc display with head pointing device for eating assist robot and emotional evaluation by gsr sensor. 4, 3674–3679. <https://doi.org/10.1109/ROBOT.2001.933189>.
- Tapus, A. & Mataric, M. J. (2008). Socially assistive robots: The link between personality, empathy, physiological signals, and task performance. 133–140. <https://cdn.aaai.org/Symposia/Spring/2008/SS-08-04/SS08-04-021.pdf>.
- Terrence, F. (2003). A survey of socially interactive robots. *Robotics and autonomous systems*, 42, 3. [https://doi.org/10.1016/S0921-8890\(02\)00372-X](https://doi.org/10.1016/S0921-8890(02)00372-X).
- Thaman, B., Cao, T., & Caporusso, N. (2022). Face mask detection using mediapipe facemesh. 378–382. <https://doi.org/10.23919/MIPRO55190.2022.9803531>.
- Topsakal, O. & Akinci, T. C. (2023). Creating large language model applications utilizing langchain: A primer on developing llm apps fast. 1(1), 1050–1056. <https://doi.org/10.59287/icaens.1127>.
- Urakami, J. & Seaborn, K. (2023). Nonverbal cues in human–robot interaction: A communication studies perspective. *ACM Transactions on Human-Robot Interaction*, 12(2), 1–21. <https://doi.org/10.1145/3570169>.
- Val-Calvo, M., Álvarez-Sánchez, J. R., Ferrández-Vicente, J. M., & Fernández, E. (2020). Affective robot story-telling human-robot interaction: exploratory real-time emotion estimation analysis using facial expressions and physiological signals. *IEEE Access*, 8, 134051–134066. <https://doi.org/10.1109/ACCESS.2020.3007109>.
- Vigil, M. A., Barhanpurkar, M. M., Anand, N. R., Soni, Y., & Anand, A. (2019). Eye spy face detection and identification using yolo. 105–110. <https://doi.org/10.1109/ICSSIT46314.2019.8987830>.
- Villa, L., Hervás, R., Cruz-Sandoval, D., & Favela, J. (2022a). Design and evaluation of proactive behavior in conversational assistants: Approach with the eva companion robot. 234–245. https://doi.org/10.1007/978-3-031-21333-5_23.

- Villa, L., Hervás, R., Dobrescu, C. C., Cruz-Sandoval, D., & Favela, J. (2022b). Incorporating affective proactive behavior to a social companion robot for community dwelling older adults. 568–575. https://doi.org/10.1007/978-3-031-19682-9_72.
- Walters, M. L., Dautenhahn, K., Te Boekhorst, R., Koay, K. L., Kaouri, C., Woods, S., Nehaniv, C., Lee, D., & Werry, I. (2005). The influence of subjects' personality traits on personal spatial zones in a human-robot interaction experiment. 347–352. <https://doi.org/10.1109/ROMAN.2005.1513803>.
- Wang, J., Lewis, M., Hughes, S., Koes, M., & Carpin, S. (2005). Validating usarsim for use in hri research. 49(3), 457–461. <https://doi.org/10.1177/154193120504900351>.
- Wharton, T. (2009). *Pragmatics and non-verbal communication*. Cambridge University Press. <https://doi.org/10.1017/CB09780511635649>.
- Wilcox, R., Nikolaidis, S., & Shah, J. (2013). Optimization of temporal dynamics for adaptive human-robot interaction in assembly manufacturing. *Robotics*, 8(441), 10–15. <http://hdl.handle.net/1721.1/116089>.
- Wistort, R. & Breazeal, C. (2009). Tofu: a socially expressive robot character for child interaction. 292–293. <https://doi.org/10.1145/1551788.1551862>.
- Xia, F., Wang, H., Fu, X., & Zhao, J. (2005). An xml-based implementation of multimodal affective annotation. 535–541. https://doi.org/10.1007/11573548_69.
- Yang, H.-D., Park, A.-Y., & Lee, S.-W. (2007). Gesture spotting and recognition for human-robot interaction. *IEEE Transactions on robotics*, 23(2), 256–270. <https://doi.org/10.1109/TR0.2006.889491>.
- Yassein, M. B., Shatnawi, M. Q., Aljwarneh, S., & Al-Hatmi, R. (2017). Internet of things: Survey and open issues of mqtt protocol. 1–6. <https://doi.org/10.1109/ICEMIS.2017.8273112>.
- Zhou, J., Feng, L., Chellali, R., & Zhu, H. (2018). Detecting and tracking objects in hri: Yolo networks for the nao “i see you” function. 479–482. <https://doi.org/10.1109/ROMAN.2018.8525582>.
- Zhu, Q. & Luo, J. (2022). Generative pre-trained transformer for design concept generation: an exploration. *Proceedings of the design society*, 2, 1825–1834. <https://doi.org/10.1017/pds.2022.185>.

Anexos

Apéndice A. Características de EVAnotebook

En el siguiente apéndice se presentan en detalle las características fundamentales que tiene EVAnotebook:

Configuración

De acuerdo con la filosofía descrita anteriormente, hemos implementado mecanismos para recibir y limpiar datos de múltiples fuentes distribuidas en tiempo real localmente con programas escritos en JavaScript, Python (específicamente, Pyodide), SWI-Prolog, SQL y otros en cooperación. Para lograr esto, utilizamos *web workers* transparentes que administran de manera efectiva el Sistema de archivos privados de origen (OPFS) y nuestra versión descentralizada de IndexDB. Estos mecanismos están integrados con las API de MQTT, WebRTC, WebSocket y HTTP, inspirándose en los conceptos presentados por ReactiveX y su implementación RxJS, donde estos conceptos nos hablan sobre el paradigma de programación y un conjunto de bibliotecas que permite a los desarrolladores trabajar con programación asíncrona e impulsada por eventos de una manera mas compositiva y declarativa (Maglie & Maglie, 2016).

Gestionar código

Por un lado, el entorno del navegador ofrece una gran cantidad de herramientas para desarrolladores, especialmente en los navegadores basados en chrome, que mejoran enormemente la experiencia de ingeniería de software. Estas herramientas incluyen poderosas capacidades de depuración, herramientas de análisis de red y la capacidad de ampliar la funcionalidad del navegador a través de extensiones.

Por otro lado, trabajar dentro del entorno del navegador también presenta ciertos desafíos que deben abordarse. Las restricciones de seguridad y los errores del navegador pueden representar obstáculos para un desarrollo fluido. Por ejemplo, la restricción de uso compartido de recursos entre orígenes (CORS) puede limitar la capacidad de realizar solicitudes a recursos de diferentes orígenes.

Confiabilidad

Si la computadora tiene la capacidad de manejar grandes conjuntos de datos, ciertamente puede cargarlos. Sin embargo, es importante tener en cuenta que nuestro enfoque principal no es el análisis por lotes, sino el análisis en línea. De hecho, es posible que el entorno del navegador no siempre sea el más adecuado para entrenar modelos de aprendizaje automático. Nos gustaría advertirle a los usuarios que los navegadores son propensos a fallas inesperadas cuando se alcanzan ciertos límites.

Seguridad

Aunque EVAnotebook funciona en un navegador, funciona principalmente como software local sin conexión de forma predeterminada, como se detalla en (Kleppmann et al., 2019). Los usuarios mantienen la propiedad total de sus datos y no almacenamos nada en la nube.

Compartir y colaborar

EVAnotebook admite la publicación de resultados en forma de demostraciones o informes, permite la edición colaborativa y proporciona funcionalidad de memoria compartida en tiempo real a través de esquemas de replicación P2P y HTTP. La resolución de conflictos se logra utilizando CRDT y Lamport Clocks.

Además, elegimos RxDB como nuestro sistema de administración de base de datos, utilizando IndexDB como almacenamiento subyacente, que es una base de datos reactiva diseñada específicamente para funcionar en navegadores y se inspira en el enfoque de replicación multimaestro de CouchDB. Nuestro modelo de replicación sigue un enfoque básicamente disponible, de estado blando y eventualmente consistente, como se describe en (Pritchett, 2008), que contrasta con el modelo ACID.

Reproducir y reutilizar

Como desarrolladores, científicos de datos e investigadores, entendemos la frustración causada por *Dependency Hell*, que se refiere a la frustración de algunos usuarios de programación que han instalado paquetes de software que dependen de versiones específicas de otros paquetes de software (Jang, 2006). Para solucionar este problema, EVAnotebook mantiene una lista estable de bibliotecas que están disponibles para su uso. Sin embargo, es importante tener en cuenta que no podemos garantizar que todas las bibliotecas de npm y pip funcionen en un entorno de navegador.

Notebooks como productos

Hoy en día, la mayoría de los usuarios finales esperan que el software funcione en entornos listos para usar, como navegadores web o aplicaciones para teléfonos inteligentes. La implementación en un navegador generalmente es más fácil en comparación con el ecosistema de teléfonos inteligentes, por lo que es la opción preferible. Nuestro notebook admite un modo de publicación en el que puede implementar sin código visible y tener acceso de solo lectura. Además, si está interesado en ejecutar un servicio en segundo plano, un navegador puede funcionar en modo autónomo, lo que elimina la necesidad de una interfaz gráfica de usuario para ejecutar un notebook.

Apéndice B. Modelos preentrenados

En el apéndice a continuación se explica sobre los modelos preentrenados de MediaPipe que se implementaron y evaluaron en la arquitectura en la subsección 4.6.1:

Modelo face mesh

Es una tecnología que permite realizar el seguimiento en tiempo real de los puntos clave en la cara humana a través de una cámara o una imagen de vídeo. Se puede utilizar para identificar expresiones faciales humanas, aplicar filtros y efectos faciales y crear avatares virtuales (Thaman et al., 2022). Utiliza modelos de aprendizaje automático (ML) que pueden funcionar con imágenes individuales o con un flujo continuo de imágenes. Además, produce puntos de referencia de cara tridimensionales, puntuaciones de blendshape (coeficientes que representan la expresión facial) para inferir superficies faciales detalladas en tiempo real y matrices de transformación para realizar las transformaciones necesarias para la representación de efectos.

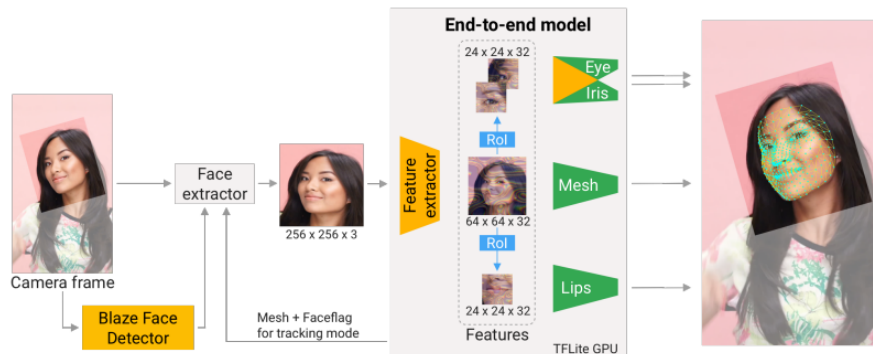


Figura 31. Modelo Face Mesh para la estimación de la orientación del rostro a partir de la extracción de puntos faciales. Tomada de (Grishchenko et al., 2020).

Modelo pose landmark

Permite detectar puntos de referencia de cuerpos humanos en una imagen o vídeo. Se puede utilizar para identificar ubicaciones clave del cuerpo, analizar la postura y categorizar los movimientos (Singh et al., 2021). Esta tarea utiliza modelos de aprendizaje automático (ML) que funcionan con imágenes individuales o vídeo. Los resultados plantean puntos de referencia en coordenadas de imagen y en coordenadas de mundo tridimensional. Este modelo está diseñado para la estimación de pose en tiempo real, lo que implica detectar y rastrear las posiciones de las articulaciones del cuerpo clave desde imágenes o secuencias de vídeo. El modelo proporciona un conjunto de puntos de referencia que corresponden a diferentes partes del cuerpo, lo que permite un seguimiento preciso de las poses y movimientos humanos.

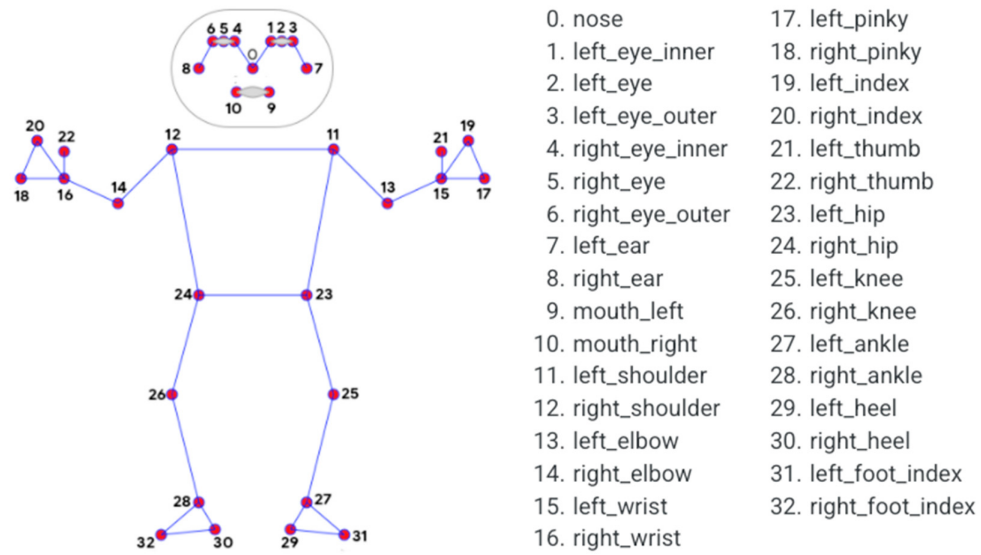


Figura 32. Modelo Pose Landmark para la detección de posturas corporales a partir de la extracción de puntos corporales. Tomada de (Singh et al., 2021).