

Avatiñ@. Diseño de un avatar robótico social y abierto para la telepresencia de alumnado escolar hospitalizado

TRABALLO FIN DE GRAO
PARA A TITULACIÓN DO
GRAO EN ROBÓTICA

AUTOR:

RAÚL TRILLO

TITORES:

ROBERTO IGLESIAS

DEPARTAMENTO DE CIENCIA DE LA
COMPUTACIÓN E INTELIGENCIA ARTIFICIAL

MARÍA JOSÉ RODRÍGUEZ

CESGA

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

TRABAJO FIN DE GRADO

AVATIÑ@. DISEÑO DE UN AVATAR ROBÓTICO SOCIAL Y ABIERTO PARA LA TELEPRESENCIA DE ALUMNADO ESCOLAR HOSPITALIZADO

AUTOR:

RAÚL TRILLO

TITORES:

ROBERTO IGLESIAS

MARÍA JOSÉ RODRÍGUEZ

*Este trabajo se presenta para cumplir con los requerimientos
normativos exigidos para conseguir el Grado en Robótica
en la*

Escuela Politécnica Superior de Ingeniería.

Fecha de la Convocatoria: Junio 2025



ESCOLA POLITÉCNICA SUPERIOR
DE ENXEÑARÍA

Autorización

Yo, Raúl Trillo, alumno de la titulación del Grado en Robótica de la EPS de Ingeniería de la Universidad de Santiago de Compostela, con DNI 58021695D, autorizo la ubicación de la siguiente copia de mi **Trabajo Fin de Grado** en la biblioteca del Intercentros para que pueda ser consultada por las personas que lo deseen.

Asinado:

Data:

“ No cuentes los días, haz que los días cuenten.”

-Muhammad Ali

Resumen

AVATIÑ@. DISEÑO DE UN AVATAR ROBÓTICO SOCIAL Y ABIERTO PARA LA TELEPRESENCIA DE ALUMNADO ESCOLAR HOSPITALIZADO

por
Raúl Trillo

La ausencia prolongada del entorno escolar por motivos de salud genera graves consecuencias en el desarrollo académico, social y emocional del alumnado, provocando aislamiento y dificultades de reintegración. Aunque existen soluciones de telepresencia comercial como el robot AV1, su elevado coste y naturaleza propietaria limitan drásticamente su accesibilidad. Este Trabajo de Fin de Grado aborda esta problemática a través del diseño, desarrollo e implementación de **Avatiñ@**, un robot de telepresencia social concebido desde una filosofía radicalmente opuesta: ser una plataforma completamente abierta, asequible y personalizable.

El objetivo principal ha sido crear una alternativa funcional y de bajo coste (estimado entre 100 y 200 euros) que democratice el acceso a esta tecnología. Para ello, se ha desarrollado un prototipo completo desde cero. El **hardware** se basa en componentes accesibles como una Raspberry Pi 4 y se aloja en un innovador chasis fabricado íntegramente mediante **impresión 3D**. Este chasis destaca por su sistema de ensamblaje modular tipo puzle, que no requiere tornillos y facilita su montaje y mantenimiento por parte de personal no técnico.

A nivel de **software**, se ha construido una plataforma integral que consta de tres componentes: el software interno del robot, un servidor backend y una interfaz web intuitiva que permite al alumno controlar el robot, moverse por el aula y expresarse a través de "ojos" animados con matrices LED.

El resultado es un prototipo funcional y robusto, listo para una fase de pruebas iniciales, que demuestra la viabilidad de crear soluciones tecnológicas avanzadas con un enfoque social. **Avatiñ@** no solo representa un logro técnico, sino que se postula como una herramienta de alto impacto para mejorar la inclusión y la calidad de vida de los niños hospitalizados, sentando las bases para una futura comunidad de desarrollo que pueda continuar expandiendo su alcance.

Resumo

AVATIÑ@. DISEÑO DUN AVATAR ROBÓTICO SOCIAL E ABERTO PARA A TELEPRESENCIA DE ALUMNADO ESCOLAR HOSPITALIZADO

por
Raúl Trillo

A prolongada ausencia da contorna escolar por motivos de saúde xera graves consecuencias no desenvolvemento académico, social e emocional do alumnado, provocando illamento e dificultades de reintegración. Aínda que existen solucións de telepresenza comercial como o robot AV1, o seu elevado custo e natureza propietaria limitan drasticamente a súa accesibilidade. Este Traballo de Fin de Grao aborda esta problemática a través do deseño, desenvolvemento e implementación de **Avatiñ@**, un robot de telepresenza social concibido desde unha filosofía radicalmente oposta: ser unha plataforma completamente aberta, alcanzable e personalizable.

O obxectivo principal foi crear unha alternativa funcional e de baixo custo (estimado entre 100 e 200 euros) que democratice o acceso a esta tecnoloxía. Para iso, desenvolveuse un prototipo completo desde cero. O **hardware** baséase en compoñentes accesibles como unha Raspberry Pi 4 e alóxase nun innovador chasis fabricado integramente mediante **impresión 3D**. Este chasis destaca polo seu sistema de ensamblaxe modular tipo crebacabezas, que non require parafusos e facilita a súa montaxe e mantemento por parte de persoal non técnico.

A nivel de **software**, construíuse unha plataforma integral que consta de tres compoñentes: o software interno do robot, un servidor backend e unha interface web intuitiva que permite ao alumno controlar o robot, moverse pola aula e expresarse a través de "ollos" animados con matrices LED.

O resultado é un prototipo funcional e robusto, listo para unha fase de probas iniciais, que demostra a viabilidade de crear solucións tecnolóxicas avanzadas cun enfoque social. **Avatiñ@** non só representa un logro técnico, senón que se postula como unha ferramenta de alto impacto para mellorar a inclusión e a calidade de vida dos nenos hospitalizados, sentando as bases para unha futura comunidade de desenvolvemento que poida continuar expandindo o seu alcance.

Abstract

AVATIÑ@. DESIGN OF A SOCIAL AND OPEN ROBOTIC AVATAR FOR THE TELEPRESENCE OF HOSPITALIZED SCHOOL CHILDREN.

by
Raúl Trillo

Prolonged absence from the school environment due to health reasons has serious consequences for students' academic, social, and emotional development, leading to isolation and difficulties with reintegration. While commercial telepresence solutions like the AV1 robot exist, their high cost and proprietary nature drastically limit their accessibility. This Bachelor's Thesis addresses this problem through the design, development, and implementation of **Avatiñ@**, a social telepresence robot conceived from a radically opposite philosophy: to be a completely open, affordable, and customizable platform.

The main objective has been to create a functional and low-cost alternative (estimated between 100 and 200 euros) that democratizes access to this technology. To achieve this, a complete prototype has been developed from scratch. The **hardware** is based on accessible components like a Raspberry Pi 4 and is housed in an innovative **3D-printed** chassis. This chassis stands out for its modular, puzzle-like assembly system, which requires no screws and facilitates easy setup and maintenance by non-technical personnel.

At the **software** level, a comprehensive platform has been built consisting of three components: the robot's internal software, a backend server, and an intuitive web interface that allows students to control the robot, move around the classroom, and express themselves through animated "eyes" using LED matrices.

The result is a functional and robust prototype, ready for an initial testing phase, demonstrating the viability of creating advanced technological solutions with a social focus. **Avatiñ@** not only represents a technical achievement but is also positioned as a high-impact tool to improve the inclusion and quality of life for hospitalized children, laying the groundwork for a future development community that can continue to expand its reach.

Agradecimientos

Estos agradecimientos van dirigidos, antes que nada, a mi familia. A mis padres, por ayudarme incondicionalmente, salieran las cosas bien o mal, simplemente por confiar en mí. Les agradezco infinitamente la educación que me dieron y esa mentalidad de hierro para no rendirme nunca, para luchar contra viento y marea pase lo que pase. Como decimos en gallego: *«nunca choviu que non escampara»*.

A mi hermano, porque gracias a él he madurado en todos los sentidos y ámbitos de mi vida.

Agradezco a mi compañero Xoel Ramos, un apoyo incondicional desde el primer hasta el último día de la carrera. Siempre ha estado ahí, luchando juntos por lo máximo.

También, lógicamente, a todos los compañeros de promoción. Puedo decir sin pelos en la lengua que hemos sido una de las mejores promociones que se podían tener en cuanto a compañerismo.

Además, agradecer tanto al equipo del CESGA, que me brindó la oportunidad de ser uno más en su maravilloso equipo, y agradecer a Roberto por darme la oportunidad de hacer este proyecto.

Por último, uno agradecimiento de los más importantes: a todos mis entrenadores. Practicar un deporte a alto nivel mientras estudias una carrera es complicado, pero ellos lo hicieron posible. Me apoyaron siempre y me enseñaron a no rendirme jamás, a darlo todo hasta el último segundo. Me inculcaron que, hasta que algo no está oficialmente perdido y no hay vuelta atrás, siempre se le puede dar la vuelta a la situación.

Índice General

Autorización	3
Resumen	5
Resumo	6
Abstract	7
Agradecimientos	8
Índice General	9
Capítulo 1	12
Introducción	12
1.1. Estado del Arte y Motivación del Proyecto	12
1.2. Impacto y Beneficios Esperados de Avatiñ@	13
1.3. Desarrollo de Competencias y Metodología de Trabajo	14
1.3.1. Punto de Partida y Evolución de Conocimientos Técnicos	14
1.3. Estructura del Documento	16
2. Diseño e Implementación del Hardware del Robot Avatiñ@	17
2.1 Introducción al Hardware	17
2.1.1. Objetivos del diseño del hardware del robot Avatiñ@	17
2.1.2 Metodología de selección de componentes y diseño	18
2.1.3 Breve comparativa o inspiración conceptual	18
2.2 Diseño y Fabricación del Chasis Robótico	19
2.2.1 Herramientas de Diseño Asistido por Computadora (CAD)	19
2.2.2 Materiales y Tecnología de Fabricación	19
2.2.4 Sistema de Ensamblaje	21
2.2.5 Modularidad y Personalización del Chasis	22
2.3 Componentes Electrónicos y Periféricos	23
2.3.1 Unidad de Procesamiento Central	23
2.3.1.1 Raspberry Pi 4 Modelo B (8GB)	23
2.3.2 Sistema de Percepción Sensorial	23
2.3.3 Sistema de Interacción y Actuación	24
2.3.4 Sistema de Alimentación	25
2.3.5 Componentes Auxiliares y de Interconexión	25
2.4 Integración del Hardware y Esquema de Conexiones	25
2.4.1 Diagrama general de interconexión de todos los componentes	26
2.4.2 Detalle de conexiones a la Raspberry Pi	26
2.5 Grados de Libertad y Cinemática del Robot	27
2.5.1 Movimientos implementados	27
2.5.2 Discusión sobre limitaciones actuales	27
Capítulo 3	28
Diseño e Implementación del Software del Robot Avatiñ@	28
3.1 Introducción al Software	28
2.2 Software Interno del Robot Avatiñ@ (Raspberry Pi)	29
2.2.1 Sistema Operativo y Entorno	29
2.2.2 Control de Actuadores	30

2.2.3 Captura y Tansmisión Multimedia	31
2.2.4 Comunicación con el Servidor	31
2.3 Servidor Web Intermediario	32
2.3.1 Arquitectura y Tecnologías del Backend	32
2.3.2 Gestión de la Comunicación	33
2.3.3 Base de Datos	33
2.4 Interfaz de Usuario (Frontend)	34
2.4.1 Tecnología y Diseño	34
2.4.2 Niveles de Acceso y Funcionalidades	34
2.4.3 Experiencia de Usuario	36
2.5 Desafíos Significativos y Soluciones Implementadas	37
Capítulo 4	38
Mejoras Futuras y Líneas de Trabajo	38
3.1 Mejoras en el Hardware y la Cinemática	38
3.1.1 Evolución de la Cinemática del Robot	38
3.1.2 Optimización de Componentes y Durabilidad	39
3.2 Mejoras en el Software y la Interacción	39
3.2.1 Autonomía de Red y Conectividad	39
3.2.2 Personalización Avanzada de la Expresividad del Robot	39
3.2.3 Experiencia de Usuario Inmersiva	40
3.2.4 Evolución de la Plataforma Web	40
3.2.5 Inteligencia y Automatización (Opcional y a Largo Plazo)	41
3.2.6 Mejoras en las Herramientas de Gestión	41
3.3 Fomento de la Comunidad y Ecosistema Open Source	42
Capítulo 5	43
Impacto Socioemocional y Educativo de Avatiñ@ en el Alumnado Ausente	43
4.1 Mitigación del Aislamiento Social y Fortalecimiento del Bienestar Emocional	43
4.2 Continuidad en el Desarrollo Social y Facilitación de la Reintegración	44
4.3 Apoyo a la Continuidad Académica y al Compromiso con el Aprendizaje	44
4.4 Empoderamiento y Autonomía del Alumno	45
4.5 El Valor Añadido de la Filosofía Open Source y la Personalización de Avatiñ@	45
Capítulo 6	46
Evolución del Proyecto y Proceso de Diseño Iterativo	46
5.1 Punto de Partida y Primeras Iteraciones	46
5.2 Primer Prototipo Físico y Pruebas Iniciales	46
5.3 Desarrollo del Prototipo a Escala Real y Desafíos Mecánicos	47
5.4 Refinamiento de los Movimientos y Actuadores	47
5.5 Integración de Componentes Electrónicos y Pruebas Funcionales	47
5.6 Hacia el Prototipo Funcional Completo	47
5.7 Evolución de la Interfaz de Usuario Web	48
Capítulo 7	49
Resultados y Conclusiones	49
6.1 Presentación de Resultados	49
6.1.1 Hitos Clave en el Desarrollo del Hardware	49
6.1.2 Hitos Clave en el Desarrollo del Software y la Interacción	50
6.1.3 Adecuación del Prototipo para Pruebas Iniciales	51
6.2 Conclusiones Generales	51
Capítulo 8	53
Referencias Bibliográficas y Recursos	53

8.1 Software, Frameworks y Entornos de Ejecución	53
8.1.1. Protocolos de Comunicación y Tecnologías Web	53
8.1.2. Componentes Electrónicos y Hojas de Datos	54
8.1.3. Artículos y Proyectos de Inspiración	54
8.1.4. Recursos Formativos y Asistentes de IA	54

CAPÍTULO 1

INTRODUCCIÓN

La ausencia prolongada del entorno escolar puede generar profundas repercusiones en el desarrollo académico, social y emocional de un niño. Cuando un alumno se ve obligado a permanecer fuera del aula durante periodos extensos, ya sea por motivos de salud u otras circunstancias, no solo enfrenta el desafío de mantenerse al día con el currículo, sino también el riesgo de desconexión de su círculo social y las dificultades de una posterior readaptación. En una etapa tan crucial como la infancia, donde se forjan las bases de la interacción social y la identidad, esta interrupción puede sembrar sentimientos de aislamiento, tristeza y ansiedad, afectando negativamente su bienestar y desarrollo integral. La pérdida del contacto diario con compañeros y profesores puede hacer que el regreso al aula se convierta en un proceso arduo, donde el alumno debe reconstruir lazos y recuperar un sentido de pertenencia.

En este contexto, la tecnología de telepresencia emerge como una herramienta prometedora para mitigar estos efectos, permitiendo al alumno ausente mantener una conexión viva y participativa con su clase. El presente Trabajo de Fin de Grado aborda el diseño, desarrollo e implementación de Avatiñ@, un robot de telepresencia social concebido como una solución innovadora, accesible y, fundamentalmente, "open source".

1.1. Estado del Arte y Motivación del Proyecto

Actualmente, el referente más conocido en el ámbito de robots de telepresencia para estudiantes es el AV1, desarrollado por la empresa noruega No Isolation. Este dispositivo ha demostrado ser valioso al permitir a niños con enfermedades de larga duración o con necesidades de salud emocional, como la ansiedad que les impide asistir a clase (EBSA, por sus siglas en inglés), conectarse con su entorno escolar. El AV1 permite al estudiante ver, oír y hablar con sus compañeros y profesores a través de un robot situado en su pupitre, que puede girar 360 grados y expresar emociones mediante luces. Su objetivo principal es ofrecer conexión social y reducir la soledad, más que ser una herramienta puramente educativa. Diversos estudios y proyectos piloto han destacado sus beneficios en cuanto a mejorar la asistencia, el bienestar emocional y facilitar la reintegración escolar.



Figura 1: Interacción niño-robot

En la edad escolar, no solamente acuden al colegio para aprender materias, sino como herramienta de socialización

y

educación, dos características básicas en cualquier persona, y que se aprenden en su mayor parte, en esta edad.

Sin embargo, el AV1 presenta dos barreras significativas que limitan su adopción generalizada: su elevado coste y su naturaleza propietaria. Adquirir un AV1 puede suponer un desembolso de varios miles de euros (aproximadamente 2.849€ por el hardware más un paquete de servicio anual de 750€, o alquileres mensuales que rondan los 220€), a lo que se pueden sumar costes de mantenimiento anuales considerables para las entidades que gestionan múltiples unidades. Además, al ser un producto de una empresa privada, no ofrece posibilidades de modificación, adaptación o personalización por parte de la comunidad educativa o de desarrolladores independientes.

Esta realidad contrasta con la creciente necesidad de soluciones inclusivas y adaptables en el ámbito educativo. La motivación principal de este proyecto, Avatiñ@, surge de la voluntad de democratizar el acceso a esta tecnología. Se busca ofrecer una alternativa que, inspirándose conceptualmente en la funcionalidad del AV1, rompa con estas barreras económicas y de flexibilidad. El objetivo es crear un robot de telepresencia que no solo sea funcional y efectivo, sino también:

Open Source: Todos los elementos del proyecto, desde los archivos de diseño 3D para la impresión del chasis hasta el software que controla el robot y la interfaz de usuario, se conciben para ser de código abierto. Esto permite que cualquier persona, institución educativa o desarrollador pueda acceder, estudiar, modificar y mejorar el proyecto libremente.

Asequible: Frente a los miles de euros del AV1, el coste de los componentes para construir un robot Avatiñ@ se estima entre 100 y 200 euros, haciéndolo viable para un espectro mucho más amplio de usuarios y centros educativos.

Fácil de Construir y Usar: El diseño mecánico ha sido optimizado para que el ensamblaje sea rápido (en cuestión de pocas horas) y requiera conocimientos técnicos limitados. La interfaz de usuario se ha diseñado para ser intuitiva tanto para los gestores como para el alumno que controla el robot.

Altamente Personalizable: La filosofía open source permite adaptar cada aspecto del robot. Los modelos 3D pueden modificarse para cambiar la estética del robot, hacerlo más amigable o adaptarlo a las preferencias del alumno que lo va a utilizar. El software también puede ser modificado para añadir nuevas funcionalidades o ajustar el comportamiento del robot. Este nivel de personalización es crucial, ya que permite que el robot no sea visto como un dispositivo genérico, sino como una verdadera extensión del alumno. Se busca que, incluso si hay varios Avatiñ@s en una misma clase, cada uno pueda tener una identidad única, representando fielmente al estudiante.

Diseñado para la Aceptación Social: Conscientes de que el robot debe ser un elemento integrador y no de rechazo, se ha buscado un diseño que pueda resultar "mono" y atractivo para los niños, fomentando la interacción natural con el resto del alumnado.

1.2. Impacto y Beneficios Esperados de Avatiñ@

La ausencia escolar prolongada puede tener consecuencias negativas que van más allá del rendimiento académico, afectando el **bienestar emocional** y las **habilidades sociales** del niño. La desconexión del grupo de iguales puede generar sentimientos de soledad y ansiedad, y dificultar la reintegración social una vez que el alumno regresa físicamente a la escuela.

Avatiñ@ se propone como una herramienta para contrarrestar estos efectos negativos, ofreciendo múltiples beneficios:

Continuidad Social y Emocional: Permite al alumno ausente mantener el contacto diario con sus amigos y profesores, participar en las conversaciones, las bromas y la vida social del aula como si estuviera allí. Esto es vital para su desarrollo emocional y para preservar su sentido de pertenencia al grupo.

Reducción del Estrés de Readaptación: Al mantenerse conectado, la vuelta física al colegio es menos traumática. El alumno no se siente como un extraño, ya que ha seguido formando parte de la dinámica de la clase. El objetivo es que la readaptación sea lo más rápida posible o, idealmente, que no exista tal periodo de readaptación problemático.

Participación Académica: Aunque el foco principal es la conexión social, Avatiñ@ también facilita que el alumno pueda seguir las explicaciones del profesor, ver la pizarra y participar en las actividades de clase, ayudando a minimizar el impacto académico de su ausencia.

Empoderamiento del Alumno: El control directo sobre el robot (movimiento, visión, voz) otorga al alumno un sentido de agencia y presencia activa en el aula, en lugar de ser un mero espectador pasivo.

Apoyo en una Etapa Crítica: La infancia es un periodo de alta vulnerabilidad y plasticidad en cuanto al desarrollo social. Intervenciones como Avatiñ@ pueden tener un impacto significativo a largo plazo, previniendo problemas de aislamiento y fomentando habilidades sociales saludables.

Inclusión y Normalización: La presencia de Avatiñ@ puede ayudar a normalizar la situación del alumno ausente y fomentar la empatía y el apoyo por parte de sus compañeros.

El proyecto Avatiñ@, por tanto, no solo representa un desafío técnico en términos de diseño de hardware y software, sino que también aspira a tener un **impacto social positivo**, ofreciendo una solución práctica, económica y adaptable para mejorar la calidad de vida y la inclusión educativa de niños que atraviesan períodos difíciles de ausencia escolar. Al ser una plataforma abierta, se espera que la comunidad pueda contribuir a su evolución.

1.3. Desarrollo de Competencias y Metodología de Trabajo

La realización del proyecto Avatiñ@ ha trascendido la mera aplicación de conocimientos previamente adquiridos, constituyendo un intenso proceso de aprendizaje y desarrollo de nuevas competencias técnicas y metodológicas. Si bien se partía con una base sólida en ciertos ámbitos, la naturaleza multidisciplinar del proyecto exigió una notable dedicación a la autoformación y a la superación de desafíos en áreas menos familiares.

1.3.1. Punto de Partida y Evolución de Conocimientos Técnicos

En cuanto a los conocimientos de partida, se contaba con una familiaridad con el lenguaje de programación Python, fundamental para el desarrollo del software interno del robot en la Raspberry Pi. Asimismo, existía una comprensión básica del funcionamiento de bases de datos y servidores, aunque no al nivel de profundidad que requirió la implementación de la plataforma web completa.

En el ámbito del hardware, la experiencia previa con la Raspberry Pi facilitó el manejo de componentes como los servomotores, si bien la optimización de su comportamiento exigió una investigación

considerable. El diseño 3D, gracias a las asignaturas cursadas durante la carrera, proporcionó una buena base, pero la complejidad del chasis de Avatiñ@ y la necesidad de optimizarlo para la impresión 3D impulsaron el aprendizaje de técnicas avanzadas, el manejo de nuevas extensiones de software CAD y un profundo conocimiento de los parámetros de impresión para equilibrar tiempo, coste y calidad estructural. Por otro lado, la interconexión de los componentes mediante cableado se consideró una tarea relativamente sencilla.

1.3.2. Superación de Desafíos en Desarrollo Web

El mayor desafío técnico residió en el desarrollo web, concretamente en el dominio de los lenguajes JavaScript, HTML y CSS, que eran cruciales para la implementación tanto del backend como del frontend del servidor. Para acometer esta tarea con garantías, fue necesario realizar dos cursos especializados a través de la plataforma Udemy, que proporcionaron los fundamentos y las técnicas avanzadas para construir una aplicación web robusta y funcional.

1.3.3. Metodología de Trabajo y Uso de Herramientas de IA

La envergadura del proyecto, que se traduce en miles de líneas de código, implicó también la adopción de una metodología de trabajo eficiente. En este sentido, una de las claves fue el aprendizaje y la optimización en el uso de diversas tecnologías y herramientas actuales, incluyendo modelos de inteligencia artificial como ChatGPT, Gemini y Cursor, entre otros. Lejos de suponer una sustitución del esfuerzo propio, estas herramientas se emplearon como catalizadores del proceso de desarrollo y aprendizaje.

La técnica utilizada consistió en un ciclo iterativo, que se puede desglosar en los siguientes pasos:

- **Primero**, definir la funcionalidad deseada.
- **Segundo**, solicitar asistencia para la generación de fragmentos de código específicos o para la resolución de dudas conceptuales.
- **Tercero**, verificar rigurosamente el funcionamiento del código propuesto.
- **Cuarto**, y fundamental, comprender en profundidad la lógica y la estructura de dicho código
- **Finalmente**, integrarlo y adaptarlo al conjunto del proyecto.

Esta aproximación metodológica no solo aceleró significativamente el desarrollo, sino que también enriqueció el proceso de aprendizaje, permitiendo asimilar nuevos conceptos y técnicas de programación de manera práctica y aplicada. Considero que esta es una forma inteligente y productiva de interactuar con las herramientas de IA en el contexto del desarrollo de software complejo.

Este documento detallará en los siguientes capítulos el proceso de diseño e implementación tanto del hardware como del software del robot Avatiñ@, así como los resultados obtenidos y las futuras líneas de trabajo.

1.3. Estructura del Documento

- **Capítulo 1: Diseño e Implementación del Hardware del Robot Avatiñ@.** Este capítulo detalla exhaustivamente el proceso de diseño mecánico del robot, la selección de materiales y componentes electrónicos, la tecnología de fabricación mediante impresión 3D, el innovador sistema de ensamblaje sin tornillos, los grados de libertad implementados y la integración de todos los elementos físicos que constituyen el robot.
- **Capítulo 2: Diseño e Implementación del Software del Robot Avatiñ@.** En esta sección se describe la arquitectura del software, abarcando tanto el firmware interno del robot desarrollado en Python para la Raspberry Pi (control de actuadores, gestión de sensores, comunicación multimedia con WebRTC/WebSockets), como el servidor backend (Node.js) que actúa de intermediario, y la interfaz de usuario frontend (React) con sus diferentes niveles de acceso y funcionalidades para la gestión y control del robot. También se abordará la estructura de la base de datos (MariaDB) y los protocolos de comunicación.
- **Capítulo 3: Mejoras Futuras y Líneas de Trabajo.** Este capítulo explora las posibles evoluciones y mejoras tanto para el hardware como para el software del proyecto Avatiñ@. Se discutirán ideas como la integración de conectividad 4G, el desarrollo de un sistema de creación de emojis personalizados, la evolución de la interfaz a una Progressive Web App (PWA), y otras posibles funcionalidades que podrían enriquecer la experiencia del usuario y la versatilidad del robot.
- **Capítulo 4: Impacto Socioemocional y Educativo de Avatiñ@ en el Alumnado Ausente.** Este apartado profundiza en los beneficios que un robot de telepresencia como Avatiñ@ puede aportar al alumnado que se enfrenta a largos periodos fuera de la escuela, abordando su bienestar emocional, desarrollo social y continuidad académica, y apoyándose en estudios e investigaciones relevantes.
- **Capítulo 5: Evolución del robot Avatiñ@.** Se narra el viaje de desarrollo del robot Avatiñ@, detallando su transformación desde la concepción inicial hasta el prototipo funcional. Se expondrá el proceso iterativo seguido, comenzando con las pruebas de los primeros diseños a escala y los desafíos mecánicos afrontados, como la optimización de la fluidez en los rodamientos. Se describirá cómo, a través de la prueba y el error, se solucionaron problemas clave de diseño en los enganches y la transmisión de fuerza, culminando con la exitosa integración de la electrónica y la evolución en paralelo de la interfaz web.
- **Capítulo 6: Resultados y Conclusiones.** Finalmente, este capítulo resumirá las principales contribuciones del trabajo, los resultados obtenidos en el desarrollo del prototipo funcional, reiterará la importancia y el impacto potencial del proyecto Avatiñ@, y ofrecerá una reflexión final sobre el aprendizaje obtenido y las perspectivas futuras que se abren a partir de esta iniciativa open source.

Capítulo 2

2. DISEÑO E IMPLEMENTACIÓN DEL HARDWARE DEL ROBOT AVATIÑ@

2.1 Introducción al Hardware

En la presente sección, detallo el diseño, los componentes y la implementación del hardware del robot Avatiñ@, un proyecto desarrollado íntegramente en el marco de mi Trabajo de Fin de Grado. Si bien he tomado como referencia conceptual y estética externa una versión anterior del proyecto Avatiñ@, así como ideas conceptuales exploradas en plataformas colaborativas como Thingiverse, considero fundamental subrayar que todo el diseño mecánico, la selección de componentes, la optimización estructural y la adaptación funcional han sido desarrollados y rediseñados al 100% por mí. Este proyecto se enmarca en una iniciativa con la entidad CESGA (Centro de Supercomputación de Galicia), vislumbrando un considerable potencial para su desarrollo y aplicación futura.

2.1.1. Objetivos del diseño del hardware del robot Avatiñ@

Durante el desarrollo del hardware del robot Avatiñ@, ha habido una serie de objetivos clave, con la intención de crear un sistema robótico eficiente, adaptable y replicable:

2.1.1.1 Optimización del espacio y distribución del peso

Buscando un diseño compacto que aloje de manera eficiente todos los componentes necesarios, asegurando al mismo tiempo una distribución de peso equilibrada para garantizar la estabilidad y correcta operatividad del robot.

2.1.1.2 Modularidad y facilidad de mantenimiento

Concibiendo el diseño de forma modular, permitiendo el fácil reemplazo o actualización de componentes individuales sin necesidad de modificar sustancialmente el resto del sistema. Esto, a su vez, simplifica las tareas de mantenimiento y reparación.

2.1.1.3 Bajo coste y accesibilidad de componentes

Ha habido una prioridad el uso de componentes electrónicos y materiales de fabricación que sean económicos y fácilmente accesibles, con el fin de hacer el proyecto viable y replicable.

2.1.1.4 Replicabilidad mediante impresión 3D

Uno de los pilares del diseño creado es la capacidad de fabricar la gran mayoría de las piezas estructurales del chasis mediante tecnología de impresión 3D, lo que facilita su construcción y la posibilidad de que otros puedan replicar o adaptar el diseño.

2.1.2 Metodología de selección de componentes y diseño

La metodología que seguida para la selección de componentes y el diseño del hardware se ha basado en un proceso iterativo. Partiendo de una investigación inicial de funcionalidades y características deseadas, tomando como referencia conceptual el robot AV1 de No Isolation. Sin embargo, la selección final de cada pieza de hardware, así como el diseño completo del chasis, los sistemas de movimiento y los soportes internos, son resultado de un trabajo de diseño original.

Este proceso ha implicado:

- La identificación de requisitos funcionales para cada subsistema del robot.
- La búsqueda y evaluación de componentes comerciales que cumplan dichos requisitos, considerando su compatibilidad, coste y disponibilidad. En los casos donde la información detallada de componentes específicos de sistemas de referencia no estaba disponible, busqué piezas equivalentes que se asemejaran en funcionalidad para el modelo Avatiñ@.
- El diseño 3D de todas las piezas mecánicas y estructurales desde cero, utilizando software CAD. Esto incluye todos los engranajes, uniones y adaptaciones específicas, incluso si la inspiración conceptual inicial provino de plataformas públicas o diseños previos. Cada elemento ha sido rediseñado y adaptado por mí para los requisitos específicos de Avatiñ@.
- El prototipado y pruebas iterativas para validar tanto la selección de componentes como el diseño mecánico.

Es importante destacar que este enfoque asegura que, aunque parta de un análisis inicial como base, el desarrollo y la implementación final del hardware constituyen una contribución original.

2.1.3 Breve comparativa o inspiración conceptual

Como punto de partida para definir el propósito y las funcionalidades generales del robot Avatiñ@, se ha investigado el robot AV1 de No Isolation. El objetivo de esta investigación preliminar fue comprender mejor el diseño y el propósito de un robot de telepresencia social, y así establecer una base sólida sobre la cual construir mi proyecto Avatiñ@.

No obstante, esta inspiración es puramente conceptual en términos de funcionalidad y nicho de aplicación. El robot Avatiñ@, en su diseño de hardware, componentes internos, mecánica y estructura, es un desarrollo completamente nuevo y diferenciado. Cualquier similitud estética con versiones previas o conceptos generales se limita a eso, a una inspiración visual o funcional inicial, siendo toda la ingeniería y el diseño detallado una obra original. Este documento se centra en el borrador actual del robot Avatiñ@, y con consciencia de que algunos detalles podrían evolucionar a medida que se avanza en el desarrollo, especialmente considerando su potencial vinculación con el CESGA y futuras líneas de trabajo.

2.2 Diseño y Fabricación del Chasis Robótico

El chasis constituye la espina dorsal del robot Avatiñ@, proporcionando soporte estructural a todos los componentes internos y externos, así como los mecanismos para su movilidad. El diseño y fabricación del chasis han sido procesos cruciales, enfocados en la optimización para la impresión 3D, la robustez, la facilidad de ensamblaje y la modularidad.

2.2.1 Herramientas de Diseño Asistido por Computadora (CAD)

2.2.1.1 Uso de FreeCAD y sus extensiones

Para la totalidad del diseño mecánico del chasis, se ha utilizado FreeCAD, un software de modelado paramétrico 3D de código abierto. Ha habido una elección por esta herramienta debido a su flexibilidad, su capacidad para manejar diseños complejos y la disponibilidad de diversas extensiones (workbenches) que facilitan tareas específicas de modelado. Todo el chasis, con sus numerosos detalles, optimizaciones para la distribución de peso y alojamientos específicos para el hardware, ha sido modelado íntegramente en FreeCAD.

2.2.2 Materiales y Tecnología de Fabricación

2.2.2.1 Selección de material para impresión 3D

Aunque el diseño es compatible con diversos filamentos termoplásticos, las pruebas iniciales y la mayoría de los componentes se han impreso utilizando PLA (Ácido Poliláctico), debido a su facilidad de impresión, buena rigidez y bajo coste. No obstante, para piezas que pudieran requerir mayor resistencia al impacto o a la temperatura, se podría considerar el uso de PETG o ABS, realizando los ajustes pertinentes en los parámetros de impresión.

2.2.2.2 Proceso de impresión 3D y consideraciones

Todas las piezas del chasis han sido diseñadas para ser fabricadas mediante tecnología de FDM (Modelado por Deposición Fundida). Han sido optimizadas las orientaciones de las piezas para minimizar la necesidad de soportes y asegurar la resistencia mecánica en las direcciones críticas. Las tolerancias entre piezas han sido cuidadosamente ajustadas para permitir un encaje preciso sin holguras excesivas.

2.2.3 Descripción Detallada de la Estructura Mecánica

La estructura del chasis se compone de varias partes interconectadas que definen su forma y funcionalidad:

2.2.3.1 Base del robot y sistema de giro horizontal

El robot cuenta con una base fija que se une a una plataforma giratoria.

- **Diseño del rodamiento con cilindro:** Entre la base fija y la plataforma giratoria ha sido integrado un rodamiento de gran diámetro diseñado a medida. Este rodamiento incorpora cilindros que ruedan en pistas concéntricas, con el espacio suficiente para permitir un giro horizontal de 360° de forma extremadamente suave y eficiente, denominado "hiperóptimo" por su bajo rozamiento.

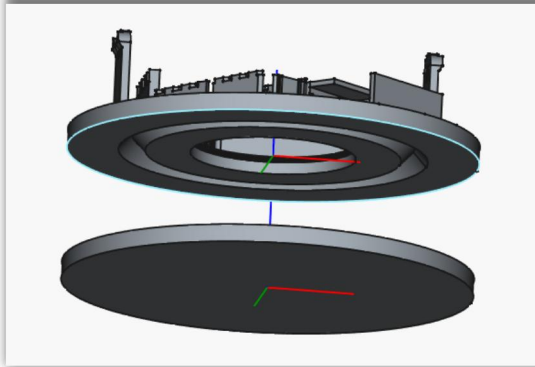


Figura 2: Rodamiento

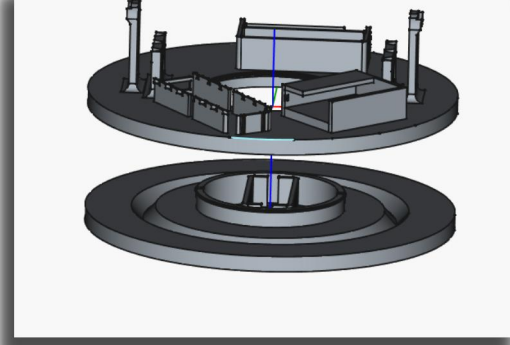


Figura 3: Rodamiento

2.2.3.2 Cuello del robot y sistema de giro vertical

Sobre la plataforma giratoria se erige el cuerpo principal, que culmina en un cuello también con capacidad de movimiento.

- **Diseño del rodamiento del cuello:** El cuello implementa un sistema de rodamiento similar al de la base para permitir el movimiento de inclinación vertical de la cabeza. Sin embargo, en este caso, debido a las cargas y la geometría específica, prescindiendo de los cilindros, ya que el diseño de las superficies de contacto es suficiente para garantizar un movimiento fluido y preciso sin ellos. En este caso hay que hacer una fuerza elevada para la introducción entre el cuello y la parte de la cabeza, además como vemos en la parte derecha, tenemos el hueco para el motor, que con la parte de la conexión de la cabeza, generará el movimiento perfecto.

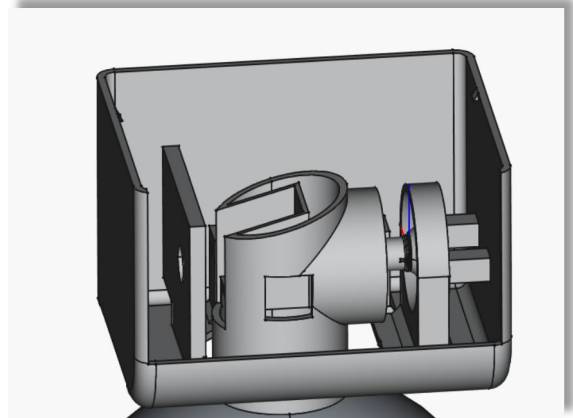


Figura 4: Enganche cuello-cabeza

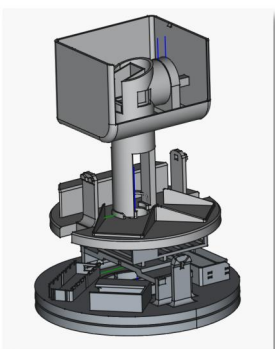


Figura 5: interior del avatiñ@

2.2.3.3 Estructura interna: compartimentos ("jaulas") para hardware

El interior del chasis está meticulosamente diseñado. Han sido creado cavidades y "jaulas" específicas para cada componente electrónico principal (Raspberry Pi, protoboard, baterías, módulo de carga, etc.), asegurando su correcta ubicación, sujeción y una óptima distribución del peso. Cada pieza tiene su lugar designado,

minimizando el espacio no utilizado.

2.2.3.4 Diseño de la cabeza intercambiable

La cabeza del robot está concebida como un módulo independiente. Se une al cuello mediante un sistema de enganches estandarizado en mi diseño, lo que permite su fácil desmontaje y la posibilidad de diseñar e imprimir cabezas con diferentes estéticas o funcionalidades (por ejemplo, con diferentes disposiciones de sensores o actuadores) sin alterar el resto del robot.

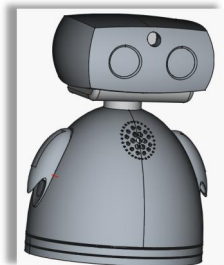


Figura 6: Opción 1 avatiñ@

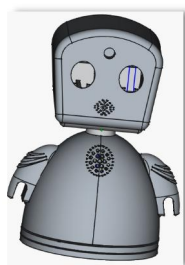


Figura 7: Opción 2 avatiñ@

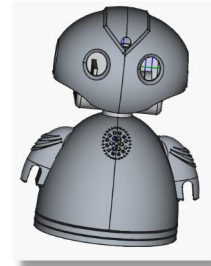


Figura 8: Opción 3 avatiñ@

2.3.4 Transmisión de movimiento entre motor y base o cabeza

La **transmisión de movimiento** desde el servomotor a la base o cabeza se realiza de manera eficiente. Dado que el enganche del servomotor es pequeño, se ha optado por utilizar una de las bases que acompañan al motor, específicamente la de **forma de cruz**. Se ha diseñado un hueco en la pieza receptora que se ajusta perfectamente a esta base, asegurando una **transmisión de movimiento óptima y sin deslizamientos**.

Adicionalmente, para maximizar la **fuerza aplicada a la base**, las conexiones con esta pieza se han ubicado lo más lejos posible del centro de rotación. Esta configuración aprovecha el principio de la palanca, permitiendo que el motor ejerza un mayor par y, en consecuencia, mueva la estructura con mayor eficacia.

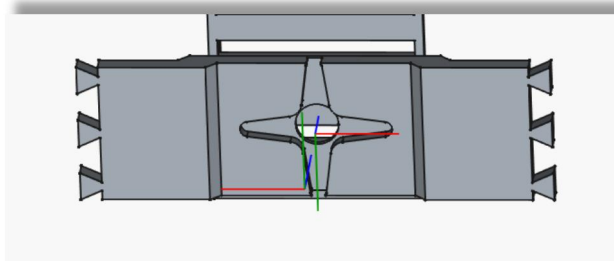


Figura 9: Enganche motor-base

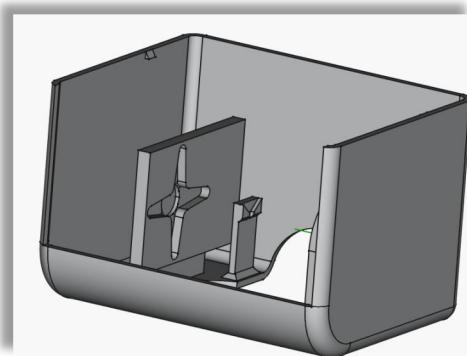


Figura 10: Enganche motor-cabeza

2.2.4 Sistema de Ensamblaje

Uno de los aspectos más innovadores de mi diseño es el sistema de ensamblaje:

2.2.4.1 Uniones y enganches: diseño tipo puzle/LEGO

El chasis se compone de 9 piezas impresas principales, además de las tapas para las diferentes "jaulas" de los componentes. Todas estas piezas se unen entre sí mediante un sistema de enganches, pestañas y ranuras que he diseñado específicamente, de manera similar a un puzle o a las piezas de LEGO.

Figura 12 Enganche del Raíl unión brazo-cuerpo

Figura 14: Enganche cuerpo-base



Figura 11: Raíl unión brazo-cuerpo

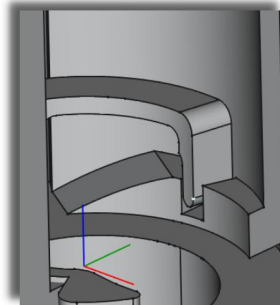
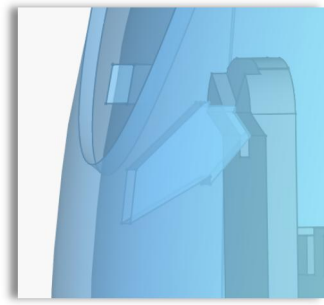


Figura 13: Enganche interior cuerpo-cuello



2.2.4.2 Ausencia de tornillos y adhesivos

Es fundamental destacar que no se requiere ningún tornillo ni pegamento para el ensamblaje completo del chasis. Todas las uniones son mecánicas, basadas en la propia geometría de las piezas impresas.

2.2.4.3 Proceso de montaje y desmontaje (consideraciones sobre fragilidad de enganches)

El montaje, una vez impresas y limpias de soportes las piezas, es rápido e intuitivo, pudiendo completarse en aproximadamente 20-30 minutos. Los enganches están diseñados para proporcionar una unión robusta y con prácticamente ninguna holgura. Si bien todas las uniones están pensadas para poder desensamblarse en caso de necesitar acceder al interior o reemplazar alguna pieza, es importante manipularlas con cuidado, ya que algunos de los elementos de enganche, por su naturaleza, pueden ser las partes más frágiles durante el proceso de desmontaje.

2.2.4.4 Número de piezas principales

Como se mencionó, el cuerpo estructural principal consta de 9 piezas, a las que se suman las tapas específicas para cada compartimento de hardware.

2.2.5 Modularidad y Personalización del Chasis

La filosofía de diseño del chasis se ha centrado en la modularidad y la facilidad de adaptación:

2.2.5.1 Facilidad de reemplazo de piezas

Si cualquier pieza del chasis sufre algún daño, puede ser reimpresa individualmente en cuestión de horas y sustituida sin mayor complicación, gracias al sistema de ensamblaje sin adhesivos.

2.2.5.2 Adaptabilidad a futuros cambios de hardware

En el caso de que se necesite integrar un nuevo componente de hardware o sustituir uno existente por otro con dimensiones diferentes, los archivos fuente de FreeCAD pueden ser modificados de forma relativamente sencilla. Esto permite adaptar la "jaula" o el soporte correspondiente al nuevo elemento sin tener que rediseñar todo el chasis.

2.2.5.3 Opciones de personalización estética (cabeza)

Como ya he mencionado, la cabeza es un módulo intercambiable. Manteniendo el mismo sistema de anclaje al cuello, se pueden diseñar e imprimir cabezas con formas y apariencias muy diversas. Esto abre la puerta a personalizar el robot para diferentes usuarios (por ejemplo, alumnos de distintas edades) o para distintos entornos de aplicación, buscando siempre la interacción más amigable posible, incluso sin poseer conocimientos avanzados de diseño 3D para realizar modificaciones estéticas básicas.

2.3 Componentes Electrónicos y Periféricos

La selección de los componentes electrónicos ha sido una fase crítica, buscando un equilibrio entre funcionalidad, coste, disponibilidad y facilidad de integración. A continuación, describo los principales componentes que dan vida al robot Avatiñ@.

2.3.1 Unidad de Procesamiento Central

2.3.1.1 Raspberry Pi 4 Modelo B (8GB)

Figura 15: Raspberry pi 4

Ha sido seleccionado este modelo como cerebro del robot por su alta eficiencia en comparación con otros modelos y su consumo de energía relativamente moderado. La versión de 8GB de RAM ofrece una capacidad de procesamiento holgada para las tareas previstas, incluyendo procesamiento de vídeo, audio y control de actuadores. Una ventaja significativa es la integración de conectividad Wi-Fi y Bluetooth en la propia placa, lo que elimina la necesidad de tarjetas de red externas o módulos adicionales, simplificando el diseño.



2.3.2 Sistema de Percepción Sensorial

Para que el robot pueda interactuar con su entorno, han incorporado los siguientes sensores:

2.3.2.1 Visión: Módulo de Cámara Raspberry Pi



Figura 16: Cámara ojo de pez

Optando por un módulo de cámara ojo de pez compatible con Raspberry Pi. Este tipo de cámaras se conecta directamente a la interfaz CSI (Camera Serial Interface) de la Raspberry Pi, permitiendo una transmisión de datos de alta velocidad y bajo consumo. La elección específica del módulo permite capturar el entorno visual del robot. Cabe destacar que cualquier cámara compatible con la raspberry pi 4 se podría conectar, desde las normales como pueden ser la raspberry pi camera V3, la más moderna y de mejor calidad, que se está llegando a comparar con la cámara de móviles de alta gama como iphone.

2.3.2.2 Audio (Entrada): Micrófono Omnidireccional I2S INMP441

Para la captación de audio, he integrado el micrófono INMP441. Este componente se caracteriza por su interfaz I2S, presenta una captación de sonido nítida, y su patrón de captación omnidireccional, que le permite recoger sonido desde todas las direcciones. Con un consumo ultra bajo es ideal para dispositivos portátiles alimentados por batería. Al igual que con la cámara, cualquier micrófono que se conectase por jack, sería completamente compatible.

2.3.3 Sistema de Interacción y Actuación

El robot Avatiñ@ interactúa y se expresa mediante los siguientes componentes:

2.3.3.1 Visual (Ojos): Matriz LED 8x8 con controlador MAX7219

Para dotar al robot de una forma de expresión visual, he utilizado dos módulos de matriz LED de 8x8, cada uno controlado por un chip MAX7219. Cada matriz consta de 64 LEDs individuales y opera a 5V DC. Se comunican mediante una interfaz serie de 4 cables y permiten el control de brillo tanto analógico como digital. En el diseño actual se utiliza módulos independientes para cada "ojo". Este es uno de los componentes más importantes del robot, debido a la interacción con el entorno que le puede dar, creando emociones y dando información al ambiente.

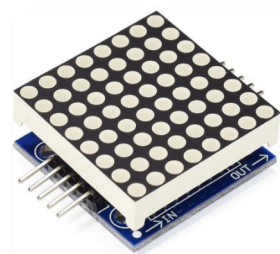


Figura 17: Matriz led 8x8

2.3.3.2 Audio (Salida): Altavoz USB Plug & Play

Para la reproducción de sonido, ha sido seleccionado un conjunto de altavoces USB diseñados específicamente para funcionar con Raspberry Pi, ofreciendo una integración sencilla mediante conexión USB.



Figura 18: Altavoces

2.3.3.3 Movimiento: Servos FS5103R

Los movimientos de giro horizontal de la base y vertical del cuello se realizan con servos FS5103R de rotación continua. Elegidos por su capacidad de giro continuo y su torque adecuado para la aplicación.

Si bien estos servos no tienen memoria de posición, este inconveniente se resolverá eficazmente mediante el software. Además, aunque su torque es moderado, el excelente diseño del modelo 3D compensa esta limitación, permitiendo un rendimiento óptimo en la aplicación.

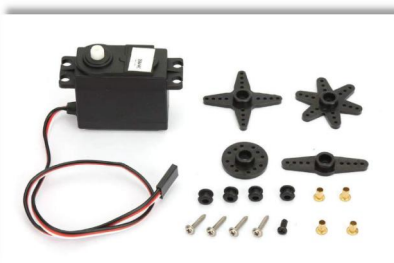


Figura 19: servomotor FS5103R

2.3.4 Sistema de Alimentación

He diseñado un sistema de alimentación autónomo para el robot:

2.3.4.1 Fuente de Energía: Baterías recargables

La alimentación principal proviene de dos baterías de Li-Ion del tipo 21700, cada una con una capacidad de 4000 mAh y un voltaje nominal de 3.7V, sumando un total de 8000 mAh de capacidad si se configuran adecuadamente para el suministro de energía. Estas baterías pueden ser reutilizadas, de una bicicleta, coche o similar que esté inutilizado, ya que son las pilas estándar. Además de esto, como se pueden reemplazar fácilmente, quitamos la variable de que se puedan "viciar" las pilas. En cuanto detectemos este fenómeno, simplemente se cambian de forma sencilla.



Figura 20: Pila

2.3.4.2 Circuitería de Carga: Módulo de carga

Para la recarga de las baterías, he incorporado un módulo de carga para baterías de Li-Ion. Estos módulos típicamente incluyen protección contra sobrecarga, sobredescarga y cortocircuito, y permiten la carga a través de una conexión USB.

2.3.4.3 Conexión de Carga: Clavija USB Tipo C

Para facilitar la carga, he integrado en el chasis un conector USB Tipo C hembra con cables, proporcionando una interfaz moderna y robusta para la alimentación externa durante la carga.



Figura 21:
Femina tipo C

2.3.5 Componentes Auxiliares y de Interconexión

2.3.5.1 Protoboard

Para facilitar las conexiones entre los diferentes módulos electrónicos, sensores y la Raspberry Pi, he incluido una protoboard, permitiendo un prototipado rápido y modificaciones sencillas en el circuito.



Figura 22:
Protoboard

2.3.5.2 Cables de Puente (Jumper Wires)

La interconexión de los componentes en la protoboard y hacia la Raspberry Pi se realiza mediante cables de puente de varios tipos (macho-macho, macho-hembra, hembra-hembra), como los referenciados en el kit de Heevhas.



Figura 23: Cables

2.4 Integración del Hardware y Esquema de Conexiones

Una vez seleccionados y descritos los componentes individuales, la siguiente fase crucial es su correcta integración y conexión. Planificando cuidadosamente cómo cada elemento se interconecta para asegurar un funcionamiento armónico del sistema.

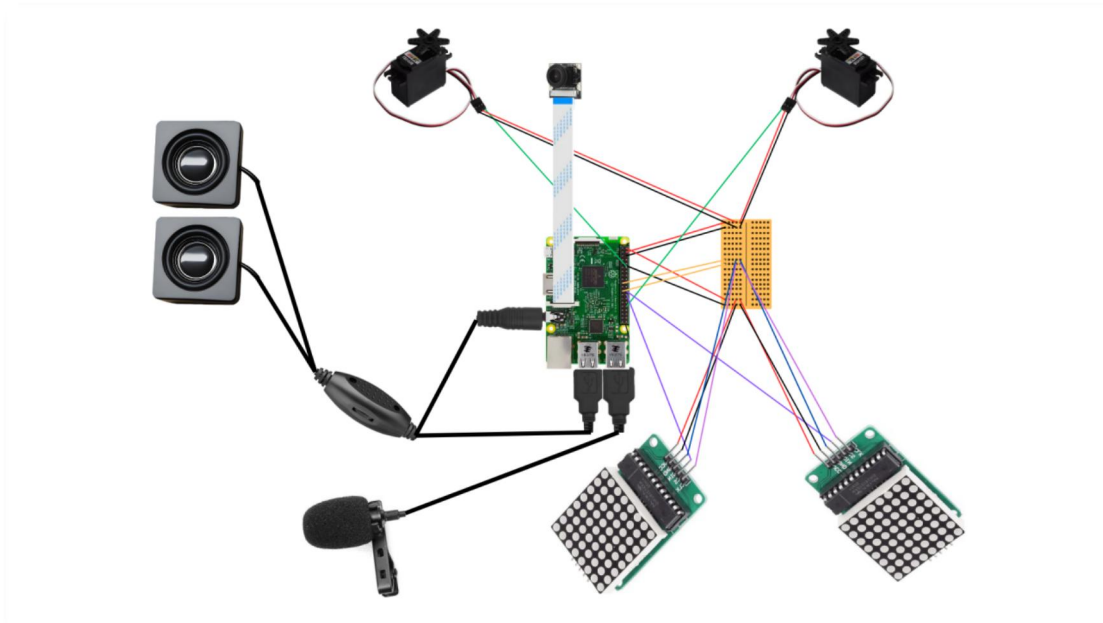


Figura 24: Esquema conexiones

2.4.1 Diagrama general de interconexión de todos los componentes

He elaborado un diagrama de conexiones que ilustra las relaciones entre la Raspberry Pi, los sensores, los actuadores, el sistema de alimentación y los periféricos. Este diagrama sirve como guía fundamental durante el ensamblaje y para futuras tareas de depuración o modificación. La protoboard juega un papel central en la organización de muchas de estas conexiones, especialmente para la distribución de alimentación y las señales de control de bajo voltaje.

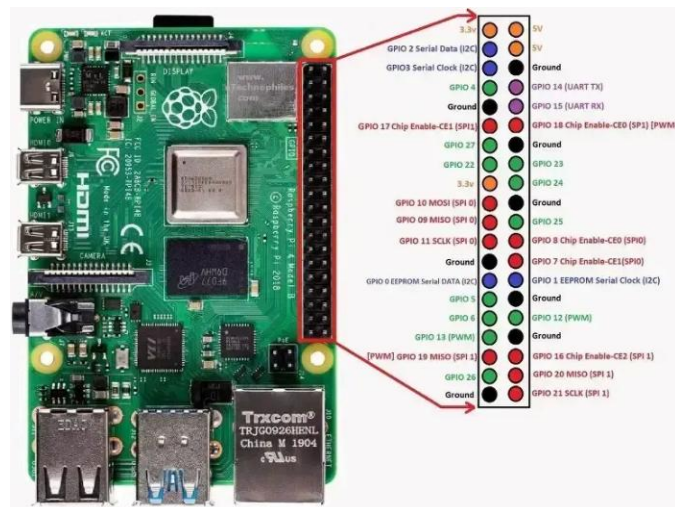
2.4.2 Detalle de conexiones a la Raspberry Pi

Una vez seleccionados los componentes, la fase de integración consiste en conectarlos de manera ordenada y funcional. Para facilitar la comprensión del sistema eléctrico y de datos, he diseñado un esquema general que ilustra todas las interconexiones.

Figura 25: Raspberry Pi conexiones

Como se puede observar en la Figura 24, todos los periféricos —servomotores, cámara, matrices LED, micrófono y altavoces— se conectan en última instancia a la Raspberry Pi 4, que actúa como el cerebro del robot. La protoboard juega un papel fundamental como punto central de distribución, especialmente para las señales de control de los servos y las matrices LED, así como para las líneas de alimentación.

Para llevar a cabo estas conexiones de forma precisa, es indispensable conocer la disposición y función de cada uno de los 40 pines del puerto GPIO (General Purpose Input/Output) de la Raspberry Pi. La Figura 25 muestra el diagrama de pines (pinout) detallado de la Raspberry Pi 4 Modelo B, especificando qué pines corresponden a las interfaces utilizadas en este proyecto, como I2C, SPI, y las líneas de alimentación de 5V y 3.3V. Este diagrama es la guía de referencia principal para replicar el montaje electrónico del robot.



2.5 Grados de Libertad y Cinemática del Robot

La capacidad de movimiento del robot Avatiñ@ está definida por los grados de libertad (GdL) que le proporcionan sus actuadores.

2.5.1 Movimientos implementados

En su diseño actual, el robot Avatiñ@ cuenta con los siguientes grados de libertad principales:

2.5.1.1 Giro horizontal de la base (Pan)

El servomotor ubicado en la base permite un giro horizontal de la estructura superior del robot. Idealmente, este movimiento alcanza los 360°, aunque en la práctica podría estar limitado por la gestión del cable externo para evitar enredos. Este movimiento permite al robot orientar su "mirada" en cualquier dirección horizontal.

2.5.1.2 Giro vertical del cuello (Tilt)

El servomotor situado en el cuello permite la inclinación de la cabeza del robot hacia arriba y hacia abajo. El rango de este movimiento será de -30° a 30° siendo 0 la posición donde mira hacia delante. Este movimiento es crucial para dirigir la cámara y para expresar ciertas "emociones" o intenciones.

2.5.2 Discusión sobre limitaciones actuales

La cinemática actual, con dos grados de libertad principales (pan y tilt), permite una buena capacidad de orientación de la cabeza y, por ende, de la cámara y los "ojos". Sin embargo, esta configuración presenta ciertas limitaciones. Por ejemplo, para seguir un objeto que se mueve lateralmente mientras el robot mantiene una orientación general de su cuerpo, se requiere el giro completo de la base.

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE DEL ROBOT AVATIÑ@

3.1 Introducción al Software

Tras el detallado diseño y ensamblaje del hardware, el siguiente pilar fundamental del proyecto Avatiñ@ reside en el desarrollo de su software. Este capítulo se adentra en la arquitectura, los componentes y las metodologías empleadas para dar vida e inteligencia al robot, permitiendo no solo su operatividad básica, sino también una interacción rica y fluida con el usuario final. La concepción del software ha sido un viaje iterativo, buscando un equilibrio entre funcionalidad robusta, eficiencia y una experiencia de usuario intuitiva, especialmente considerando el contexto educativo al que se orienta Avatiñ@.

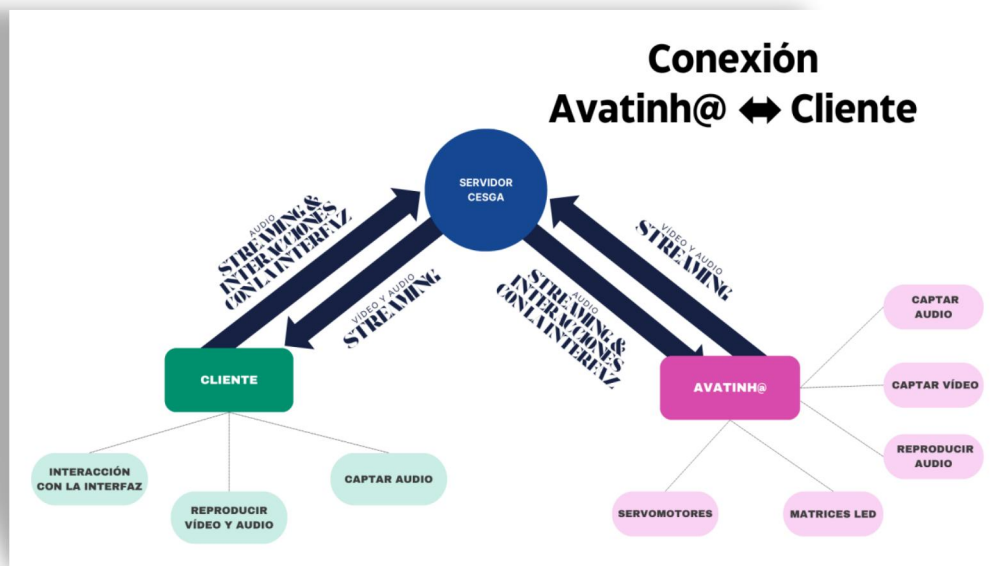


Figura 27: Conexión avatiñ@-cliente

La arquitectura del software se ha estructurado en tres grandes bloques interconectados:

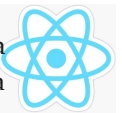
- **Software Interno del Robot:** Residente en la Raspberry Pi 4, este componente es el cerebro directo de Avatiñ@, encargado de gestionar todos los actuadores, sensores y la comunicación multimedia en tiempo real.
- **Servidor Web Intermediario:** Actúa como un puente crucial entre el robot y los usuarios, gestionando las conexiones, la lógica, el acceso a bases de datos y la transmisión de información.

- **Interfaz de Usuario (Frontend):** Es la ventana a través de la cual los distintos tipos de usuarios interactúan con el sistema Avatiñ@, desde la administración hasta el control directo del robot por parte del alumno.
-

La filosofía de desarrollo se ha centrado en la modularidad, para facilitar la escalabilidad y el mantenimiento; la eficiencia, optimizando recursos como el ancho de banda y la capacidad de procesamiento; y, sobre todo, en la experiencia de usuario, buscando que la interacción con Avatiñ@ sea lo más natural y sencilla posible.

Para materializar esta visión, se ha optado por un conjunto de tecnologías específicas, seleccionadas por su idoneidad para cada tarea:

- **Python:** Para el desarrollo del software interno en la Raspberry Pi, aprovechando su amplia gama de librerías para el control de hardware, procesamiento multimedia y networking.
- **Node.js:** Como entorno de ejecución para el backend del servidor, permitiendo construir una API RESTful eficiente y escalable en JavaScript.
- **React:** Para la creación de la interfaz de usuario, gracias a su flexibilidad, su ecosistema y la facilidad que ofrece para desarrollar interfaces dinámicas y reactivas, con la vista puesta en una futura Progressive Web App (PWA).
- **MariaDB:** Como sistema gestor de la base de datos relacional, por su robustez, fiabilidad y compatibilidad.



Este capítulo desgranará cada uno de estos componentes, explicando su diseño, implementación y los desafíos encontrados, así como las soluciones aplicadas para lograr un sistema cohesivo y funcional.

2.2 Software Interno del Robot Avatiñ@ (Raspberry Pi)

El corazón palpitante del robot Avatiñ@ reside en el software desarrollado específicamente para la Raspberry Pi 4 Modelo B. Este software es el responsable directo de interpretar las órdenes, gestionar los sensores y actuadores, y establecer la comunicación con el mundo exterior.

2.2.1 Sistema Operativo y Entorno

Como base para el software interno, se ha seleccionado Raspberry Pi OS (anteriormente conocido como Raspbian), una distribución de Linux optimizada para el hardware de la Raspberry Pi. Esta elección se fundamenta en su estabilidad, amplio soporte comunitario y la facilidad para configurar las interfaces de hardware necesarias.

El lenguaje de programación principal para todo el desarrollo en la Raspberry Pi ha sido Python. Su sintaxis clara, la vasta cantidad de librerías disponibles para el control de GPIOs, interfaces como I2C, SPI, y módulos de cámara, así como su capacidad para manejar tareas de red, lo convierten en una opción idónea para este proyecto.

La configuración inicial del sistema operativo incluye la habilitación de:

- Conectividad Wi-Fi: Esencial para la comunicación con el servidor web.
- Interfaz CSI (Camera Serial Interface): Para la conexión directa del módulo de cámara.

- Puertos GPIO (General Purpose Input/Output): Para el control de los servomotores y las matrices LED.

2.2.2 Control de Actuadores

Una parte significativa del software interno se dedica al control preciso de los actuadores que dotan de movimiento y expresividad al robot.

2.2.2.1 Control de Servomotores FS5103R

Los movimientos de giro horizontal (pan) de la base y vertical (tilt) del cuello se realizan mediante dos servomotores FS5103R de rotación continua. Si bien estos motores son económicos, presentan ciertos desafíos:

- **Control a Bajas Velocidades:** La librería estándar para el control de estos servos a menudo presenta un funcionamiento errático o a tirones a velocidades bajas o medias-bajas, que son cruciales para movimientos suaves y naturales. Tras una fase de investigación y experimentación considerable, se ha desarrollado una lógica de control en Python que modula los pulsos PWM (Pulse Width Modulation) de una manera no convencional, logrando un movimiento hiperfluido en todo el rango de velocidades deseadas. Esto fue un pequeño gran logro, ya que la diferencia en la percepción del movimiento del robot es notable.
- **Calibración por Software:** Estos motores no vienen con una calibración de fábrica precisa y carecen de memoria de posición. Aunque disponen de un potenciómetro físico para un ajuste manual, este proceso es engorroso y poco práctico para el usuario final. Por ello, se ha implementado un proceso de calibración guiado por software. Antes del primer uso, o cuando sea necesario, el usuario, a través de la interfaz, puede ajustar los pulsos para encontrar el punto neutro de cada motor (el pulso en el que el motor permanece quieto). Además, se realiza una calibración de la velocidad angular, que permite al software estimar la posición angular del motor en todo momento, basándose en el tiempo y la velocidad aplicada. Esta estimación es fundamental para establecer límites de giro y evitar daños mecánicos o el enrollamiento de cables.

2.2.2.2 Control de Matrices LED 8x8

Para dotar al robot de una forma de expresión visual, se utilizan dos matrices LED de 8x8, una para cada "ojo". La programación de estas matrices, aunque conceptualmente sencilla, requiere una gestión cuidadosa:

- Cada uno de los 64 LEDs por matriz se controla individualmente (encendido o apagado, un 1 o un 0).
- Se han preprogramado una serie de expresiones o "emojis" (por ejemplo, ojos contentos, tristes, curiosos, etc.), que consisten en patrones de LEDs específicos.
- La creación de animaciones, aunque no es el foco principal en esta etapa, es posible mediante la secuenciación rápida de diferentes patrones, una tarea que se simplifica gracias a las librerías de Python que abstraen la comunicación SPI con los controladores MAX7219.



Figura 28: Matriz led corazón



Figura 29: Matriz led confundido



Figura 31: Matriz led enfadado



Figura 30: Matriz led feliz

- ◆ [Vídeo animado del corazón](#)
- ◆ [Vídeo movimiento sin emociones](#)

2.2.3 Captura y Tansmisión Multimedia

Una de las funcionalidades clave de Avatiñ@ es su capacidad de telepresencia, lo que implica la transmisión de audio y vídeo en tiempo real.

- **Flujo Multimedia:** Se establece un flujo de vídeo unidireccional desde la cámara del robot hacia la interfaz del usuario y un flujo de audio bidireccional entre el robot y el usuario.
- **Tecnologías:** Para esta tarea, se ha optado por WebRTC (Web Real-Time Communication), un estándar abierto que permite la comunicación en tiempo real directamente entre navegadores y dispositivos, y WebSockets para la señalización y el intercambio de mensajes de control.
- **Optimización de Latencia:** Uno de los mayores retos fue la latencia en la transmisión.
- Inicialmente, se experimentaban retrasos de 1 a 2 segundos, lo cual es inaceptable para una interacción fluida. Mediante una ardua labor de programación, prueba y error, optimizando buffers, la configuración de los códecs y la gestión de los paquetes de red, se ha conseguido reducir esta latencia a un rango de 100-200 milisegundos, lo que proporciona una experiencia mucho más inmediata y natural.
- **Gestión del Eco en Audio Bidireccional:** El audio bidireccional continuo presenta el clásico problema del eco: el micrófono del robot capta el audio que emiten sus propios altavoces (la voz del usuario remoto). Para solucionar esto, se implementó una lógica de "half-duplex" inteligente: mientras el alumno habla a través de la interfaz, el micrófono del robot se silencia temporalmente (o su ganancia se reduce drásticamente). Cuando el alumno deja de hablar, el micrófono del robot se reactiva para captar el sonido del entorno. Esta solución, aunque simple, ha demostrado ser muy efectiva para eliminar el eco y permitir conversaciones claras y ordenadas.

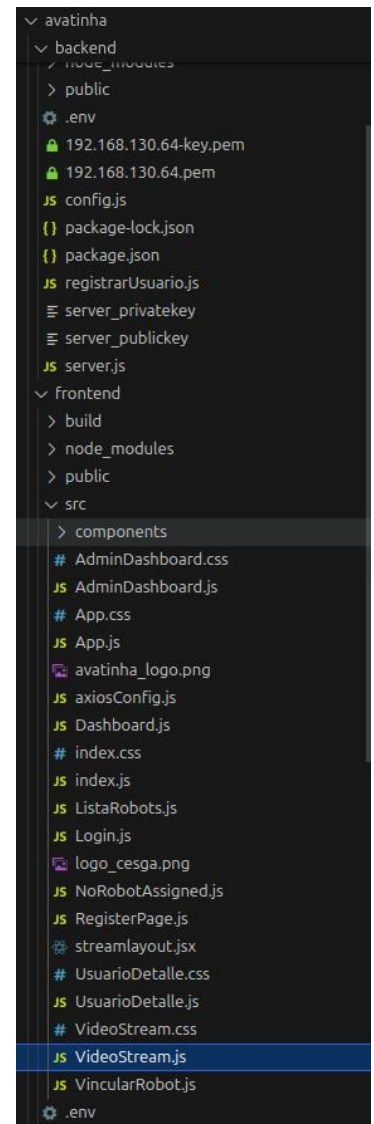


Figura 32: Servidor

2.2.4 Comunicación con el Servidor

El software interno del robot debe comunicarse constantemente con el servidor web para recibir comandos y enviar información de estado.

- **Recepción de Comandos:** El robot expone internamente ciertos puntos (endpoints locales y funciones que son llamadas por un script que escucha al servidor) para recibir comandos del servidor. Estos comandos incluyen toda la comunicación e interacción del usuario con la interfaz, así como son los movimientos deseados para los motores de pan y tilt.
- **Interpretación de Comandos del Joystick:** La interfaz de usuario presenta un joystick virtual. Cuando el usuario mueve este joystick, la interfaz envía las coordenadas (por ejemplo, a 45º con una cierta magnitud) al servidor, y este las retransmite al robot. El software del robot utiliza fórmulas matemáticas para descomponer este vector de movimiento en las velocidades angulares apropiadas para cada uno de los dos motores, permitiendo un control intuitivo y simultáneo de ambos ejes.
- **Proceso de Vinculación:** Para conectar un nuevo robot Avatiñ@ al sistema, se ha diseñado un proceso sencillo pero seguro. Al ejecutar un script específico en la terminal de la Raspberry Pi (una vez conectada a una red Wi-Fi), el robot se comunica con el servidor y genera un código numérico único de 6 cifras. Este código está internamente asociado al número de serie de la Raspberry Pi. Este código es el que el profesor utilizará en la interfaz web para vincular el robot a su cuenta. Es un sistema similar al que usan plataformas como Netflix para vincular nuevos dispositivos.

Aunque la descripción de estas funcionalidades pueda parecer concisa, su implementación ha supuesto un esfuerzo de programación considerable, con muchas horas dedicadas a la depuración y optimización para asegurar un funcionamiento robusto y fiable.

2.3 Servidor Web Intermediario

El servidor web es una pieza central en la arquitectura de Avatiñ@, actuando como el nexo de unión entre los robots físicos y las interfaces de usuario. Su desarrollo se ha realizado con Node.js y se encuentra alojado en las instalaciones del CESGA (Centro de Supercomputación de Galicia), entidad colaboradora en este proyecto.

2.3.1 Arquitectura y Tecnologías del Backend

El backend del servidor se ha construido como una API RESTful utilizando Node.js, junto con frameworks populares como Express.js para facilitar el enrutamiento y la gestión de peticiones HTTPS. Esta elección permite un desarrollo rápido y eficiente en JavaScript, manteniendo una coherencia tecnológica con el frontend (React).

La API define una serie de endpoints que permiten realizar todas las acciones necesarias:

- Autenticación y gestión de sesiones de usuario.
- Registro y gestión de colegios/profesores y alumnos.
- Vinculación y gestión de robots.
- Transmisión de comandos de control a los robots.
- Intermediación en la señalización WebRTC y los mensajes WebSocket.

El alojamiento en la infraestructura del CESGA proporciona un entorno robusto, seguro y con la capacidad de red necesaria para soportar las comunicaciones en tiempo real del proyecto.

2.3.2 Gestión de la Comunicación

Una de las funciones primordiales del servidor es la gestión eficiente de la comunicación:

- **Intermediario Robot-Cliente:** El servidor no solo retransmite comandos, sino que también gestiona la lógica de conexión, asegurando que el robot correcto se comunique con el usuario correcto.
- **Señalización WebRTC/WebSockets:** Facilita el "handshake" inicial para las conexiones WebRTC, intercambiando información de sesión (SDP) y candidatos ICE entre el robot y el cliente. Los WebSockets se utilizan para una comunicación persistente de bajo nivel para comandos y estado que no requieren el ancho de banda de WebRTC.
- **Interacción con la Base de Datos:** Todas las operaciones que requieren persistencia de datos (registros de usuarios, vinculación de robots, etc.) pasan por el servidor, que interactúa con la base de datos MariaDB.

2.3.3 Base de Datos

Se ha optado por MariaDB como sistema gestor de bases de datos relacional. Se han definido principalmente dos bases de datos (o conjuntos de tablas lógicamente separadas):

- **UsuariosDB:** Almacena la información relativa a los usuarios del sistema.
 - ◆ Campos típicos: id_usuario, nombre_usuario, nivel_acceso (1 para admin, 2 para profesor/colegio, 3 para alumno), id_registrador (para saber qué admin registró a qué profesor, o qué profesor a qué alumno), fecha_registro, etc. La parte más importante, la contraseña de cada usuario.
 - ◆ Esta estructura permite organizar a los usuarios y filtrar la información relevante para cada nivel (por ejemplo, un profesor solo ve los alumnos que ha registrado).
 - ◆ Seguridad: La seguridad de las contraseñas es primordial. En ningún caso se almacenan contraseñas en texto plano. Se utiliza un algoritmo de hashing robusto (bcrypt) para generar un hash encriptado de la contraseña. Cuando un usuario intenta iniciar sesión, la contraseña proporcionada se hashea de la misma manera y se compara con el hash almacenado. Esta es una medida de seguridad estándar e irrenunciable para proteger la privacidad de los usuarios. Ni siquiera el administrador del sistema puede conocer las contraseñas reales.
- **VinculacionDB:** Contiene los avatinh@s vinculados, y saber a que profesor/alumno pertenecen, o si no están vinculados a nadie todavía.
 - ◆ Campos: Número de identificación de 6 cifras, nombre asignado por el profesor, profesor vinculado y alumno vinculado.

La correcta estructuración y gestión de estas bases de datos es vital para el funcionamiento ordenado y seguro de toda la plataforma Avatiñ@.

2.4 Interfaz de Usuario (Frontend)

La interfaz de usuario es el punto de contacto directo con los administradores, profesores y, lo más importante, los alumnos. Se ha desarrollado como una aplicación web utilizando el framework React.

2.4.1 Tecnología y Diseño

La elección de React se basa en su capacidad para crear interfaces de usuario interactivas y reutilizables mediante un paradigma basado en componentes. Esto no solo agiliza el desarrollo, sino que también facilita el mantenimiento y la escalabilidad futura. Una consideración importante ha sido la posibilidad de evolucionar la aplicación web hacia una Progressive Web App (PWA). Una PWA puede ofrecer una experiencia muy similar a una aplicación nativa (instalable en el dispositivo, notificaciones push, acceso offline limitado) sin necesidad de pasar por tiendas de aplicaciones y, crucialmente para este proyecto, sin requerir acceso a funcionalidades nativas complejas del móvil, lo que simplifica enormemente su desarrollo y despliegue.

El diseño se ha enfocado en la claridad y la facilidad de uso, minimizando la complejidad visual y asegurando que las funciones principales sean accesibles de forma intuitiva para cada tipo de usuario. Se ha trabajado para que la interfaz sea responsiva, adaptándose a diferentes tamaños de pantalla, aunque el uso principal se prevé en tablets u ordenadores para una mejor experiencia de control.

2.4.2 Niveles de Acceso y Funcionalidades

La interfaz presenta diferentes vistas y funcionalidades según el nivel de acceso del usuario, aunque todos comparten un mismo punto de entrada (login):

2.4.2.1 Administrador

El rol de administrador tiene como principal función la gestión de las entidades educativas:

- **Registro de Colegios/Profesores:** Permite dar de alta nuevas instituciones o docentes en el sistema.
- **Listado y Gestión:** Visualización de los colegios/profesores registrados, con opciones para editar su información o gestionar su estado.
- *(Referencia a Imagen: Aquí se podría incluir una captura de pantalla del panel de administrador mostrando la lista de colegios).*

2.4.2.2 Profesor/Centro

Este nivel de usuario gestiona los robots y los alumnos a su cargo:

- **Vinculación de Robots Avatiñ@:** Dispone de una sección para introducir el código de 6 cifras generado por un nuevo robot, asociándolo así a su cuenta.
- **Registro de Alumnos:** Permite crear cuentas para los alumnos. Para simplificar el proceso y proteger la privacidad, solo se requiere un nombre de usuario (ej. "Raul_Trillo"), sin necesidad de correos electrónicos, lo cual es más adecuado para niños que no suelen tener cuentas de email propias.
- **Asignación de Robots:** Una vez registrados los alumnos y vinculados los robots, el profesor puede asignar un robot específico a un alumno para su uso.
- **Listados:** Visualización de los robots vinculados a su centro y de los alumnos registrados, con opciones para gestionar sus asignaciones.

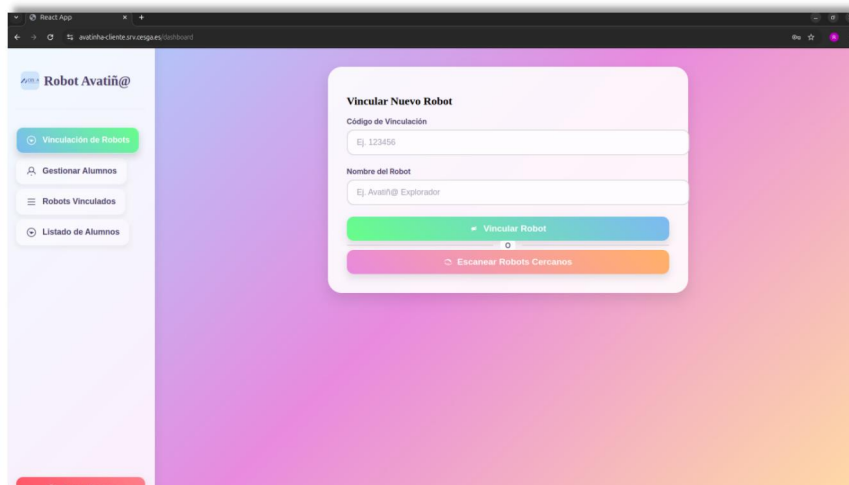


Figura 33: Interfaz profesor

2.4.2.3 Alumno

La interfaz del alumno es la más enfocada en la interacción directa con el robot:

- **Proceso de Primer Registro/Configuración de Contraseña:** Cuando un profesor crea un usuario para un alumno, este accede por primera vez a la plataforma. Introduce el nombre de usuario proporcionado por el profesor. El sistema verifica su existencia en la base de datos. Si es correcto, se le permite al alumno establecer su propia contraseña. Este paso es crucial para que el alumno sienta el control y la privacidad de su cuenta.
- **Inicio de Sesión:** Una vez registrada la contraseña, el alumno puede iniciar sesión con su nombre de usuario y la contraseña que ha establecido.
- **Interfaz de Control del Robot:** Tras el inicio de sesión, el alumno es dirigido directamente a la pantalla de control del robot. Esta pantalla está diseñada para ser inmersiva y fácil de usar:
 - ◆ **Visualización del Vídeo:** El flujo de vídeo de la cámara del robot ocupa la mayor parte de la pantalla. La porción de pantalla que ocupa puede variar ligeramente dependiendo de la resolución y el aspect ratio de la cámara utilizada en el robot.
 - ◆ **Joystick Dinámico:** Para controlar el movimiento del robot (cabeza y cuerpo), se implementa un joystick virtual. Este joystick no es visible permanentemente, sino que aparece en el punto de la pantalla donde el alumno toca con el dedo o hace clic con el ratón. Al soltar, el joystick desaparece. Esto ofrece una gran flexibilidad y permite al

alumno controlar el robot tocando en cualquier parte del área de vídeo, sin tener que buscar un control fijo.

- ◆ **Selección de Emojis:** En una zona accesible de la pantalla (una barra lateral), se muestran los 7 emojis preprogramados. Al tocar uno de ellos, se envía la orden al robot para que muestre la expresión correspondiente en sus matrices LED.
- ◆ **Controles Adicionales:** En la parte inferior de la pantalla (o en una posición similarmente accesible) se encuentran varios iconos de control:
 - ✧ Micrófono (on/off): Para activar o desactivar la transmisión de su voz al robot
 - ✧ Altavoz (on/off): Para silenciar o activar el sonido proveniente del robot.
 - ✧ Zoom de Cámara (+/-): Permite realizar un zoom digital sobre la imagen de la cámara del robot (si la cámara y el software del robot lo soportan).
 - ✧ Volumen de Salida (+/-): Controla el volumen del audio que se reproduce en los altavoces del robot, permitiendo al alumno ajustar si quiere hablar en voz baja con un compañero cercano o más alto para que le escuche el profesor a distancia.

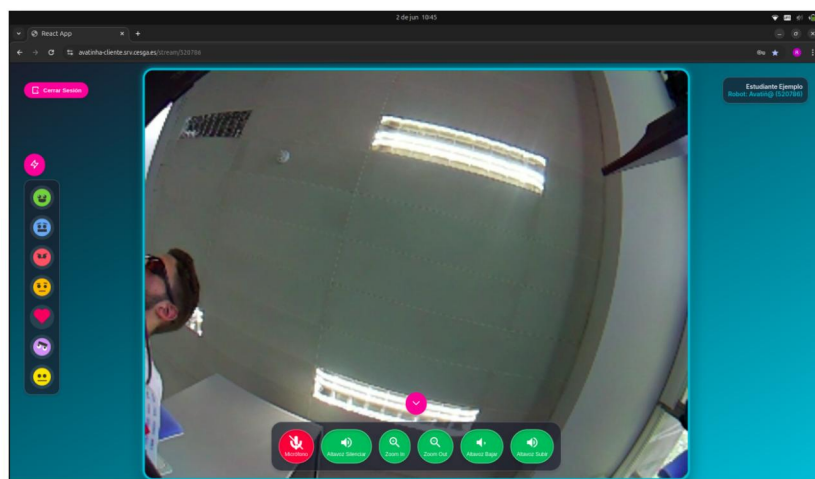


Figura 34: Interfaz alumno

2.4.3 Experiencia de Usuario

Se ha puesto especial énfasis en que el flujo de interacción sea coherente:

- **Login Unificado:** A pesar de los tres niveles de acceso, todos los usuarios utilizan la misma página de inicio de sesión.
- **Redirección Automática:** Una vez que el usuario inicia sesión, el sistema lo redirige automáticamente a la interfaz y funcionalidades correspondientes a su nivel de acceso.
- **Interfaz Intuitiva:** Especialmente para el alumno, se ha buscado que los controles sean obvios y que la interacción con el robot sea lo más directa y divertida posible.
- **Interfaz amigable:** Se ha buscado en todo momento una interfaz, mezclando los colores, los emojis y las pequeñas animaciones, que sean lo más amigables posible para la interacción de niños con ella.

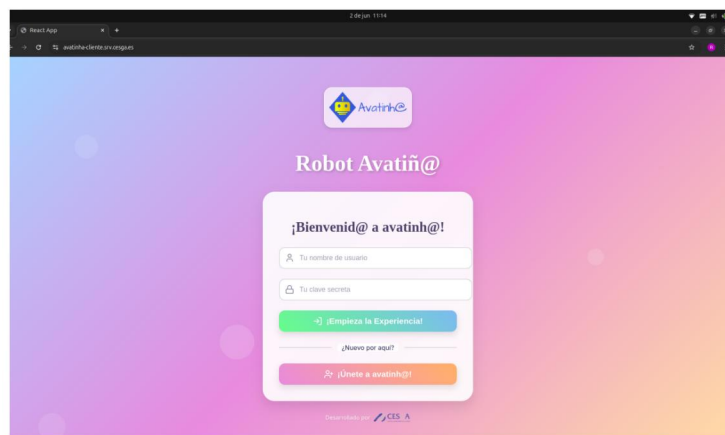


Figura 35: Login

Este [último vídeo](#) muestra una navegación e iteración con la web, en todos sus puntos.

2.5 Desafíos Significativos y Soluciones Implementadas

El desarrollo del software de Avatiñ@ no ha estado exento de desafíos. Algunos de los más notables y las soluciones aplicadas incluyen:

- **Latencia en Streaming Multimedia:** Como se mencionó, la reducción de la latencia en la transmisión de vídeo y audio mediante WebRTC/WebSockets fue un reto considerable, superado mediante la optimización exhaustiva de la configuración de red y el procesamiento de datos. Lograr esos 100-200ms fue un punto de inflexión.
- **Control Preciso de Servomotores Económicos:** Conseguir un movimiento fluido y controlable con los servos FS5103R, especialmente a bajas velocidades, requirió salirse de las implementaciones estándar y desarrollar una lógica de control PWM a medida.
- **Eliminación de Eco en Audio Bidireccional:** La solución de silenciar selectivamente el micrófono del robot mientras el usuario remoto habla, aunque simple, resolvió eficazmente un problema que de otro modo haría la comunicación muy difícil.
- **Sistema de Vinculación Seguro y Sencillo:** Crear un método para que los profesores puedan añadir nuevos robots al sistema que sea a la vez fácil de usar y razonablemente seguro era importante. El sistema de código de 6 cifras cumple bien esta función.
- **Gestión de Múltiples Niveles de Usuario:** Diseñar una arquitectura de software y base de datos que soporte diferentes roles y permisos de manera clara y segura, utilizando una única interfaz de login, requirió una planificación cuidadosa.

CAPÍTULO 4

MEJORAS FUTURAS Y LÍNEAS DE TRABAJO

El desarrollo del robot Avatiñ@ hasta su estado actual ha sentado unas bases funcionales robustas y ha cumplido con los objetivos primordiales de crear una plataforma de telepresencia social open source, asequible y personalizable. Sin embargo, la propia naturaleza de un proyecto de estas características invita a una continua evolución y a la exploración de nuevas funcionalidades que puedan enriquecer aún más la experiencia del usuario y ampliar las capacidades del robot. Este capítulo se dedica a delinear diversas propuestas de mejora y futuras líneas de trabajo, tanto a nivel de hardware como de software e interacción, que podrían llevar a Avatiñ@ al siguiente nivel.

3.1 Mejoras en el Hardware y la Cinemática

Si bien el diseño actual del hardware es funcional y eficiente, existen áreas donde futuras iteraciones podrían introducir mejoras significativas.

3.1.1 Evolución de la Cinemática del Robot

Como se mencionó brevemente en el capítulo de hardware, la cinemática actual del robot, con dos grados de libertad principales (giro de base y giro vertical del cuello), es efectiva, pero podría enriquecerse para dotar al robot de movimientos más naturales y expresivos.

Añadir un Tercer Eje de Movimiento en el Cuello (Yaw):

- **Descripción:** Consistiría en incorporar un servomotor adicional en la unión del cuello con la cabeza. Este servo permitiría un movimiento de giro horizontal de la cabeza (conocido como "yaw") independiente del giro de la base. Este movimiento podría tener un rango limitado, por ejemplo, de -30° a $+30^\circ$, lo cual sería suficiente para la mayoría de las interacciones sutiles.
- **Ventajas:** Esta adición transformaría la capacidad expresiva del robot. Permitiría realizar movimientos más naturales y humanos, como asentir o negar con la cabeza de forma más convincente, o seguir objetos y personas con la mirada sin necesidad de girar toda la base para pequeños desplazamientos angulares. Combinando este nuevo eje con el giro de la base (pan) y la inclinación del cuello (tilt), se podría lograr una cinemática de tres ejes para la cabeza, mejorando significativamente la interacción y la percepción de "vida" y atención del robot.

- **Implicaciones:** Esta mejora requeriría una revisión del diseño mecánico del cuello y la cabeza para alojar el nuevo servomotor y asegurar un movimiento fluido. También implicaría la adición de un canal de control para este servo en el software y, posiblemente, ajustes en el controlador de servos si se utiliza uno externo. La lógica de control del movimiento en el software del robot también necesitaría ser actualizada para gestionar este tercer eje.

3.1.2 Optimización de Componentes y Durabilidad

- **Batería y Sistema de Alimentación:** Investigar baterías con mayor densidad energética para prolongar la autonomía del robot o explorar sistemas de carga más eficientes, como la carga inalámbrica o bases de carga por contacto, para simplificar el proceso.
- **Robustez de Piezas Impresas:** Aunque el PLA es adecuado para prototipado, explorar materiales de impresión 3D más resistentes y duraderos (como PETG o ASA) para las piezas estructurales clave, especialmente si el robot va a tener un uso intensivo. También se podría refinar el diseño de los enganches para aumentar su resistencia al desmontaje frecuente.

3.2 Mejoras en el Software y la Interacción

El software es el alma del robot y ofrece un vasto campo para futuras mejoras que pueden potenciar la personalización, la usabilidad y la inmersión.

3.2.1 Autonomía de Red y Conectividad

Integración de Conectividad Móvil (Tarjeta SIM 4G/5G):

- **Descripción:** Una de las mejoras más impactantes sería dotar al robot de su propia conexión a internet mediante una tarjeta SIM y un módem 4G (o incluso 5G en el futuro).
- **Ventajas:** Esto eliminaría la dependencia de redes Wi-Fi externas, que no siempre están disponibles, son estables o fáciles de configurar en todos los entornos escolares o domésticos. El robot ganaría una autonomía operativa total, pudiendo conectarse desde prácticamente cualquier lugar con cobertura móvil.
- **Implicaciones:** Requeriría la selección de un módulo hat o USB compatible con la Raspberry Pi, la gestión de la conexión de datos a nivel de software y consideraciones sobre el consumo energético adicional del módem.

3.2.2 Personalización Avanzada de la Expresividad del Robot

Sistema de Creación de Emojis y Animaciones Personalizadas (Estilo Scratch):

- **Descripción:** Desarrollar una herramienta web integrada en la plataforma del profesor/centro que permita crear expresiones y animaciones para las matrices LED de Avatiñ@ de forma visual e intuitiva, similar a la programación por bloques de Scratch.
- **Funcionamiento:** Los usuarios podrían arrastrar y soltar bloques para definir qué LEDs se encienden o apagan, crear secuencias de patrones para formar animaciones (parpadeos, guiños,

ojos que se mueven) y asociar estas creaciones a nuevos "botones de emoción" en la interfaz del alumno. Se podrían definir parámetros como la duración de cada fotograma de la animación, repeticiones, etc.

- **Ventajas:** Esto llevaría la personalización a un nivel completamente nuevo. Cada alumno podría tener un conjunto de expresiones únicas para su Avatiñ@, o los profesores podrían diseñar "packs de emociones" temáticos para diferentes actividades o para adaptarse mejor a las necesidades emocionales de cada niño. Fomentaría la creatividad y la apropiación del robot por parte del alumno.
- **Implicaciones:** Requeriría un desarrollo significativo en el frontend (la interfaz de creación) y en el backend (para almacenar y gestionar estas personalizaciones). El software del robot también necesitaría ser capaz de descargar e interpretar estos nuevos patrones de LED dinámicamente.

3.2.3 Experiencia de Usuario Inmersiva

Integración con Gafas de Realidad Virtual (VR) para Control y Visualización:

- **Descripción:** Permitir que el alumno controle y experimente la clase a través de unas gafas de Realidad Virtual. El vídeo de la cámara del robot se transmitiría directamente a las gafas VR, ofreciendo una vista en primera persona.
- **Control por Movimiento de Cabeza:** El head tracking (seguimiento de la cabeza) de las gafas VR se utilizaría para controlar directamente los motores del cuello (y de la base, si es necesario) del robot Avatiñ@. Si el alumno gira la cabeza hacia la izquierda en el mundo real, la cabeza del robot giraría hacia la izquierda en el aula, proporcionando una forma de exploración del entorno sumamente intuitiva y natural.
- **Ventajas:** Esto ofrecería un nivel de inmersión sin precedentes. El alumno se sentiría mucho más "presente" en el aula, pudiendo mirar a su alrededor de forma natural, como si realmente estuviera allí. Podría mejorar significativamente la atención y la sensación de conexión.
- **Implicaciones:** Esta es una mejora ambiciosa. Requeriría:
 - ◆ Compatibilidad con APIs de VR (por ejemplo, WebXR para soluciones basadas en navegador).
 - ◆ Una transmisión de vídeo de muy baja latencia y alta calidad optimizada para VR.
 - ◆ Un sistema robusto para traducir los datos de orientación del casco VR en comandos precisos para los servomotores del robot, gestionando la calibración y los límites de movimiento.
 - ◆ Consideraciones sobre el mareo por movimiento (motion sickness) y la ergonomía para el alumno.

3.2.4 Evolución de la Plataforma Web

Transformación a Progressive Web App (PWA):

- **Descripción:** Evolucionar la actual aplicación web de React hacia una PWA.
- **Ventajas:** Mejoraría la accesibilidad y la experiencia de usuario, permitiendo "instalar" la aplicación en dispositivos móviles y de escritorio, recibir notificaciones push (por ejemplo, cuando la clase está a punto de empezar o si hay un mensaje del profesor) y, potencialmente, ofrecer algunas funcionalidades offline básicas.
- **Implicaciones:** Implica la implementación de Service Workers, un Web App Manifest y asegurar que la aplicación cumple con los criterios de una PWA.

3.2.5 Inteligencia y Automatización (Opcional y a Largo Plazo)

Integración de Funcionalidades Básicas de Inteligencia Artificial (IA):

- **Reconocimiento de Voz Simple:** Para comandos locales básicos como "Avatiñ@, mira a la pizarra" o "Avatiñ@, saluda", sin necesidad de que el alumno interactúe siempre con la interfaz táctil.
- **Detección de Objetos o Rostros Básica:** Podría permitir al robot, por ejemplo, orientarse automáticamente hacia el profesor cuando este habla, o "reconocer" a compañeros (aunque esto último tiene importantes implicaciones de privacidad que deberían ser cuidadosamente gestionadas).
- **Reconocimiento y transcripción de la voz del profesor:** Se podría entrenar una IA con el objetivo de reconocer la voz de un profesor, transcribiendo todo lo que él dice, escribiendo en la pantalla, por ejemplo, para alumnos con problemas auditivos, incluso utilizando esta información para entrenar una LLM como llama3 para ayudar a resolver sus dudas post clase.
- **Ventajas:** Podría hacer la interacción más fluida y reducir la carga cognitiva del alumno para controlar el robot.
- **Implicaciones:** Requeriría el uso de librerías de IA en la Raspberry Pi (como TensorFlow Lite) y un entrenamiento de modelos adecuado, además de un hardware capaz de soportarlo sin mermar el rendimiento de las funciones principales.

3.2.6 Mejoras en las Herramientas de Gestión

Panel de Administración y Profesor Enriquecido:

- Añadir herramientas de seguimiento del uso del robot (tiempo de conexión, interacciones más usadas).
- Funcionalidades para que los profesores puedan asociar material educativo o actividades específicas al uso de Avatiñ@.
- Un sistema de comunicación más directo entre profesor y alumno a través de la plataforma.

3.3 Fomento de la Comunidad y Ecosistema Open Source

- **Creación de un Repositorio Público y Documentación Detallada:** Aunque el proyecto es open source, formalizar un repositorio público bien organizado (en plataformas como GitHub o GitLab) con toda la documentación, archivos de diseño, código fuente y guías de montaje es crucial para fomentar la adopción y contribución por parte de la comunidad.
- **Foro o Comunidad de Usuarios y Desarrolladores:** Establecer un espacio (un foro, un servidor de Discord, etc.) donde los usuarios puedan compartir experiencias, resolver dudas, proponer mejoras y colaborar en el desarrollo de nuevas funcionalidades o personalizaciones.

Estas líneas de desarrollo futuro, desde mejoras incrementales hasta la incorporación de tecnologías más avanzadas como la VR, demuestran el vasto potencial de Avatiñ@. La naturaleza abierta del proyecto es, en sí misma, una invitación a la innovación continua, con la esperanza de que la comunidad pueda abrazar y expandir esta iniciativa para beneficiar a un número cada vez mayor de estudiantes.

CAPÍTULO 5

IMPACTO SOCIOEMOCIONAL Y EDUCATIVO DE AVATIÑ@ EN EL ALUMNADO AUSENTE

Como se ha esbozado en la introducción de este trabajo, la ausencia prolongada del entorno escolar representa un desafío multidimensional para el alumnado infantil. Más allá de la interrupción del aprendizaje formal, esta situación puede desencadenar una cascada de efectos negativos en el bienestar emocional, el desarrollo social y la motivación del niño. El proyecto Avatiñ@, al ofrecer una presencia virtual activa y participativa en el aula, se erige como una herramienta con un notable potencial para mitigar estas consecuencias adversas y fomentar un desarrollo más saludable e inclusivo. Este capítulo profundiza en los beneficios específicos que un robot de telepresencia como Avatiñ@ puede aportar al alumnado que se enfrenta a largos periodos fuera de la escuela.

4.1 Mitigación del Aislamiento Social y Fortalecimiento del Bienestar Emocional

Uno de los impactos más inmediatos y perjudiciales de la ausencia escolar prolongada es el aislamiento social. La desconexión del grupo de iguales y del entorno escolar puede llevar a sentimientos de soledad, tristeza, ansiedad e incluso depresión en los niños. Avatiñ@ ataca directamente este problema al:

- **Mantener la Conexión Social:** El robot permite al alumno no solo ver y oír lo que sucede en clase, sino también interactuar activamente con sus compañeros y profesores. Puede participar en conversaciones, responder preguntas, unirse a las bromas y, en definitiva, seguir formando parte de la vida social del aula. Esta continuidad es crucial para que el niño no se sienta olvidado o excluido.
- **Reducir la Sensación de Soledad:** La presencia constante, aunque sea virtual, de sus compañeros y el ambiente escolar ayuda a combatir la soledad que a menudo acompaña a las enfermedades o situaciones que obligan al aislamiento. El robot se convierte en sus "ojos y oídos" en un lugar donde desearía estar.
- **Mejorar el Estado de Ánimo y la Autoestima:** Al sentirse conectado, participe y capaz de interactuar, el estado de ánimo del niño tiende a mejorar. La posibilidad de seguir

contribuyendo y siendo reconocido por sus compañeros y profesores puede reforzar su autoestima, que a menudo se ve mermada por la enfermedad o la ausencia.

- **Fomentar un Sentido de Normalidad y Pertenencia:** Avatiñ@ ayuda a normalizar la situación del alumno ausente. Aunque no esté físicamente presente, su "avatar" sí lo está, lo que le permite mantener un rol activo dentro del grupo y reforzar su sentido de pertenencia a la comunidad escolar.

Estudios sobre el uso de robots de telepresencia, como el AV1, han mostrado que estos dispositivos pueden incrementar la asistencia escolar (virtual), mejorar el compromiso, la autoestima, las relaciones con el personal y los compañeros, y el bienestar emocional general de los alumnos.

4.2 Continuidad en el Desarrollo Social y Facilitación de la Reintegración

La etapa escolar es fundamental para el desarrollo de habilidades sociales. La ausencia prolongada puede interrumpir este proceso y dificultar la reintegración posterior.

- **Práctica Continua de Habilidades Sociales:** A través de Avatiñ@, el alumno sigue practicando habilidades de comunicación, cooperación y resolución de conflictos en un entorno social real, aunque sea mediado por la tecnología.
- **Mantenimiento de Vínculos de Amistad:** Las amistades son un pilar fundamental en la infancia. El robot facilita que estos lazos no se debiliten por la distancia física, permitiendo interacciones cotidianas que nutren las relaciones.
- **Suavizar la Reintegración Escolar:** Uno de los beneficios más significativos es la facilitación del proceso de reintegración cuando el alumno puede volver físicamente al aula. Al no haber perdido el contacto ni la familiaridad con sus compañeros y la dinámica de clase, el "shock" del regreso se minimiza. El alumno no se siente como un extraño, sino como alguien que, en cierto modo, nunca se fue del todo. Esto puede reducir drásticamente la ansiedad asociada al retorno y acelerar la readaptación.

4.3 Apoyo a la Continuidad Académica y al Compromiso con el Aprendizaje

Si bien el componente social y emocional es primordial, Avatiñ@ también juega un papel importante en el ámbito académico:

- **Acceso a la Instrucción en Tiempo Real:** El alumno puede seguir las explicaciones del profesor, ver la pizarra (gracias a la movilidad de la cámara del robot) y participar en las actividades académicas como si estuviera en su pupitre.
- **Mantenimiento del Compromiso:** La capacidad de interactuar y participar activamente mantiene al alumno más comprometido con el proceso de aprendizaje que si dependiera únicamente de recibir apuntes o clases grabadas.
- **Reducción del Desfase Curricular:** Aunque la telepresencia no sustituye completamente la experiencia presencial, ayuda a minimizar la pérdida de contenido académico y a mantenerse al día con el ritmo de la clase, lo que es crucial para evitar un desfase que luego sea difícil de recuperar.

- **Estimulación Cognitiva:** La participación activa en el entorno de aprendizaje, la necesidad de procesar información visual y auditiva en tiempo real, y la interacción social contribuyen a mantener la estimulación cognitiva del alumno.

4.4 Empoderamiento y Autonomía del Alumno

El diseño de Avatiñ@, con su interfaz de control intuitiva, busca empoderar al alumno:

- **Sensación de Control y Agencia:** El alumno no es un receptor pasivo de información, sino que controla activamente su "presencia" en el aula. Decide hacia dónde mirar, con quién interactuar y cuándo hablar. Esta sensación de control es psicológicamente importante, especialmente para niños que pueden sentirse impotentes debido a su situación de salud.
- **Fomento de la Independencia:** La capacidad de manejar el robot de forma autónoma le otorga un grado de independencia en su participación escolar, reduciendo la necesidad de que otros actúen constantemente como intermediarios.

4.5 El Valor Añadido de la Filosofía Open Source y la Personalización de Avatiñ@

Más allá de los beneficios inherentes a la telepresencia, el enfoque open source y la alta personalización de Avatiñ@ aportan ventajas adicionales significativas:

- **Adaptabilidad a Necesidades Individuales:** La posibilidad de modificar el aspecto físico del robot (mediante la impresión 3D de diferentes carcasas o cabezas) y sus expresiones (con el futuro sistema de creación de emojis) permite adaptar el robot a la personalidad y preferencias de cada niño. Esto puede aumentar la aceptación del robot por parte del propio alumno y de sus compañeros, haciéndolo sentir más "suyo" y menos como un dispositivo médico o tecnológico impersonal.
- **Accesibilidad Económica:** El bajo coste de Avatiñ@ (entre 100 y 200 euros) en comparación con soluciones comerciales (varios miles de euros) democratiza el acceso a esta tecnología, permitiendo que más niños, escuelas y familias puedan beneficiarse de ella, independientemente de sus recursos económicos. Esto tiene un impacto social directo al promover la equidad en el acceso a herramientas de apoyo educativo.
- **Implicación de la Comunidad Educativa:** Al ser un proyecto abierto, se fomenta la implicación de profesores, padres e incluso los propios alumnos en la mejora y adaptación del robot, creando un sentido de comunidad y propiedad compartida en torno a la herramienta.

En conclusión, Avatiñ@ no es simplemente un dispositivo tecnológico, sino un puente que busca reconectar al niño ausente con su mundo escolar, abordando sus necesidades emocionales, sociales y educativas de una manera integral. Al reducir el aislamiento, facilitar la interacción social, mantener el compromiso académico y ofrecer una plataforma personalizable y accesible, este proyecto tiene el potencial de mejorar significativamente la calidad de vida y las perspectivas de desarrollo de los niños que deben pasar largos periodos alejados de sus aulas, ayudándoles a sentir que, a pesar de la distancia, siguen siendo una parte valiosa y activa de su comunidad.

CAPÍTULO 6

EVOLUCIÓN DEL PROYECTO Y PROCESO DE DISEÑO ITERATIVO

El desarrollo del robot Avatiñ@ no ha sido un proceso lineal, sino un camino iterativo de diseño, prototipado, prueba y mejora continua. En este capítulo, describo la evolución del proyecto desde su concepción inicial hasta el estado actual, destacando los desafíos encontrados y las soluciones implementadas.

5.1 Punto de Partida y Primeras Iteraciones

El presente TFG comenzó con una idea básica: realizar un robot con unas funcionalidades y una estética definidas. Sin embargo, en la fase inicial, carecía de un plan concreto sobre cómo implementar las conexiones electrónicas o los detalles mecánicos más finos. El proceso de diseño de los enganches y las uniones internas, por ejemplo, pasó por varias versiones conceptuales antes de llegar a una solución viable.

Paralelamente al modelado del chasis, comencé a desarrollar las primeras versiones del software, centradas en aspectos fundamentales como los controladores de los servomotores y la comunicación básica con la Raspberry Pi.

5.2 Primer Prototipo Físico y Pruebas Iniciales

Una vez que tuve un primer diseño del exterior, aunque todavía sin las "jaulas" internas ni otras complejidades, me centré en el componente más crítico: el rodamiento de la base. Este elemento fue probado y rediseñado múltiples veces hasta alcanzar un nivel de optimalidad que garantizara un movimiento de giro muy fluido.

Para validar estos primeros diseños mecánicos, imprimí un prototipo a una escala del 50%, como se puede apreciar en la Figura X.X. Este modelo a escala, aunque no era completamente funcional, me permitió verificar tolerancias y la viabilidad general de los encajes. Como en esta fase el robot todavía no tenía una cabeza diseñada, le añadimos unos ojos simbólicos que nos acompañaron durante todo el proceso de desarrollo.

Además comenzamos a diseñar también el [software del movimiento](#).



Figura 36: Prototipo enganche cabeza

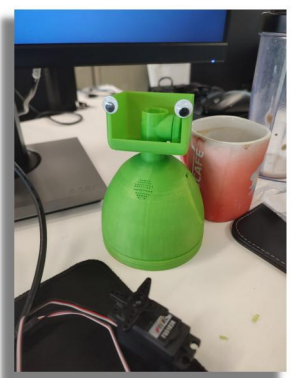


Figura 37: Prototipo avatiñ@ en miniatura

5.3 Desarrollo del Prototipo a Escala Real y Desafíos Mecánicos

El siguiente paso consistió en imprimir el chasis a tamaño real (100%) y abordar el desafío de la robustez. Durante esta etapa exploré diversas ideas mediante un proceso continuo de prueba y error. Por ejemplo, consideré la posibilidad de añadir un tornillo pasante que atravesara el cuello y la cabeza para dar más rigidez, una idea que finalmente descarté por complicar el ensamblaje y no aportar una mejora significativa.

La primera base que imprimí a tamaño real demostró que el mecanismo de giro funcionaba, pero carecía de la fluidez deseada. Los enganches entre el cuerpo principal y la base, aunque funcionales, presentaban una ligera holgura que no me satisfacía y que identifiqué como un punto clave a corregir.

[Vídeo prueba cabeza](#)

[Vídeo prueba base](#)

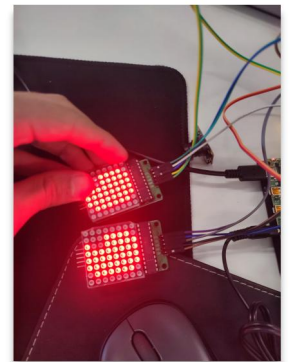
[Vídeo prueba rodamiento](#)

5.4 Refinamiento de los Movimientos y Actuadores

Con el prototipo a escala real en mano, me centré en perfeccionar los movimientos vertical y horizontal:

- [Movimiento Vertical](#): Debo destacar que el mecanismo de inclinación del cuello funcionó a la perfección desde la primera prueba, validando el diseño del rodamiento sin cilindros.
- [Movimiento Horizontal](#): En la base, aunque el giro era posible, me encontré con un problema de diseño inesperado. Los enganches de la pieza superior al servo estaban situados muy cerca del eje del motor. Debido al principio de la palanca, esta proximidad impedía que el motor ejerciera la fuerza necesaria para mover la estructura con soltura.

Figura 37: Ojos led



5.5 Integración de Componentes Electrónicos y Pruebas Funcionales

Una vez corregido el problema de diseño en la base, procedí a instalar las matrices de LED en la cabeza para formar los "ojos" del robot. Realicé las primeras pruebas de animaciones y, sinceramente, quedé muy contento con la capacidad expresiva y la fluidez de los resultados obtenidos.

5.6 Hacia el Prototipo Funcional Completo

En la fase final del desarrollo del hardware, ensamblé la versión definitiva de la cabeza y solucioné por completo los [problemas de fluidez de la base y de falta de fuerza del motor](#). Como se puede observar en el Vídeo, la velocidad y suavidad del giro en esta última versión son espectaculares.

Paralelamente, logré implementar y probar con éxito el sistema de vídeo y sonido bidireccional, una de las funcionalidades clave del proyecto, cuyo funcionamiento se demuestra en el [Vídeo](#).

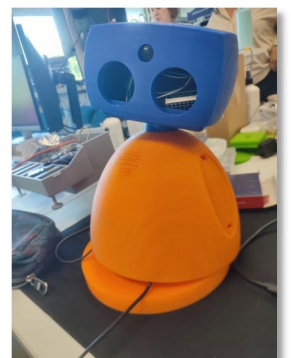


Figura 38: Prototipo avatiñ@

Finalmente, el [último Vídeo](#) muestra el movimiento integrado de la cabeza y la base en el último modelo, evidenciando un funcionamiento fluido y preciso, resultado de todo el proceso de refinamiento iterativo.

5.7 Evolución de la Interfaz de Usuario Web

El desarrollo del software, en particular de la interfaz de control, también siguió un proceso evolutivo. La primera versión de la web era puramente funcional, pero estéticamente muy pobre. A través de varias iteraciones, y como se puede comparar en las Figuras Y.Y e Y.Z, la interfaz ha evolucionado hasta convertirse en la plataforma actual: una web con un diseño limpio, moderno y agradable, que facilita enormemente la interacción con el robot.

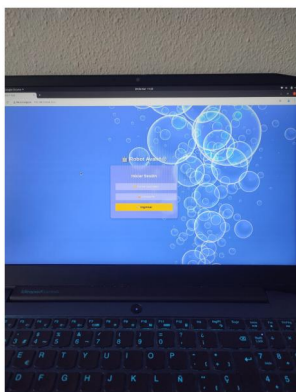


Figura 39: Primer login

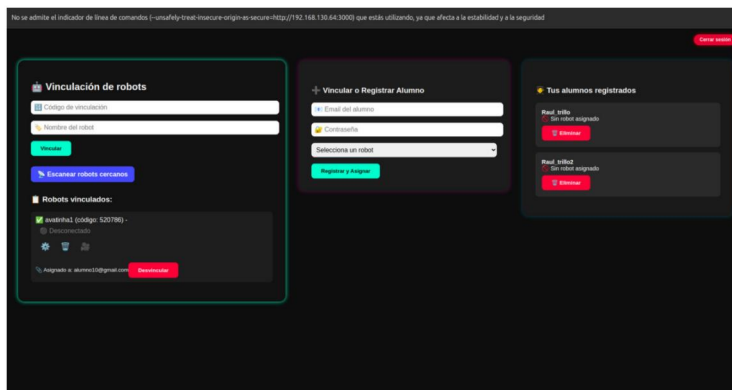


Figura 40: Primera página profesor

Ahora, por último, dejo [un vídeo](#) donde se puede ver el funcionamiento completo del robot, desde animaciones, audio, movimiento fluido, la interfaz web y todo lo que se puede hacer con avatiñ@.

CAPÍTULO 7

RESULTADOS Y CONCLUSIONES

Este capítulo final se dedica a la presentación y discusión de los resultados obtenidos tras el ciclo completo de diseño, desarrollo e implementación del robot de telepresencia social Avatiñ@. Asimismo, se exponen las conclusiones generales derivadas de este Trabajo de Fin de Grado, reflexionando sobre los logros alcanzados, las limitaciones identificadas y el vasto potencial que este proyecto abre para futuras iniciativas.

6.1 Presentación de Resultados

El desarrollo del proyecto Avatiñ@ ha culminado en la creación de un prototipo de robot de telepresencia social plenamente funcional y usable. Si bien es cierto que, como todo proyecto de esta envergadura, siempre existen márgenes de mejora y se requerirían algunas semanas adicionales de trabajo para refinar ciertos aspectos y, fundamentalmente, para llevar a cabo pruebas exhaustivas en entornos reales con niños, el estado actual del robot es notablemente avanzado. Se ha conseguido un sistema robusto que permite iniciar las primeras pruebas de concepto, especialmente en lo referente a la aceptación social del robot y la validación de sus funcionalidades básicas de interacción.

A continuación, se detallan los hitos más significativos alcanzados en las diferentes áreas del proyecto:

6.1.1 Hitos Clave en el Desarrollo del Hardware

El diseño y la fabricación del hardware constituyeron una fase primordial, donde se materializó la estructura física y funcional del robot. Los resultados en este ámbito han sido especialmente satisfactorios:

- **Un Chasis Innovador y Eficiente:** Se ha logrado un diseño de chasis excepcional, fabricado íntegramente mediante impresión 3D. Su principal innovación reside en un sistema de ensamblaje tipo puzzle/LEGO, que no requiere el uso de tornillos ni adhesivos, tal como se detalló en el Capítulo 1. Esto no solo simplifica enormemente el montaje, permitiendo que cualquier persona con conocimientos limitados pueda ensamblar el robot en un tiempo reducido (aproximadamente 20-30 minutos una vez impresas las piezas), sino que también facilita las tareas de mantenimiento y modificación.

- **Optimización del Espacio y Alojamiento Seguro de Componentes:** Cada componente electrónico (Raspberry Pi, baterías, protoboard, módulo de carga, etc.) cuenta con un alojamiento específico y seguro dentro del chasis (*jaulas*). Este diseño meticuloso asegura que ninguna pieza quede suelta, previniendo desconexiones accidentales y optimizando la distribución del peso para una correcta estabilidad.
- **Diseño Optimizado de Rodamientos:** Los sistemas de giro horizontal de la base y vertical del cuello se apoyan en rodamientos de gran diámetro diseñados a medida e integrados en la estructura impresa. El diseño del rodamiento de la base, con cilindros, ha sido fruto de un proceso de prueba y error hasta alcanzar una optimalidad notable en su suavidad de giro (360°) y bajo rozamiento, calificado como *hiperóptimo*. El rodamiento del cuello, aunque prescinde de los cilindros, también ofrece un movimiento fluido y preciso para la inclinación de la cabeza.
- **Modularidad y Personalización:** La concepción modular del chasis, especialmente la cabeza intercambiable, sienta las bases para una fácil personalización estética y funcional, un aspecto clave de la filosofía del proyecto.

6.1.2 Hitos Clave en el Desarrollo del Software y la Interacción

El software es el componente que dota de inteligencia y capacidad de interacción a Avatiñ@. Los avances en esta área han sido cruciales para superar algunos de los desafíos más significativos del proyecto:

- **Control de Movimiento Fluido y Preciso:** A pesar de utilizar servomotores económicos (FS5103R) que inicialmente presentaban dificultades para movimientos suaves a bajas velocidades, se ha desarrollado una lógica de control en Python que consigue un movimiento hiperfluido en todo el rango de operación. La implementación de un sistema de calibración por software para estos motores también ha sido un éxito.
- **Sistema de Comunicación Multimedia de Baja Latencia:** Uno de los mayores desafíos técnicos era lograr una transmisión de vídeo y audio bidireccional con un retraso mínimo. Mediante el uso de WebRTC y WebSockets, y tras un intenso trabajo de optimización, se ha conseguido reducir la latencia a un rango de 100-200 milisegundos. Este logro es fundamental para una interacción natural y efectiva, eliminando la frustración que pueden causar los retardos pronunciados.
- **Calidad de Audio Bidireccional:** Se ha implementado un sistema de audio bidireccional funcional, con una solución efectiva para la cancelación del eco (silenciando el micrófono del robot mientras el alumno habla), permitiendo conversaciones claras.
- **Interfaz de Usuario Intuitiva y Completa:** Se ha desarrollado una interfaz de usuario basada en React que, según la percepción del desarrollador, ha alcanzado un nivel de pulcritud y funcionalidad tal que prácticamente poco habría que mejorar en esta parte para la operativa actual. La interfaz contempla los diferentes niveles de usuario (administrador, profesor/centro, alumno) y ofrece controles intuitivos como el joystick dinámico y la selección de emojis.
- **Expresividad Visual a Través de Matrices LED:** Los "ojos" del robot, implementados con matrices LED 8x8, proporcionan una interacción visual muy buena con el entorno, permitiendo al robot mostrar diferentes emociones y estados, lo que enriquece la comunicación no verbal.
- **Plataforma Robusta de Gestión:** El backend desarrollado en Node.js, junto con la base de datos MariaDB, proporciona una plataforma sólida para la gestión de usuarios, robots y la comunicación general del sistema, alojada en las instalaciones del CESGA.

6.1.3 Adecuación del Prototipo para Pruebas Iniciales

Considerando los resultados obtenidos, el prototipo actual de Avatiñ@ se considera suficientemente maduro y estable para iniciar una fase de pruebas preliminares. Estas pruebas iniciales podrían centrarse en:

- **Aceptación Social:** Observar cómo interactúan los niños (tanto el usuario remoto como los compañeros de clase) con el robot, su diseño y sus funcionalidades.
- **Usabilidad de la Interfaz:** Validar la facilidad de uso de la interfaz de control por parte de los alumnos y de las interfaces de gestión por parte de profesores.
- **Rendimiento en Entornos Reales:** Evaluar el comportamiento del robot en un aula real, considerando aspectos como la conectividad de red, la calidad del audio y vídeo en un entorno con ruido ambiente, y la duración de la batería.

Aunque, como se detalla en el Capítulo 3 (Mejoras Futuras y Líneas de Trabajo), existen diversas vías para enriquecer Avatiñ@, el estado actual ya constituye una base sólida para validar los conceptos fundamentales del proyecto.

6.2 Conclusiones Generales

Al finalizar este Trabajo de Fin de Grado, es posible afirmar con satisfacción que se han alcanzado y, en muchos aspectos, superado los objetivos inicialmente propuestos. El proyecto Avatiñ@ no solo ha culminado en un prototipo funcional, sino que ha sentado las bases de una plataforma con un inmenso potencial transformador.

- **Éxito en la Creación de una Alternativa Viable:** El principal objetivo era desarrollar un robot de telepresencia social open source, de bajo coste y altamente personalizable, como alternativa a soluciones comerciales costosas y cerradas como el AV1. Este objetivo se ha cumplido plenamente. Avatiñ@ demuestra que es posible crear una herramienta sofisticada y efectiva con recursos accesibles y una filosofía abierta.
- **Democratización de la Tecnología de Telepresencia:** Con un coste estimado de materiales entre 100 y 200 euros, Avatiñ@ tiene el potencial de hacer accesible la tecnología de telepresencia a un número mucho mayor de niños, familias e instituciones educativas que no podrían permitirse las soluciones existentes. Esta democratización es, quizás, la contribución más significativa del proyecto.
- **Innovación en Diseño y Funcionalidad:** Se han aportado soluciones innovadoras en el diseño del chasis (ensamblaje sin tornillos, rodamientos integrados), en el control de actuadores (movimiento hiperfluido de servos económicos) y en la optimización de la comunicación multimedia (baja latencia). Estos logros técnicos son valiosos en sí mismos y pueden servir de referencia para otros proyectos open source.
- **Una Base Sólida para el Futuro:** El estado actual del proyecto representa un éxito rotundo y constituye una base MUY amplia y robusta sobre la cual construir futuras investigaciones y desarrollos. Las líneas de trabajo identificadas en el Capítulo 3 (conectividad 4G, personalización avanzada de emojis, integración VR, etc.) son solo algunas de las muchas posibilidades que se abren.
- **Impacto Social Potencial:** Más allá de los aspectos técnicos, el verdadero valor de Avatiñ@ reside en su potencial para mejorar la calidad de vida de niños que atraviesan situaciones difíciles de ausencia escolar. Al facilitar la conexión social, emocional y académica, este robot puede ser una herramienta crucial para su bienestar y desarrollo integral, extendiendo su

impacto positivo también a sus familias, aliviando parte de la carga y la preocupación asociadas a estas situaciones.

- **Colaboración y Continuidad:** La colaboración con entidades como el CESGA, y el interés que este tipo de iniciativas pueda despertar en otras instituciones, son fundamentales para asegurar la continuidad y la evolución del proyecto. Avatiñ@ está listo para ser el punto de partida de una línea de trabajo que puede aportar beneficios tangibles a la sociedad gallega y, potencialmente, a comunidades más amplias gracias a su naturaleza abierta.

Por último, este Trabajo de Fin de Grado ha demostrado la viabilidad de crear soluciones tecnológicas avanzadas con un enfoque social, abierto y colaborativo. Avatiñ@ no es solo un robot; es una manifestación de cómo la ingeniería y la tecnología pueden ponerse al servicio de la inclusión y el bienestar de los más vulnerables, ofreciendo esperanza y conexión en momentos de dificultad. El aprendizaje obtenido durante su desarrollo ha sido inmenso, y la satisfacción de haber creado una herramienta con tanto potencial es la mayor recompensa.

Finalmente, como un recurso visual complementario y muy significativo, me gustaría compartir un enlace. [Este último vídeo es muy importante para entender todo lo que llevamos hablando en este proyecto](#); se trata de un pequeño documental realizado por el canal público "La 1" centrado en la USC y en este proyecto y aunque centrado en el robot AV1, ilustra de manera efectiva la problemática y el impacto socioemocional discutido a lo largo de este trabajo. Permite visualizar el tipo de interacciones y beneficios que se buscan, ofreciendo un contexto claro sobre cómo Avatiñ@, en un futuro, podría sustituir y mejorar soluciones existentes.

CAPÍTULO 8

REFERENCIAS BIBLIOGRÁFICAS Y RECURSOS

8.1 Software, Frameworks y Entornos de Ejecución

- [1] FreeCAD (Versión 0.21). (2024). FreeCAD: Your own 3D parametric modeler. Disponible en: <https://www.freecad.org>
- [2] Python (Versión 3.9). Python Software Foundation. (2024). Python Language Reference. Disponible en: <https://www.python.org>
- [3] Node.js (Versión 18.x). OpenJS Foundation. (2024). Node.js. Disponible en: <https://nodejs.org/>
- [4] React (Versión 18). Meta Platforms, Inc. (2024). React – A JavaScript library for building user interfaces. Disponible en: <https://react.dev/>
- [5] MariaDB Server (Versión 10.6). MariaDB Corporation. (2024). MariaDB Server. Disponible en: <https://mariadb.org/>

8.1.1. Protocolos de Comunicación y Tecnologías Web

- [6] World Wide Web Consortium (W3C). (2021). WebRTC 1.0: Real-time Communication Between Browsers. W3C Recommendation. Esta especificación oficial ha sido el documento de referencia principal para la implementación de la comunicación de audio y vídeo en tiempo real. Disponible en: <https://www.w3.org/TR/webrtc/>
- [7] I. Grigorik, A. (2013). High Performance Browser Networking: Nuts & Bolts of Modern Web Performance. O'Reilly Media.
- [8] Johnston, A. B., & Burnett, D. C. (2016). WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web. Digital Codex.

8.1.2. Componentes Electrónicos y Hojas de Datos

[9] Raspberry Pi Foundation. (2024). Raspberry Pi 4 Model B - Technical Specifications. Disponible en: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>

[10] Maxim Integrated. (2019). MAX7219/MAX7221 - Serially Interfaced, 8-Digit LED Display Drivers Datasheet. Disponible en: <https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf>

8.1.3. Artículos y Proyectos de Inspiración

[11] No Isolation. (2024). AV1 - The telepresence robot for children with long-term illness. Disponible en: <https://www.noisolation.com/av1>

[12] Newhart, V. A., & Olson, J. S. (2017). "The social and academic benefits of a telepresence robot for a student with a long-term illness". International Journal of Social Robotics, 9(1), 53-66.

[13] Thingiverse. (2024). Thingiverse - Digital Designs for Physical Objects. Disponible en: <https://www.thingiverse.com>

8.1.4. Recursos Formativos y Asistentes de IA

[14] Herrera, F. (2024). React Native: Crea aplicaciones para iOS y Android [Curso Online]. Udemy. Recurso formativo consultado para afianzar conocimientos avanzados en el ecosistema de React, aplicables al desarrollo de la interfaz de usuario. Disponible en: <https://www.udemy.com/course/react-native-crea-aplicaciones-moviles-reales-ios-y-android/>

[15] OpenAI. (2024). ChatGPT. [Modelo de Lenguaje Grande].

[16] Google. (2024). Gemini. [Modelo de Lenguaje Grande].