

## Herramienta Web para la gestión de usuarios en HRI

Quemada-Torres, Eulogio.\*, Moreno, Francisco-Angel, Galindo, Cipriano, Gonzalez-Jimenez, Javier

Grupo de Percepción Artificial y Robótica Inteligente (MAPIR), Dept. de Ingeniería de Sistemas y Automática, Instituto Universitario en Ingeniería Mecatrónica y Sistemas Ciberfísicos (IMECH.UMA), Universidad de Málaga, Blvr. Louis Pasteur, 35, 29071 Málaga, España.

### Resumen

Los robots que operan en entornos con personas están dotados de sistemas de interacción humano-robot (HRI) basados, principalmente, en el reconocimiento facial. Para facilitar el despliegue y mantenimiento en escenarios reales de estos robots es muy conveniente contar con un sistema que, de manera remota, monitorice, controle y actualice la información que el robot almacena sobre las personas. Este trabajo presenta una herramienta web, compatible con ROS2, para este fin. Además de la gestión y edición de identidades, y de los parámetros de configuración de la detección y reconocimiento facial, la solución provee utilidades para generar automáticamente sesiones de interacción, generando estadísticas accesibles en tiempo real. El artículo describe la arquitectura del sistema y demuestra su utilidad en diversos escenarios con el robot social Sancho, donde se validan la inscripción remota de visitantes y la monitorización continua de la actividad del robot.

*Palabras clave:* Robótica social, Gestión de datos, Interacción humano-robot, Interfaces web, ROS 2

### Web Tool for User Management in HRI

#### Abstract

Robots operating in human environments are equipped with Human-Robot Interaction (HRI) systems, primarily based on facial recognition. To facilitate the deployment and maintenance of these robots in real-world scenarios, a system that remotely monitors, controls, and updates the information the robot stores about people is highly desirable. This work presents a ROS 2-compatible web tool for this purpose. In addition to identity management and editing, and configuration of facial detection and recognition parameters, the solution provides utilities to automatically generate interaction sessions, producing real-time accessible statistics. The article describes the system architecture and demonstrates its utility in various scenarios with the social robot Sancho, where remote visitor enrollment and continuous monitoring of the robot's activity are validated.

*Keywords:* Social robotics, Data management, Human-robot interaction, Web interfaces, ROS 2

## 1. Introducción

La creciente necesidad de robots asistenciales, de servicio o educativos que operan en entornos humanos complejos y dinámicos ha impulsado, en los últimos años, el desarrollo de sistemas de interacción humano-robot (HRI, por sus siglas en inglés). Un aspecto importante de los sistemas HRI es la capacidad de personalizar la interacción con el usuario, entendiendo el contexto y ajustando su comportamiento a las características del mismo. A medida que estos sistemas se implantan en entornos reales, surge la necesidad de herramientas que permitan monitorizar, supervisar y gestionar de forma efectiva el conocimiento que el sistema HRI adquiere sobre las personas con las

que interactúa.

En particular, los sistemas de reconocimiento facial utilizados en HRI requieren mecanismos transparentes y accesibles que permitan a los operadores humanos verificar qué información ha sido almacenada (por ejemplo, *embeddings* faciales, imágenes capturadas o identificadores personales), así como editar o ampliar dicha información cuando sea necesario. Esta capacidad resulta especialmente relevante en contextos colaborativos o demostrativos, donde se espera que un robot reconozca a personas específicas en momentos concretos, incluso antes de haberlas visto físicamente. Para abordar estas necesidades, podemos encontrar iniciativas como ROS4HRI (*Robot Opera-*

\*Autor para correspondencia: eulogioquemada@uma.es  
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

ting System for Human-Robot Interaction) (Mohamed and Lemaignan, 2021), un marco que agrupa paquetes, convenciones y herramientas diseñadas para facilitar el desarrollo de robots interactivos utilizando ROS. ROS4HRI establece estándares para la representación y procesamiento de señales sociales, promoviendo la interoperabilidad entre componentes de percepción y acción en HRI. Dentro de este marco, existen herramientas como `hri_face_detect` (Lemaignan and Juricic, 2023), un nodo compatible con ROS4HRI que permite la detección y seguimiento de rostros, extracción de puntos clave y estimación de la pose 3D de la cabeza, utilizando tecnologías como YuNet (Wu et al., 2023) y Mediapipe Face Mesh (Lugaresi et al., 2019). Sin embargo, estas soluciones se centran principalmente en la detección y seguimiento en tiempo real, sin proporcionar interfaces para la gestión y supervisión de las bases de datos de reconocimiento facial.

En este artículo presentamos RUMI (*ROS User Management Interface*), una herramienta web desarrollada específicamente para facilitar la supervisión y gestión remota de sistemas HRI basados en reconocimiento facial. Esta interfaz permite visualizar y editar la base de datos interna del sistema (incluyendo imágenes de rostros, vectores de *embedding* o descriptores, número de interacciones y metadatos asociados), así como añadir nuevas identidades de forma remota mediante la carga de imágenes etiquetadas. Gracias a esta utilidad, un operador puede preparar con antelación al sistema HRI para el reconocimiento de visitantes o participantes, mejorando así la fluidez y naturalidad de las interacciones. RUMI está diseñada para integrarse en la arquitectura ROS 2 y proporciona una solución robusta para la gestión de identidades en sistemas HRI. Hemos puesto esta herramienta a disposición de la comunidad robótica en <https://github.com/eulogioqt/rumi>.

En las siguientes secciones se describe la arquitectura del sistema, los principales componentes implementados y los escenarios de uso que ilustran su aplicabilidad. Asimismo, se discuten las ventajas que ofrece en comparación con otros enfoques.

## 2. Descripción del sistema y funcionalidades

RUMI proporciona una interfaz web para la gestión y supervisión de los usuarios que el robot es capaz de identificar y con los que puede interactuar. Nuestro sistema define lo que denominamos "sesiones de interacción", períodos continuos de tiempo en los que el usuario está interactuado con el robot y en los que el sistema almacena y monitoriza los siguientes parámetros:

- Identificador facial. Un ID único que identifica a una identidad concreta.
- Lista de detecciones, que incluye:
  - Marca de tiempo. Establece la ubicación temporal de la detección.
  - Puntuación de detección facial. Proporcionado por el detector de rostros y relacionado con la confianza del método en que una región en la imagen contiene un rostro.

- Puntuación de reconocimiento facial. Valor normalizado que expresa la certeza del clasificador de que el rostro pertenece a una identidad concreta.

A continuación describimos las funcionalidades principales que proporciona RUMI, así como el conjunto de elementos que conforman su funcionamiento.

### 2.1. Funcionalidades

- **Gestión automática de sesiones.** El sistema implementa un mecanismo automático para la gestión de sesiones de interacción con cada persona detectada y clasificada. Una sesión se abre cuando una nueva persona entra en el campo de visión del sistema, iniciando el registro continuo de la información asociada a dicha interacción, descrita anteriormente: instante de detección, puntuación de detección facial y la puntuación de reconocimiento facial. A lo largo de la sesión, estos datos se almacenan de forma periódica, lo que permite un seguimiento detallado y configurable de la actividad de cada individuo. La sesión se cierra automáticamente cuando la persona ha permanecido ausente más allá de un determinado umbral de tiempo. Tanto el tiempo de espera para cerrar una sesión como la frecuencia con la que se registran nuevas detecciones pueden ajustarse *on-line*, ofreciendo control preciso sobre el volumen y la granularidad de la información recogida.
- **Gestión de identidades.** La herramienta proporciona una interfaz accesible e intuitiva para registrar, editar y eliminar las identidades conocidas por el sistema HRI. A través de un formulario web, es posible añadir nuevos usuarios mediante la subida de imágenes etiquetadas o la captura directa desde la cámara, asociando un identificador textual a cada rostro. Esta acción desencadena el proceso interno de análisis y aprendizaje del Sistema HRI. Además de la creación, el sistema permite modificar existencialmente los nombres asociados a las identidades existentes, así como eliminar registros de forma individual. Estas operaciones permiten mantener una base de datos actualizada y organizada, facilitando el control por parte del operador y adaptándose a las necesidades cambiantes del entorno de despliegue.
- **Visualización de sesiones.** La interfaz web permite consultar en tiempo real la lista de identidades conocidas y explorar el historial de sesiones de interacción asociadas a cada una. Para cada sesión se presentan tanto datos temporales como métricas cuantitativas sobre las detecciones realizadas, incluyendo gráficas que muestran la evolución temporal de las puntuaciones de detección y clasificación facial, así como la duración y características de cada interacción. Esta representación visual facilita el análisis detallado del comportamiento del sistema y de la relación mantenida con cada persona, aportando una capa adicional de transparencia y supervisión. Así, por ejemplo, podríamos explorar la duración habitual de las interacciones con un usuario en concreto o si en alguna de las sesiones la confianza en la identificación del usuario ha sido baja. Además, la interfaz se actualiza de forma dinámica gracias a un mecanismo de comunicación

en tiempo real con el Sistema HRI. De este modo, los eventos relevantes, como el reconocimiento de una nueva identidad, se reflejan automáticamente en pantalla, sin necesidad de intervención manual por parte del operador.

- **Acceso remoto a la herramienta.** Una de las ventajas de utilizar una interfaz basada en web es la posibilidad de acceder de forma remota a la información almacenada por el sistema. Siempre que el servidor esté configurado adecuadamente, es posible consultar y gestionar los datos desde cualquier dispositivo con conexión a Internet, sin necesidad de acceder físicamente al robot ni a su red local. Esto resulta especialmente útil en escenarios donde los robots operan de forma autónoma en entornos públicos o distribuidos, permitiendo a los responsables del sistema supervisar su funcionamiento de manera individualizada, actualizar la base de datos de identidades o revisar estadísticas de interacción desde ubicaciones remotas. La accesibilidad remota refuerza así la utilidad práctica de la herramienta en contextos reales de despliegue.

### 3. Implementación del sistema

La estructura general de RUMI (Fig. 1) especifica la interacción entre sus módulos centrales para permitir el control remoto del conocimiento social del robot.

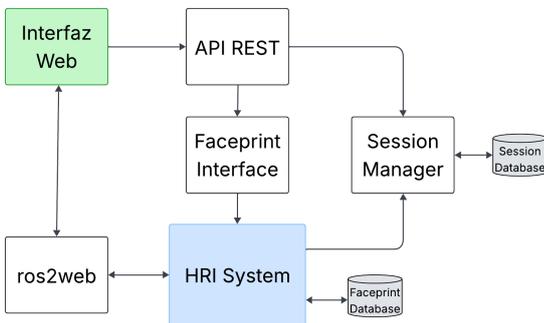


Figura 1: Arquitectura general de RUMI. Interacción entre la Interfaz Web, la API de acceso a datos, los módulos de gestión de sesiones e identidades y el Sistema HRI subyacente.

Su base es el módulo Sistema HRI, responsable de la percepción (por ejemplo, detección y reconocimiento del rostro). Los eventos de detección generados por este sistema son consumidos por el módulo Session Manager, que los ordena por "sesiones de interacción" significativas y administra su almacenamiento. Paralelamente, el módulo ros2web se utiliza como pasarela para transmitir un flujo de datos en tiempo real desde el Sistema HRI (por ejemplo, imágenes o eventos determinados) a la Interfaz Web.

La gestión de la base de datos de identidades del robot se realiza a través de la interfaz de rostros (Faceprint Interface), que abstrae las operaciones CRUD<sup>1</sup> sobre los datos faciales almacenados en el Sistema HRI. Por otro lado, la

API REST sirve como punto central de acceso remoto, proporcionando funcionalidades para leer y modificar tanto las identidades como el historial de sesiones (obtenido desde el Session Manager).

Finalmente, la Interfaz Web interactúa con la API REST para realizar consultas y ediciones, y con ros2web para visualizar información en tiempo real, proporcionando al operador una visión general y actualizada del estado del sistema. Las siguientes subsecciones detallan cada uno de estos módulos.

#### 3.1. Session Manager

El núcleo lógico de RUMI es el módulo Session Manager, encargado de transformar la secuencia continua de detecciones faciales generada por el Sistema HRI en unidades temporales de interacción fácilmente analizables (sesiones de interacción). Así, cada vez que recibe un evento de detección, esto es, un identificador facial junto con sus puntuaciones de localización y clasificación, el módulo decide si abre una nueva sesión, actualiza una existente o cierra las inactivas. Para ello mantiene en memoria un registro de sesiones activas y aplica dos parámetros configurables *on-line*: (i) el tiempo de inactividad tras el cual una sesión se considera finalizada, y (ii) el intervalo mínimo entre detecciones que deben almacenarse dentro de la misma sesión, evitando así la redundancia cuando la tasa de imágenes del sistema de visión es elevada.

Al cerrarse una sesión, todos sus metadatos (marcas temporales de inicio y fin, historial de puntuaciones, identificador del rostro, etc.), se vuelcan de forma automática en una base de datos interna dedicada, preservando la información para análisis posteriores sin interferir con el procesamiento en línea. Este módulo expone, además, un servicio de consulta que permite recuperar el historial completo o filtrado por usuario, funcionalidad que la Interfaz Web aprovecha para generar estadísticas y gráficas, como expondremos en la Sección 3.4. Con esta arquitectura, el módulo Session Manager proporciona al operador un mecanismo robusto y transparente para documentar la interacción humano-robot.

#### 3.2. Interfaz de rostros

Para integrarse con cualquier sistema de reconocimiento facial, RUMI define una interfaz de rostros de alto nivel, denominada Faceprint Interface, que el desarrollador debe implementar con cinco operaciones habituales en el acceso a las bases de datos: (i) crear, (ii) actualizar y (iii) borrar caras, (iv) obtener una cara específica y (v) obtener todas las caras.

Cada uno de estos métodos recibe o devuelve objetos JSON simplificados, de modo que la herramienta permanece agnóstica al motor de base de datos subyacente, al formato interno de los *embeddings* e incluso al lenguaje de programación elegido. En la práctica, basta con que el desarrollador herede la clase abstracta correspondiente y redirija esas llamadas a los servicios o rutinas que ya utilice su propio Sistema HRI (por ejemplo, servicios de ROS, *endpoints* internos o consultas directas a una base de datos).

A nivel de implementación, esta decisión de diseño aporta dos ventajas concretas:

<sup>1</sup> Acrónimo, en inglés, de *Crear, Leer, Actualizar y Borrar*.

- **Acoplamiento mínimo.** El Sistema HRI conserva su arquitectura y nomenclatura originales (ni siquiera tiene por qué estar integrado en ROS); RUMI actúa únicamente como cliente de los métodos CRUD anteriores.
- **Portabilidad directa.** Un mismo despliegue de RUMI puede emplearse con diversos *backends* de reconocimiento, locales o en la nube, mediante la simple sustitución de la clase que implementa la interfaz.

La lógica de gestión de sesiones, descrita en la subsección anterior, no requiere ninguna intervención adicional: basta con que el Sistema HRI publique los eventos de detección en el tópico acordado para que RUMI almacene y visualice automáticamente las interacciones.

### 3.3. API

El servicio API constituye la única puerta de entrada a los datos gestionados por RUMI. Su diseño sigue el estilo *REST* tradicional (Fielding, 2000): peticiones HTTP con mensajes JSON que la propia API valida, pagina y responde utilizando códigos de estado estándar. Internamente, todas las operaciones delegan en la *Faceprint Interface*; así, la lógica del robot queda desacoplada de la tecnología de almacenamiento y la herramienta se puede consultar o administrar desde cualquier dispositivo sin interferir con el grafo ROS 2.

La API resuelve una necesidad frecuente en despliegues HRI: supervisar el sistema, añadir o renombrar rostros y obtener estadísticas agregadas sin detener nodos ni interactuar directamente con la capa DDS de ROS 2. Al operar de forma *stateless* y separada de *ros2web*, mantiene tiempos de respuesta estables incluso cuando la tasa de detecciones es elevada, y facilita la conexión de cuadros de mando externos, scripts de análisis y experimentos A/B.

Funcionalmente, actúa como intermediaria entre los dos núcleos de la arquitectura: por un lado, dirige las operaciones CRUD de identidades a la *Faceprint Interface* implementada por el usuario; por otro, recupera el historial de sesiones almacenado automáticamente por el *Session Manager*. La interfaz web se apoya exclusivamente en estas llamadas, lo que permite desplegar RUMI en redes distintas a la del robot (o incluso a través de Internet) sin exponer *topics* de ROS ni canales *WebSocket*. En conjunto, la API dota al sistema de una capa de acceso remoto robusta y estandarizada, muy adecuada para la supervisión y el mantenimiento de robots sociales en escenarios reales.

### 3.4. *ros2web*

El módulo *ros2web* implementa una pasarela bidireccional entre ROS y el navegador que permite supervisar o, incluso, generar flujos de datos desde una página HTML estándar. Su cometido se solapa parcialmente con *rosbridge\_suite* (Toris et al., 2023), pero difiere en filosofía e implementación: mientras que *rosbridge* expone toda la API de ROS (publicaciones, suscripciones, servicios, parámetros, etc.), *ros2web* se ha escrito desde cero con un objetivo más acotado, reenviando sólo *topics* concretos y notificando eventos ligados a RUMI con una configuración mínima.

Esta herramienta consta de dos componentes acoplados de forma ligera. En primer lugar, un nodo ROS 2 que publica y

reenvía mensajes en un espacio específico de ROS. Este nodo, al iniciarse, puede recibir una lista de *topics* declarada en un archivo *launch* y suscribirse automáticamente a ellos. Adicionalmente, expone un servicio que permite añadir nuevos *topics on-line*, asignándoles un alias legible. En segundo lugar, implementa un servidor *WebSocket*, que corre en un *thread* independiente y mantiene las conexiones con los clientes web. Cada mensaje entrante o saliente al servidor se serializa en JSON mediante un protocolo ligero (tipo y datos), asegurando compatibilidad con cualquier entorno que procese JSON.

Como ejemplo de uso en nuestra herramienta RUMI, gracias a *ros2web* podemos visualizar en el navegador un video anotado en directo, ya que puede generar una ventana flotante que muestra una imagen (típicamente usado para ver la cámara del robot de manera remota). Esta ventana se actualiza en tiempo real y puede ser cerrada y desplazada por el usuario. En otro caso habitual de uso, cuando el Sistema HRI crea, actualiza o elimina un rostro, se publica un mensaje en un *topic* registrado en *ros2web*, de manera que se actualice la galería de usuarios de manera automática y se muestre en el navegador un aviso contextual.

Las principales ventajas que aporta este módulo a la hora de exponer información proveniente de ROS en el navegador incluyen:

- **Configuración mínima:** No requiere modificar nodos existentes, solo exponer los *topics* de interés.
- **Baja latencia:** La conexión entre los mensajes ROS y el servidor *WebSocket* mantiene retardos menores a 10 ms, suficientes para supervisión interactiva.
- **Elasticidad:** Pueden añadirse o retirarse nuevos *topics* dinámicamente, adaptándose a demostraciones, experimentos o entornos públicos con distinto ancho de banda.
- **Seguridad y simplicidad:** A diferencia de *rosbridge*, el cliente web no puede publicar en *topics* arbitrarios, sino que sólo envía mensajes a un espacio determinado y son los nodos receptores quienes deciden cómo procesarlos. Esto evita interferencias accidentales en las comunicaciones entre los nodos de ROS.

En conjunto, *ros2web* dota a RUMI de capacidades de actualización en tiempo real y comunicación bidireccional con un navegador, mejorando la transparencia y la usabilidad del Sistema HRI. No obstante, en este trabajo dicha bidireccionalidad no se ha utilizado, empleándose únicamente para el envío de datos desde el Sistema HRI a la interfaz web.

### 3.5. Interfaz web

La Interfaz Web se articula en tres vistas consecutivas que guían al operador desde la supervisión global del sistema hasta el análisis pormenorizado:

1. **Galería de rostros reconocidos** (Fig. 2). Presenta una cuadrícula dinámica con nombre, identificador y fecha de registro. El operador puede (a) añadir nuevas identidades mediante un asistente guiado o (b) recargar manualmente la lista para obtener la información más reciente. En cualquier caso, la galería se actualiza al instante cuando el sistema HRI crea, renombra o elimina un rostro gracias a los eventos recibidos vía *ros2web*.

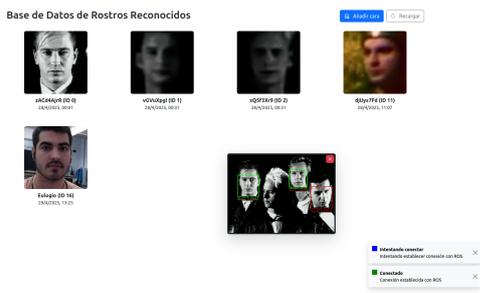


Figura 2: Galería de rostros.

2. **Vista de detalle del usuario** (Fig. 3). Al seleccionar un rostro, la aplicación muestra su ficha completa, que incluye la imagen, la máxima puntuación de detección, el número de descriptores almacenados y la fecha de alta, junto con su lista paginada de sesiones (aunque por espacio esta lista no se muestra en la figura).

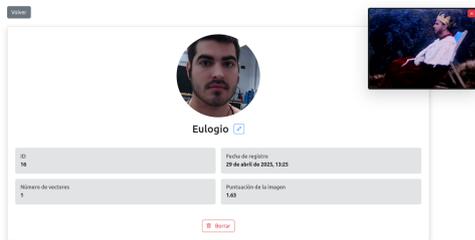


Figura 3: Ficha de identidad.

3. **Panel de análisis de sesión** (Fig. 4). Al seleccionar una sesión se abre una ventana que presenta estadísticas resumidas (media, mediana, desviación, duración) y dos gráficas temporales: evolución de la puntuación de detección y de clasificación.

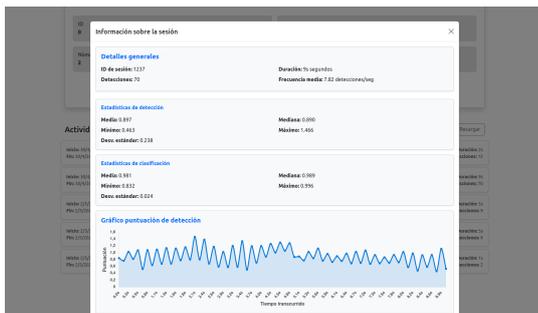


Figura 4: Ventana modal con estadísticas y gráficas de una sesión.

El uso combinado de la API REST (consultas y edición) y de ros2web (eventos en tiempo real) mantiene la información mostrada coherente con el estado interno del Sistema HRI, con latencias del orden de milisegundos. La interfaz es *responsive*, de modo que se adapta tanto a monitores de escritorio como a tabletas de laboratorio; su despliegue sólo requiere exponer la API a la red para habilitar la supervisión remota.

### 3.6. Sistema HRI

RUMI puede integrarse con cualquier arquitectura basada en ROS 2 que implemente la Faceprint Interface y publique los eventos de detección descritos en la Sección 3.1. En

este trabajo hemos integrado nuestra herramienta con el sistema HRI multimodal descrito con detalle en Cañete et al. (2024), cuyo módulo de percepción facial se resume en la Fig. 5.

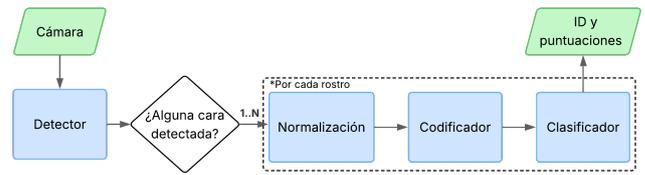


Figura 5: Flujo de detección e identificación facial empleado en nuestro sistema HRI de referencia.

Dicho sistema consta de cuatro etapas:

1. **Detección de rostros.** Utilizamos el detector HOG+SVM de DLIB (King, 2009) por su buen compromiso entre precisión y coste computacional, aunque el *pipeline* es compatible con otros detectores, como MT-CNN (Zhang et al., 2016).
2. **Normalización.** Las caras detectadas se alinean mediante los puntos faciales de DLIB y se normalizan en brillo/contraste en el espacio YCrCb, siguiendo el procedimiento de Baltanas-Molero et al. (2020).
3. **Codificación.** Cada rostro se transforma en un vector de 512 dimensiones empleando la red FaceNet (Schroff et al., 2015), lo que aporta invarianza a pose, iluminación y expresión.
4. **Clasificación.** El vector resultante se compara por distancia coseno con los vectores almacenados. Si la similitud supera un umbral, la cara se asocia a la identidad correspondiente y se actualiza en línea su representación media; en caso contrario el sistema solicita la identidad al operador para poder aprenderla.

Tras la clasificación, el nodo publica un mensaje con el identificador del usuario y sus puntuaciones de detección y reconocimiento. Este mensaje es consumido por el Session Manager, mientras que la información estructurada de cada identidad se expone a través de la implementación concreta de la Faceprint Interface. De este modo, el mismo *pipeline* de visión puede reutilizarse en otros proyectos simplemente sustituyendo la capa de exposición.

## 4. Validación del sistema

Para validar las funcionalidades de RUMI, hemos integrado la herramienta con nuestro robot Sancho, un robot de servicio con base móvil diferencial y una cabeza *pan-tilt* de dos grados de libertad (Fig. 6). La cámara ojo-de-pep y los micrófonos montados en dicha cabeza constituyen la principal fuente sensorial para la interacción humano-robot, mientras que el movimiento de la cabeza, la expresión mostrada en los ojos y un generador de voz y permite responder al usuario.

En los experimentos siguientes, Sancho ejecuta el sistema HRI presentado en la Sección 3.6 y se conecta a RUMI de manera transparente a la arquitectura ROS 2 de su sistema.



Figura 6: Cabeza del robot móvil Sancho empleada en la validación.

#### 4.1. Casos de uso

A continuación describimos cuatro escenarios habituales en robótica que ilustran las capacidades de RUMI.

**Caso A – Pre-inscripción de usuarios.** El operador registra, desde el navegador, la fotografía anotada de un invitado el día anterior de que éste interactúe con Sancho; al producirse la interacción, el robot no necesita información adicional y lo reconoce de forma inmediata.

**Caso B – Registro estadístico sin modificar el sistema HRI.** Durante una jornada de interacción con el robot, se generan miles de eventos de detección. RUMI agrupa automáticamente la información en sesiones y genera estadísticas y gráficas sin intervención manual, accesibles vía API y a través de la interfaz web.

**Caso C – Supervisión remota con vídeo en directo.** Exponiendo el *topic* de la cámara, un operador observa lo que el robot está viendo sin necesidad de estar cerca de él y desde cualquier dispositivo con conexión a Internet.

**Caso D – Edición segura *on-line* de la base de identidades.** Se renombran varias identidades y otras se eliminan mientras el robot continúa operando, sin necesidad de detener nodos ni acceder directamente a la base de datos que el sistema HRI usa para guardar las identidades.

En todos estos escenarios, la actualización de la interfaz web ocurre en tiempo real tras la publicación de los eventos ROS 2, sin apreciarse retardos significativos ni pérdida de fluidez en la interacción. Para demostrar estos casos de uso, se han añadido al repositorio <https://github.com/eulogioqt/rumi> una serie de vídeos ilustrativos de su ejecución.

Estos ejemplos demuestran algunas de las funcionalidades que RUMI proporciona, incluyendo (i) el aprendizaje de nuevas identidades de forma remota, (ii) el análisis de las interacciones de manera transparente al proceso de detección y reconocimiento de caras, (iii) la supervisión de la percepción del robot a distancia y (iv) la gestión de la base de datos sin interrumpir su operación, todo ello integrado en un entorno robótico real con el robot Sancho.

## 5. Conclusiones y trabajos futuros

En este artículo hemos presentado RUMI, una herramienta que ofrece una solución web mínima para supervisar, editar y analizar identidades faciales en sistemas HRI sin necesidad de interrumpir la operación del robot. Probada con el robot móvil Sancho, esta herramienta permite aprender nuevas identidades de forma remota, generar estadísticas automáticas de interacción y mostrar la percepción del robot en tiempo real al operador.

De cara al futuro, prevemos adaptar la salida de RUMI a los estándares de ROS4HRI, incorporar un sistema de permisos que distinga entre usuarios con capacidad de visualización, edición o administración, y extender el enfoque a otros canales multimodales (audio y gestos) manteniendo la misma filosofía de integración ligera sobre eventos de ROS.

## Agradecimientos

Este trabajo ha sido desarrollado en el contexto de los proyectos MINDMAPS (PID2023-148191NB-I00) y Voxeland (JA.B1-09), financiados por el Ministerio de Ciencia e Innovación y la Universidad de Málaga, respectivamente.

## Referencias

- Baltanas-Molero, S., Ruiz-Sarmiento, J. R., Gonzalez-Jimenez, J., 2020. A Face Recognition System for Assistive Robots. In: International Conference on Applications of Intelligent Systems (APPIS). DOI: 10.1145/3378184.3378225
- Cañete, A., Quemada-Torres, E., Ruiz-Sarmiento, J.-R., Moreno, F. Á., Gonzalez-Jimenez, J., 2024. Sistema multimodal para la orientación de robots móviles hacia su interlocutor. *Jornadas de Automática* (45).
- Fielding, R. T., 2000. Architectural styles and the design of network-based software architectures. University of California, Irvine.
- King, D. E., 2009. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research* 10, 1755–1758.
- Lemaignan, S., Juricic, L., 2023. hri\_face\_detect: ROS4HRI-compliant face detection node. [https://github.com/ros4hri/hri\\_face\\_detect](https://github.com/ros4hri/hri_face_detect), accessed: 2025-04-22.
- Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M., Lee, J., et al., 2019. Mediapipe: A framework for perceiving and processing reality. In: Third workshop on computer vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR). Vol. 2019.
- Mohamed, Y., Lemaignan, S., 2021. ROS for Human-Robot Interaction. <https://wiki.ros.org/hri>, accessed: 2025-04-22.
- Schroff, F., Kalenichenko, D., Philbin, J., 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 815–823.
- Toris, R., Lee, J., colaboradores, 2023. rosbridge\_suite — ROS WebSocket interface. [https://wiki.ros.org/rosbridge\\_suite](https://wiki.ros.org/rosbridge_suite), versión 0.11.17, septiembre 2023.
- Wu, W., Peng, H., Yu, S., 2023. YuNet: A Tiny Millisecond-level Face Detector. Machine Intelligence Research, 1–10. DOI: 10.1007/s11633-023-1423-y
- Zhang, K., Zhang, Z., Li, Z., Qiao, Y., 2016. Joint Face Detection and Alignment Using Multi-task Cascaded Convolutional Networks. *IEEE Signal Processing Letters* 23 (10), 1499–1503. DOI: 10.1109/LSP.2016.2603342