



Universidad
Pontificia
de Salamanca

Servicio de Investigación
y Transferencia



UPSAINNOVA24.25

Proyectos del Club Universitario de Innovación

UPSA INNOVA
2024_2025

UPSA INNOVA

2024_2025

INVESTIGADORES Y ALUMNOS DE LOS PROYECTOS DEL CUI
2024_2025



Universidad Pontificia
de Salamanca

Dykinson, S.L.

No está permitida la reproducción total o parcial de este libro, ni su incorporación a un sistema informático, ni su transmisión en cualquier forma o por cualquier medio, sea éste electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y por escrito del editor. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (art. 270 y siguientes del Código Penal).

Diríjase a Cedro (Centro Español de Derechos Reprográficos) si necesita fotocopiar o escanear algún fragmento de esta obra. Puede contactar con Cedro a través de la web www.conlicencia.com o por teléfono en el 917021970/932720407

Este libro ha sido sometido a evaluación por parte de nuestro Consejo Editorial
Para mayor información, véase www.dykinson.com/quienes_somos

©Copyright by Universidad Pontificia de Salamanca
Madrid, 2026

Editorial DYKINSON, S.L.
Meléndez Valdés, 61 - 28015 Madrid
Teléfono (+34) 915442846 - (+34) 915442869
e-mail: info@dykinson.com
<http://www.dykinson.es>
<http://www.dykinson.com>

ISBN: 979-13-7047-215-3

Preimpresión:
New Garamond Diseño y Maquetación, S.L.

ÍNDICE

LEARNHACK: INNOVACIÓN TECNOLÓGICA PARA UN EDUCACIÓN INCLUSIVA Y ACCESIBLE..... 15

Lucía de la Vega Encinas, Udane Murgia Aramendi, Hugo de Castro Roncero, Sofía Alejandra García Arrojo y Lucía Martín Gómez

RESUMEN.....	15
1. CONTEXTUALIZACIÓN DEL PROYECTO	16
2. JUSTIFICACIÓN Y OBJETIVOS.....	16
2.1. Justificación del proyecto.....	16
2.2. Objetivos del proyecto	17
2.3. Diseño, desarrollo e implementación:	18
3. CONCLUSIONES Y REFLEXIONES FINALES DE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO.....	19
BIBLIOGRAFÍA	20
ANEXOS.....	20

ASIS: ASISTENTE VIRTUAL INTERACTIVO SIMPLIFICADO 23

Leyre García San Cayetano, Gabriel Mateos Ruíz, Hugo Sampedro Hernández y Rubén Martín García

RESUMEN.....	23
1. CONTEXTUALIZACIÓN DEL PROYECTO	24
2. JUSTIFICACIÓN Y OBJETIVOS.....	24
2.1. Justificación	24
2.2. Objetivos Generales	25
2.3. Objetivos Específicos	25
3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN	25
3.1. Diseño Conceptual.....	25
3.2. Diseño Técnico	26
3.3. Desarrollo	26
3.4. Implementación y Validación	27

4. CONCLUSIONES Y REFLEXIONES FINALES SOBRE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO	27
BIBLIOGRAFÍA	28
ANEXOS.....	28
SISTEMA DE INDUMENTARIA Y EQUIPAMIENTO PARA LA ACTIVIDAD FÍSICA EN LA NATURALEZA <SIENAT>	31
<i>Sergio López García, Lucía Romero Lobo, Lucía Suárez Collado, Pablo Elipe Lorenzo, Marcos Talavera Escribano, Carmen del Arco Chás y Pelayo Diez Fernández</i>	
RESUMEN.....	31
1. CONTEXTUALIZACIÓN DEL PROYECTO	32
2. JUSTIFICACIÓN Y OBJETIVOS.....	33
3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN	34
3.1. Diseño de la solución	34
3.2. Desarrollo tecnológico.....	34
3.3. Implementación.....	37
4. CONCLUSIONES Y REFLEXIONES FINALES SOBRE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO	37
BIBLIOGRAFÍA	38
COILAPP	41
<i>Carlos Caño Gómez, Alberto Pedrero Esteban, Ana Lugones Hoya</i>	
RESUMEN.....	41
1. CONTEXTUALIZACIÓN DEL PROYECTO	41
2. JUSTIFICACIÓN Y OBJETIVOS.....	42
3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN	43
3.1. Frontend	44
3.2. Backend.....	45
3.3. Base de datos	46
4. CONCLUSIONES Y REFLEXIONES FINALES DE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO	46
BIBLIOGRAFÍA	47

SMARTSKIN: INTELIGENCIA ARTIFICIAL APLICADA A LA DERMATOLOGÍA..... 49

Andrea Pérez Galán, Claudia Martín Villoria y Montserrat Mateos Sánchez

RESUMEN.....	49
1. CONTEXTUALIZACIÓN DEL PROYECTO	49
2. JUSTIFICACIÓN Y OBJETIVOS.....	50
3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN	51
4. CONCLUSIONES Y REFLEXIONES FINALES SOBRE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO	53
BIBLIOGRAFÍA	54
ANEXOS.....	55

DIVA: DINERO VISUAL Y ACCESIBLE 57

Amparo Casado Melo, María José Madrid Martín, Alfredo Sánchez-Fuentes Rivero, Ricardo Trujillo Antiveros, Irene Calvo Crespo, Mar Serrano Deyá e Iria Vargas Pérez

RESUMEN	57
1. CONTEXTUALIZACIÓN DEL PROYECTO	58
2. JUSTIFICACIÓN Y OBJETIVOS.....	60
3. OBJETIVOS.....	60
3.1. Objetivos específicos.....	61
4. DISEÑO, DESARROLLO E IMPLEMENTACIÓN	61
5. CONCLUSIONES	65
BIBLIOGRAFÍA	66

PSYCHE ASISTENTE VIRTUAL PARA EL PSICÓLOGO BASADO EN INTELIGENCIA ARTIFICIAL 69

Diego Tabernero Montero, Esteban Ristori Fernández, Carlos Martín Velázquez, Miguel Gutiérrez Hernández y Manuel Martín-Merino Acera

RESUMEN.....	69
1. CONTEXTUALIZACIÓN DEL PROYECTO	69
2. JUSTIFICACIÓN Y OBJETIVOS.....	71
2.1. Justificación	71
2.2. Objetivos.....	71

3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN	72
3.1. Arquitectura del sistema.....	72
4. IMPLEMENTACIÓN TECNOLÓGICA.....	73
4.1. Sistema de transcripción de audio	73
4.2. Sistema de inteligencia artificial	73
5. FUNCIONALIDADES IMPLEMENTADAS.....	74
5.1. Generación de diarios de sesión.....	74
5.2. Análisis cronológico.....	75
5.3. Gestión de archivos y seguridad.....	75
6. CONCLUSIONES Y REFLEXIONES FINALES DE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO	76
6.1. Principales logros alcanzados	76
6.2. Limitaciones identificadas.....	76
6.3. Reflexiones sobre la transferencia del conocimiento.....	77
6.4. Potencial de escalabilidad y continuidad	77
7. IMPACTO ESPERADO.....	77
BIBLIOGRAFÍA	78
TURNOFÁCIL. CHATBOT INTELIGENTE PARA OPTIMIZACIÓN DE RESERVAS Y CITAS	79
<i>Ana María Fermoso García, Juan Ramón Martín San Román y Paula Isabel Juanes Gil</i>	
RESUMEN.....	79
1. CONTEXTUALIZACIÓN DEL PROYECTO	80
2. JUSTIFICACIÓN Y OBJETIVOS.....	81
2.1. Objetivos.....	82
3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN	83
3.1. Arquitectura general del sistema.....	83
3.1.1. Capa de interacción con el usuario	83
3.1.2. Capa de orquestación (Flask backend).....	83

3.1.3. *Capa de inteligencia artificial*..... 84

3.1.4. *Capa de datos*..... 84

3.1.5. *Capa de disponibilidad* 85

3.2. Flujo de trabajo del sistema..... 86

3.3. Desarrollo y tecnologías empleadas..... 86

4. RESULTADOS Y VALIDACIÓN DEL SISTEMA..... 87

5. CONCLUSIONES Y REFLEXIONES FINALES SOBRE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO 89

BIBLIOGRAFÍA Y REFERENCIAS ELECTRÓNICAS..... 89

SISTEMA DE NUTRICIÓN AVANZADO Y PERSONALIZADO (SNAP)..... 91

Alejandro López Barrantes, Roberto Berjón Gallina y María Encarnación Beato Gutiérrez

RESUMEN..... 91

1. CONTEXTUALIZACIÓN DEL PROYECTO 91

2. JUSTIFICACIÓN Y OBJETIVOS..... 92

3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN 93

4. CONCLUSIONES Y REFLEXIONES FINALES 94

BIBLIOGRAFÍA 95

SMART TRACKING FRAMEWORK FOR IOT – GATEWAI 97

Roberto Berjón Gallinas, María Encarnación Beato Gutiérrez, Montserrat Mateos Sánchez y Ana María Feroso García

RESUMEN..... 97

1. CONTEXTUALIZACIÓN 98

2. DISEÑO E IMPLEMENTACIÓN..... 100

2.1. Definición de las funcionalidades de la aplicación o task pool..... 100

2.2. Recopilación de la información de tareas 104

2.3. Definición del API REST del gateway a partir de su especificación OpenAPI..... 105

2.3.1. *Parámetros simples* 107

2.3.2. *Parámetros complejos*:..... 108

2.4. Implementación del API Rest	109
2.5. Ejecución de peticiones que implican secuenciación de tareas y agregación de datos	112
3. CONCLUSIÓN	121
REFERENCIAS BIBLIOGRÁFICAS.....	122
SISTEMA PARA LA GESTIÓN EFICIENTE DE RECURSOS MATERIALES DEPORTIVOS (SISGER)	123
<i>Pelayo Díez Fernández, Esther Sánchez Moro, Pablo Elipe Lorenzo, David Revesado Carballares y Sergio López García</i>	
RESUMEN	123
1. CONTEXTUALIZACIÓN DEL PROYECTO	124
2. JUSTIFICACIÓN Y OBJETIVOS	125
3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN	126
4. CONCLUSIONES Y REFLEXIONES FINALES DE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO	131
BIBLIOGRAFÍA	131
DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE APERTURA DE PUERTAS AUTOMÁTICO BASADO EN VISIÓN ARTIFICIAL Y BIOMETRÍA FACIAL Y OCULAR (VABRO)	133
<i>Alfonso José López Rivero, Vidal Alonso Secades, Jorge Zakour Dib, Manuel Martín-Merino Acera, Antonio Ferreras García y Jorge Chamorro Sánchez</i>	
RESUMEN.....	133
1. CONTEXTUALIZACIÓN DEL PROYECTO.....	134
2. JUSTIFICACIÓN Y OBJETIVOS	134
3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN	135
4. CONCLUSIONES Y REFLEXIONES FINALES SOBRE LA TRANSFERENCIA DEL CONOCIMIENTO	137
BIBLIOGRAFÍA	137
ANEXOS.....	138
PLATAFORMA INTELIGENTE DE EVALUACIÓN DINÁMICA Y ADAPTATIVA (IDEA)	139
<i>Rubén Martín García, Sergio Alonso Rollán, Samuel Marcos Pablos, Rebeca Cordero Gutiérrez y Lucía Martín Gómez</i>	

RESUMEN.....	139
1. CONTEXTUALIZACIÓN DEL PROYECTO	139
2. JUSTIFICACIÓN Y OBJETIVOS	140
3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN	141
3.1. Actividades técnicas	141
3.2. Actividades de diseño	141
3.3. Actividades funcionales y pedagógicas	142
3.4. Resultados obtenidos	142
4. CONCLUSIONES Y REFLEXIONES FINALES.....	143
BIBLIOGRAFÍA	144

LEARNHACK: INNOVACIÓN TECNOLÓGICA PARA UN EDUCACIÓN INCLUSIVA Y ACCESIBLE



LUCÍA DE LA VEGA ENCINAS

UDANE MURGIA ARAMENDI

HUGO DE CASTRO RONCERO

SOFÍA ALEJANDRA GARCÍA ARROJO

LUCÍA MARTÍN GÓMEZ

RESUMEN

Este capítulo aborda los desafíos de la inclusión educativa de estudiantes sordos y mudos y presenta LearnHack, una aplicación web basada en inteligencia artificial cuyo objetivo es eliminar las barreras de comunicación en el aula y en otros entornos. Según la OMS, para 2050 más de 700 millones de personas tendrán pérdida auditiva severa, y en España ya se contabiliza alrededor de un millón de personas con discapacidad auditiva, de las cuales unas 100.000 dependen de la lengua de signos. Sin embargo, los recursos actuales son dramáticamente insuficientes, lo cual genera desigualdad en el acceso a la educación y la participación social.

LearnHack se plantea como una solución absoluta que combina la traducción de lengua de signos a texto y voz, la conversión de texto a voz, subtítulos inteligentes en tiempo real y herramientas interactivas para el aula. Su diseño está orientado a la accesibilidad y la usabilidad, garantizando una experiencia intuitiva tanto para estudiantes como para profesores. A nivel técnico, se ha desarrollado un código utilizando HTML, CSS y Java e integrando modelos de visión por ordenador, procesando de esta manera el lenguaje natural y la síntesis de voz.

Los resultados de pruebas piloto nos demuestran altos niveles de precisión y satisfacción, así como una reducción significativa en la carga mental de usuarios. Sin embargo,

existen limitaciones relacionadas con la infraestructura tecnológica, la formación docente y la amplia diversidad lingüística. En conclusión, LearnHack constituye un paso pionero hacia una educación más equitativa, ofreciendo una herramienta escalable, adaptable y con potencial de impacto tanto en el ámbito nacional como internacional.

1. CONTEXTUALIZACIÓN DEL PROYECTO

Actualmente, según la Organización Mundial de la Salud (OMS, 2021), se estima que para 2050 más de 700 millones de personas en el mundo tendrán algún grado de pérdida auditiva severa. En España, alrededor de 1 millón de personas tienen una discapacidad auditiva, de las cuales unas 100.000 dependen principalmente de lo que es la lengua de signos (CNSE, 2024). Sin embargo, como señala la Universidad de Salamanca (SID-INICO, 2024) los intérpretes no son tan abundantes y los sistemas que deberían ser de apoyo resultan insuficientes, en muchos casos no cumplen las necesidades completamente, es por eso que muchos estudiantes siguen acudiendo a clase sin intérprete. Todo esto deja a miles de estudiantes y profesionales sin un acceso pleno a la comunicación. Es por esto por lo que nace LearnHack, una aplicación web que aprovecha la inteligencia artificial (IA) para eliminar estas barreras. A diferencia de las soluciones existentes que tienden a centrarse únicamente en la transcripción de voz a texto o en un intento de traducción muy básico de lengua de signos, LearnHack se convierte en una aplicación de transcripción. Se integra múltiples funciones como, por ejemplo: traducción de lengua de signos a texto y voz; conversión de texto a voz, dando así voz a personas con discapacidad del habla; subtítulos inteligentes y adaptativos en tiempo real y herramientas didácticas que pueden integrarse en los planes de estudio. Pese a que el enfoque principal está en el entorno educativo, la intención futura es que la flexibilidad de este proyecto puede llegar a permitir también su aplicación no solo en las aulas sino también entornos tanto como profesionales y sociales. Con LearnHack no solo buscamos lo que puede entenderse cómo facilitar la comunicación, sino también fortalecer a los usuarios, garantizando que la discapacidad auditiva no sea un obstáculo para el aprendizaje y se pueda apostar una educación sin barreras para todos. Tal y como subraya la UNESCO (2020), la inclusión educativa es clave para garantizar el cumplimiento de los Objetivos de Desarrollo Sostenible.

2. JUSTIFICACIÓN Y OBJETIVOS

2.1. Justificación del proyecto

Asegurar la igualdad entre los alumnos con y sin discapacidad sigue siendo una dificultad que no se ha resuelto en el entorno educativo actual, ya que la accesibilidad a la tecnología aún está por alcanzarse. A pesar de que hay políticas internacionales,

como la Convención sobre los Derechos de las Personas con Discapacidad (CDPD) (UNESCO, 2020), que fomentan la inclusión, lo cierto es que numerosas instituciones no cuentan con instrumentos eficaces para derribar las barreras comunicativas. La educación inclusiva no se sostiene solamente en la voluntad de los docentes o el empeño individual del alumnado, sino también en la disponibilidad de soluciones que satisfagan necesidades particulares de forma eficaz y sostenible.

La falta de recursos adecuados genera situaciones de desigualdad: estudiantes sordos que no pueden seguir el ritmo de la clase, personas con discapacidad del habla que no logran participar en debates o exposiciones y profesores que carecen de el conocimiento o de apoyos tecnológicos para atender a esta diversidad en las aulas. Los muros de comunicación no solo afectan al rendimiento académico, sino también a la autoestima, la integración social y las oportunidades de futuro de quienes las sufren. El proyecto LearnHack se defiende como una respuesta a esta necesidad aún sin cubrir, completamente desprotegida. Su aporte se centra en ofrecer una herramienta completa que no se limite a un único tipo de funcionalidad, sino que combine varias tecnologías con un mismo fin: eliminar las barreras de comunicación y apoyar una sociedad igualitaria. Además, LearnHack se plantea como una solución escalable y adaptable, lo cual va a permitir responder no solo a la diversidad lingüística (los diferentes tipos de lenguas de signos) sino también a los distintos niveles educativos. Es así que este proyecto no solo va a atender a un problema específico, sino que abrirá la posibilidad de generar un cambio en la manera en que se concibe la accesibilidad en un aula.

2.2. Objetivos del proyecto

El objetivo general de LearnHack es ofrecer una solución tecnológica innovadora que vaya a garantizar a inclusión real de los estudiantes tanto sordos como mudos dentro de el sistema educativo, asegurando, de esta manera, que puedan acceder al aprendizaje, participar en actividades y comunicarse en igualdad de condiciones con el resto de la comunidad académica.

Es por ello que LearnHack busca desarrollar un sistema de comunicación en tiempo real que combine la traducción de lengua de signos, la generación automática de subtítulos y la conversión de texto a voz, con la idea de facilitar tanto la comprensión del contenido como la expresión oral de los estudiantes con discapacidad auditiva o del habla. Otro de los propósitos fundamentales es proporcionar a los usuarios una herramienta que fomente su autonomía, reduciendo la dependencia de intérpretes humanos y eliminando las limitaciones que suelen impedir la plena integración en el aula. Asimismo, el proyecto pretende validar su impacto en entornos educativos reales, midiendo su eficacia a través de indicadores como la accesibilidad alcanzada, la satisfacción de docentes y alumnos además de la mejora en los resultados de aprendizaje. También se plantea el desarrollo de una interfaz sencilla, intuitiva y accesible, de manera que pueda ser utilizada sin necesidad de formación técnica previa y que se adapte a diferentes niveles educativos.

Por último, LearnHack aspira a convertirse en una solución escalable, con plataformas digitales de enseñanza y en un futuro adaptable a distintas lenguas de signos, lo que abre la puerta a su implementación en diferentes países y contextos de educación. Con ello, el proyecto contribuye a avanzar hacia una educación más equitativa, en línea con unos compromisos internacionales y con Objetivos de Desarrollo Sostenible, especialmente en ODS, centrado en garantizar una educación que tiene calidad inclusiva para todo el mundo.

2.3. Diseño, desarrollo e implementación:

El diseño de LearnHack se ha creado bajo un enfoque centrado en la accesibilidad y la usabilidad, con el fin de garantizar que cualquier persona, independientemente de sus capacidades, pueda utilizar la aplicación de manera intuitiva. La selección cromática responde a criterios de accesibilidad visual recogidos en pautas internacionales como las WCAG: se optó por una paleta de blanco, morado y azul oscuro que asegura contraste suficiente y facilita la comprensión lectora, además de mantener coherencia estética con la identidad del proyecto (Figura 1).

La arquitectura de la aplicación continua un proceso claro que permite al usuario interactuar sin dificultades. En la pantalla inicial se muestran el logotipo y las opciones de registro e inicio de sesión (Figura 2). Una vez validado el acceso, el sistema da la bienvenida al usuario y despliega un menú con las principales funcionalidades. Lo cual puede resumirse en tres fases: entrada, procesamiento y salida. En la fase de entrada, la aplicación capta señales mediante voz (micrófono), texto o lengua de signos (cámara). Durante el procesamiento, se aplican algoritmos de inteligencia artificial que reconocen patrones visuales o fonéticos y los interpretan dentro de su contexto. Finalmente, en la salida, el usuario recibe la información en formato de texto, voz digital o representaciones visuales de signos, según la necesidad específica.

Las funcionalidades implementadas responden directamente a las barreras de comunicación que experimentan los usuarios. Entre ellas se encuentran la transcripción en tiempo real de voz a texto y signos animados, la traducción de lengua de signos a texto o voz tanto en transmisión en directo como en vídeos previamente grabados, y la lectura de texto en voz alta, especialmente útil para personas con dificultades en el habla. A nivel educativo, se añadieron herramientas interactivas como un chat de apoyo y un botón de “levantar la mano”, lo que refuerza la participación del alumnado en clase.

En cuanto a la implementación técnica, LearnHack se desarrolló en Visual Studio Code utilizando HTML, CSS y Java como lenguajes principales. El HTML define la estructura y organización del contenido, el CSS controla la presentación visual, y Java gestiona la lógica funcional e integra los modelos de inteligencia artificial para reconocimiento de voz, interpretación de signos y síntesis de voz. Esta arquitectura permite que la aplicación es ampliable, adaptable a diferentes dispositivos y navegadores y fácilmente integrable en plataformas de gestión del aprendizaje (LMS) como

Moodle o Google Classroom, lo que garantiza su funcionamiento en contextos educativos de distinta naturaleza.

Además, el diseño, desarrollo e implementación de LearnHack trata de combinar criterios de accesibilidad visual, una arquitectura técnica clara y una comprobación estricta exhaustiva, lo que refuerza su potencial como herramienta de inclusión educativa tanto en el ámbito nacional como internacional.

3. CONCLUSIONES Y REFLEXIONES FINALES DE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO

Como bien hemos ido señalando durante este capítulo las implementaciones de las soluciones tecnológicas inclusivas como LearnHack abren un mundo prometedor en la educación de estudiantes sordos y mudos. Sin embargo, también se plantea unos cuantos límites que condicionan lo que es la transferencia del conocimiento.

Al principio, continúa habiendo impedimentos tecnológicos, así como también de las infraestructuras. La eficacia de los sistemas de traducción a tiempo real depende principalmente de la estabilidad de las conexiones con internet, la disponibilidad de los dispositivos perfectamente equipados (principalmente con una cámara de buena calidad) y también la capacidad de los colegios, institutos... para poder actualizar los dispositivos y la tecnología correspondiente. Esto puede también suponer una desigualdad entre los propios centros escolares que dispongan de distintos niveles de recursos.

También hay limitaciones educativas y sus adaptaciones. La introducción de una herramienta innovadora no garantiza al 100% un cambio radical en las maneras de dar clase o la propia cultura escolar. Es necesario incluir en esta nueva inclusión de la tecnología con nuevas maneras de enseñar, una amplia formación a los profesores y además ajustar diferentes métodos que puedan ayudar al uso de esta aplicación.

Consideramos también muy importante tener en mente las grandes limitaciones de la lengua y sus normas. Como bien se ha comentado ya a lo largo de la historia, la diversidad de lenguas de signos constituye un reto, como ya ha señalado la European Union of the Deaf en su informe de accesibilidad educativa (EUD, 2022) y es que la gran amplitud de diversidad que hay dentro de lo que es las lenguas de signos y su variabilidad en los múltiples usos provocan que haya problemas en lograr una traducción automática muy precisa. Esto limita de manera drástica la progresión de la solución para los posibles contextos tan diversos en las culturas y las geografías.

En conclusión, aunque las limitaciones señaladas condicionan la transferencia del conocimiento, no reducen el valor de LearnHack como iniciativa pionera. Más bien resaltan la importancia de acompañar la innovación tecnológica con políticas educativas, inversiones en infraestructura y un trabajo conjunto con la sociedad actual. Solo así será posible fortalecer soluciones que no solo funcionen en el laboratorio,

sino que transformen de manera efectiva la realidad educativa de miles de estudiantes que pertenecen a todas partes del mundo.

BIBLIOGRAFÍA

CNSE. Datos sobre personas con discapacidad auditiva en España.

Confederación Estatal de Personas Sordas, 2024.

EUD. Informe sobre accesibilidad educativa y diversidad en lengua de signos.

European Union of the Deaf, 2022

OMS. World report on hearing 2021. *Organization Mundial de la Salud*, 2021.

SID-INCO. Informe sobre la disponibilidad de intérpretes y sistemas de apoyo para personas con discapacidad auditiva. Salamanca, Universidad de Salamanca, 2024.

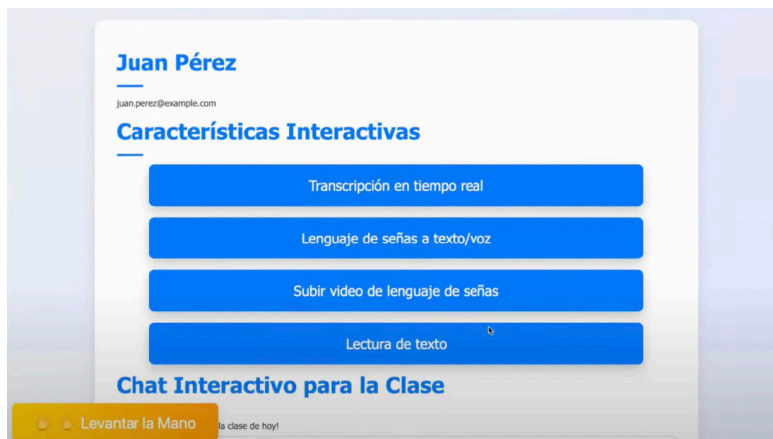
UNESCO. Informe sobre educación inclusiva y cumplimiento de los Objetivos de Desarrollo Sostenible. *Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura*, 2020

ANEXOS

Figura 1. Logo de LearnHack



Figura 2. Pantalla principal de la aplicación



ASIS: ASISTENTE VIRTUAL INTERACTIVO SIMPLIFICADO



LEYRE GARCÍA SAN CAYETANO

GABRIEL MATEOS RUÍZ

HUGO SAMPEDRO HERNÁNDEZ

RUBÉN MARTÍN GARCÍA

RESUMEN

En el presente capítulo se describe el proyecto **ASIS**, un asistente virtual diseñado para mejorar la autonomía de las personas mayores y de aquellas que se encuentran en situación de dependencia. Este trabajo surge de la reflexión en torno a la brecha digital y a las dificultades de accesibilidad que enfrentan ciertos colectivos a la hora de interactuar con la tecnología.

ASIS busca dar respuesta a este problema mediante un **dispositivo físico autónomo**, que funciona con comandos de voz y que no requiere del uso de un teléfono inteligente. El sistema está basado en la plataforma **ESP32**, lo que garantiza un consumo energético reducido, una alta portabilidad y la posibilidad de integrarse con servicios en la nube para el procesamiento de audio y la gestión de información en tiempo real.

ASIS integra inteligencia artificial para reconocer frases complejas, simplificarlas en acciones concretas y responder de manera clara al usuario. El sistema se complementa con una aplicación para dispositivos Android que facilita la configuración inicial, la gestión de perfiles personalizados y la supervisión de dispositivos conectados. El núcleo del asistente se apoya en una API propia y en un motor de interpretación de comandos, encargado de procesar palabras clave, ejecutar funciones y emitir respuestas por voz.

1. CONTEXTUALIZACIÓN DEL PROYECTO

ASIS surge en el marco de las convocatorias Talent UPSA, una iniciativa promovida por la Universidad Pontificia de Salamanca mediante la cual se desarrollan proyectos con elevado impacto global como respuesta a la necesidad de un asistente accesible, intuitivo y no invasivo que se convierte en una herramienta clave para favorecer la inclusión digital y la autonomía personal. La idea inicial surgió tras un taller de impresión 3D impartido en nuestra universidad, que nos inspiró a explorar cómo la tecnología podía ponerse al servicio de colectivos vulnerables.

El envejecimiento de la población mundial es un hecho constatable. Las proyecciones de la **Organización Mundial de la Salud (OMS)** apuntan a que, en 2050, más de una de cada cinco personas tendrá más de 60 años. Este fenómeno demográfico implica una serie de desafíos tanto sociales como sanitarios y tecnológicos, especialmente relacionados con el fomento de la autonomía y la calidad de vida en la vejez.

A este contexto se suma el impacto de la **brecha digital**, que afecta de manera directa a los colectivos de personas mayores y dependientes. El acceso a internet, a los servicios digitales o al uso de aplicaciones móviles continúa siendo limitado para quienes, por motivos de edad o discapacidad, encuentran barreras en la tecnología tradicional. Las interfaces gráficas, los menús complejos o la necesidad de disponer de un smartphone se convierten en obstáculos que marginan a una parte importante de la sociedad.

La necesidad de **tecnologías inclusivas** se hace evidente. Frente a este reto, nuestro equipo concibió la idea de desarrollar un asistente virtual diferente a los que ya existen en el mercado. A diferencia de otros sistemas comerciales, ASIS no depende de un teléfono móvil ni de aplicaciones externas, sino que funciona como un dispositivo independiente que responde a comandos de voz y se conecta directamente a internet.

Así, ASIS se presenta como un **punto entre innovación tecnológica y necesidades sociales reales**, con la ambición de transformar la manera en que las personas mayores interactúan con la tecnología.

2. JUSTIFICACIÓN Y OBJETIVOS

2.1. Justificación

El proyecto ASIS se justifica en la intersección de tres grandes necesidades sociales:

En primer lugar, la **accesibilidad**. Muchas de las soluciones tecnológicas actuales se diseñan para un público generalista, dejando de lado las necesidades específicas de quienes tienen limitaciones físicas, cognitivas o simplemente carecen de familiaridad con los dispositivos digitales. ASIS propone un modelo de interacción basado en la voz, que resulta natural y sencillo incluso para usuarios con poca experiencia tecnológica.

En segundo lugar, la **autonomía personal**. La posibilidad de desenvolverse con mayor independencia en el día a día no solo repercute en la comodidad, sino también en el bienestar psicológico y emocional. Un asistente que pueda responder a comandos básicos, informar de la localización o establecer contacto con cuidadores ofrece un grado extra de seguridad que se traduce en confianza para el usuario.

Por último, la **inclusión social**. Reducir la brecha digital es uno de los objetivos estratégicos de las políticas públicas y de los proyectos de innovación social en la actualidad. ASIS se enmarca en esta línea, aportando una solución que no solo es viable técnicamente, sino que además responde a un problema social de gran magnitud.

2.2. Objetivos Generales

El objetivo principal del proyecto es el diseño e implementación de un **asistente virtual inclusivo y accesible**, que funcione de manera autónoma y proporcione herramientas útiles para la vida diaria de las personas mayores y dependientes.

2.3. Objetivos Específicos

- Desarrollar un dispositivo accesible y portátil.
- Implementar una aplicación Android que permita la configuración inicial, gestión de perfiles y control de dispositivos conectados.
- Diseñar una API propia y un sistema de interpretación de comandos.
- Proporcionar una **interfaz natural de uso**, donde el único requisito sea la voz, eliminando la necesidad de pantallas o smartphones.
- Realizar pruebas de validación en entornos reales que permitan comprobar la efectividad y aceptación del dispositivo.

3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN

3.1. Diseño Conceptual

Desde el inicio, se definió que ASIS debía ser un **dispositivo físico autónomo**. Esta decisión vino motivada por la necesidad de prescindir del smartphone, un elemento que en muchos casos constituye una barrera para las personas mayores. El dispositivo debía ser portátil, sencillo de transportar y de bajo consumo, de manera que su autonomía energética permitiera un uso cotidiano sin complicaciones.

La interacción principal se diseñó en torno a la **voz**, por ser el canal de comunicación más accesible y universal. Esto implica que el hardware debe integrar un

micrófono de calidad para la captura de audio y un altavoz capaz de reproducir respuestas claras y audibles.

3.2. Diseño Técnico

En el plano técnico, ASIS se compone de tres niveles:

1. **Nivel de hardware:** Consiste en una pequeña carcasa con colgante que puede llevarse cómodamente al cuello. Incluye un botón que activa el modo de escucha del asistente, permitiendo que el usuario emita comandos de voz de manera sencilla.
2. **Nivel de comunicaciones:** responsable de la transmisión de datos entre el dispositivo y los servidores en la nube, utilizando protocolos ligeros y eficientes.
3. **Nivel de servicios en la nube:** encargado de procesar los comandos de voz, gestionar la base de datos de usuarios y ejecutar las funciones inteligentes del sistema.

Este diseño modular ofrece una gran flexibilidad, permitiendo incorporar nuevas funcionalidades en el futuro sin modificar en exceso la arquitectura del dispositivo.

3.3. Desarrollo

El desarrollo de ASIS se llevó a cabo en varias etapas sucesivas. En primer lugar, realizamos una **investigación preliminar** sobre las tecnologías disponibles y las necesidades más comunes de los potenciales usuarios. Posteriormente, se abordó el **diseño del hardware**, seleccionando la ESP32 por sus características de bajo consumo y su versatilidad para proyectos de IoT.

La siguiente fase consistió en una aplicación Android, diseñada para la configuración inicial del dispositivo y para gestionar los perfiles de usuario. La aplicación ofrece opciones de registro e inicio de sesión, y permite revisar dispositivos conectados, lo que facilita una personalización adaptada a las necesidades de cada persona.

También nos centramos en la **implementación del software de control**, utilizando librerías para el reconocimiento y la síntesis de voz, así como módulos para la comunicación con los servidores en la nube a través de una API propia que procesa los comandos de voz y devuelve las respuestas correspondientes. Este sistema interpreta frases complejas, las simplifica en acciones concretas y ejecuta la función solicitada, devolviendo siempre una respuesta por voz. En paralelo, se diseñó la infraestructura de servidor, encargada de realizar el procesamiento de los comandos de voz y devolver la respuesta al usuario en tiempo real.

El desarrollo del proyecto ha requerido integrar hardware, software y algoritmos de IA, lo que supuso un importante aprendizaje en múltiples disciplinas tecnológicas.

3.4. Implementación y Validación

Las pruebas se diseñaron con un doble objetivo: comprobar la **viabilidad técnica** del sistema y evaluar su **usabilidad** desde el punto de vista del usuario final.

Los resultados fueron prometedores. ASIS demostró ser capaz de responder a los comandos en tiempos razonables y de mantener una comunicación estable con la nube. Sin embargo, también se identificaron limitaciones, como la necesidad de optimizar el rendimiento del reconocimiento de voz.

4. CONCLUSIONES Y REFLEXIONES FINALES SOBRE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO

El desarrollo de ASIS permitió comprobar que es posible **integrar tecnología de bajo coste con servicios en la nube** para crear un asistente virtual accesible y funcional. El proyecto alcanzó su objetivo de diseñar un prototipo capaz de mejorar la autonomía de las personas mayores y dependientes, al tiempo que sentó las bases para futuros desarrollos.

No obstante, de cara a la transferencia del conocimiento a la sociedad y a su eventual comercialización, se detectaron algunas limitaciones. La primera de ellas está relacionada con la **aceptación social**: la introducción de un nuevo dispositivo requiere un proceso de adaptación, especialmente en colectivos poco habituados al uso de tecnología.

Otra limitación tiene que ver con la **dependencia de la conectividad móvil**. Aunque se empleó una tarjeta SIM para garantizar el acceso a internet sin necesidad de smartphone, existen áreas rurales donde la cobertura sigue siendo insuficiente.

Asimismo, la **seguridad y privacidad de los datos** constituye un reto de primer orden. Al tratarse de un sistema que procesa información sensible y se conecta a servidores externos, resulta imprescindible garantizar altos estándares de protección de datos.

Finalmente, la **escalabilidad** del prototipo hacia un producto comercial plantea desafíos relacionados con la cadena de producción y suministro, la durabilidad del hardware y la necesidad de soporte técnico.

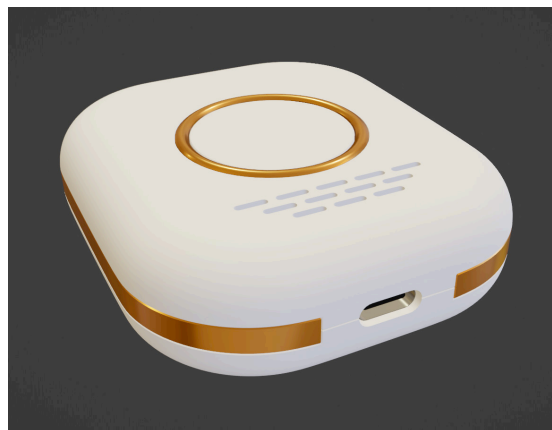
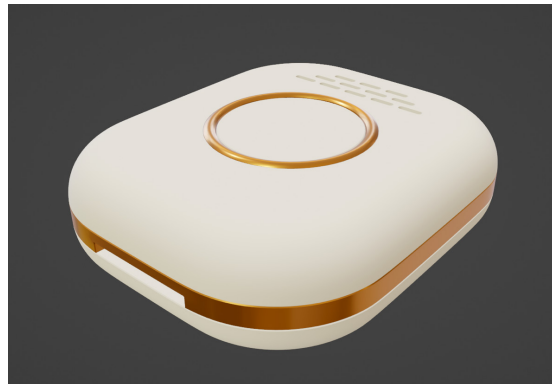
Más allá de lo técnico, el proyecto ha fortalecido nuestras capacidades de trabajo en equipo, creatividad, comunicación y gestión de recursos. Sin embargo, también hemos identificado limitaciones relacionadas con la transferencia del conocimiento, como la necesidad de recursos avanzados de hardware para un despliegue masivo. En definitiva, ASIS representa un paso inicial en la construcción de soluciones accesibles que pueden impactar positivamente en la vida de muchas personas.

BIBLIOGRAFÍA

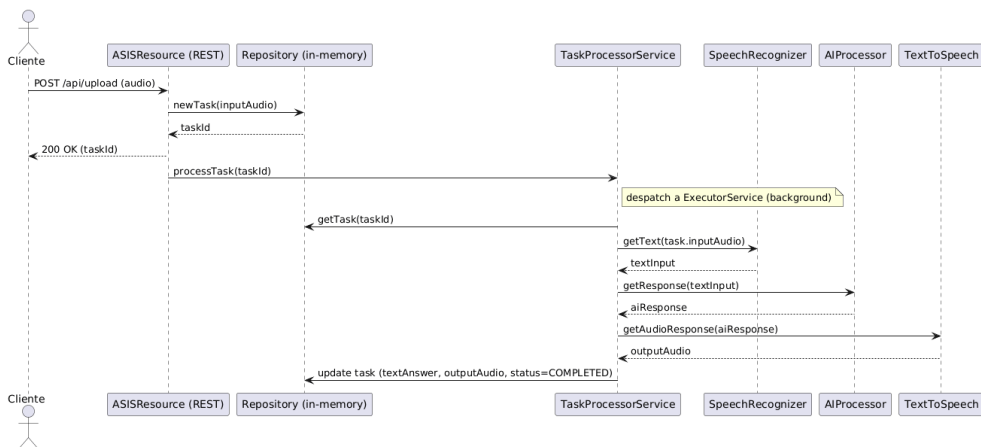
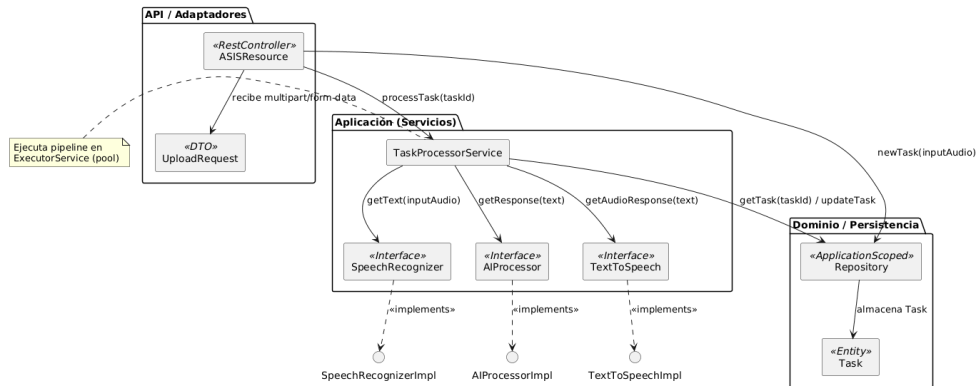
- Organización Mundial de la Salud (2021). *World Report on Ageing and Health*.
- Instituto Nacional de Estadística (2022). *Proyecciones de población en España*.
- Espressif Systems (2022). *ESP32 Technical Reference Manual*.
- Kumar, P., & Singh, R. (2020). *IoT based Assistive Technologies for Elderly Care*. Journal of Ambient Intelligence.
- Corti, K. & Gillespie, A. (2016). *Co-constructing interactivity: Assisted living technologies for older adults*. Universal Access in the Information Society.

ANEXOS

Esquemas técnicos del diseño del hardware



Diagramas de la arquitectura del sistema

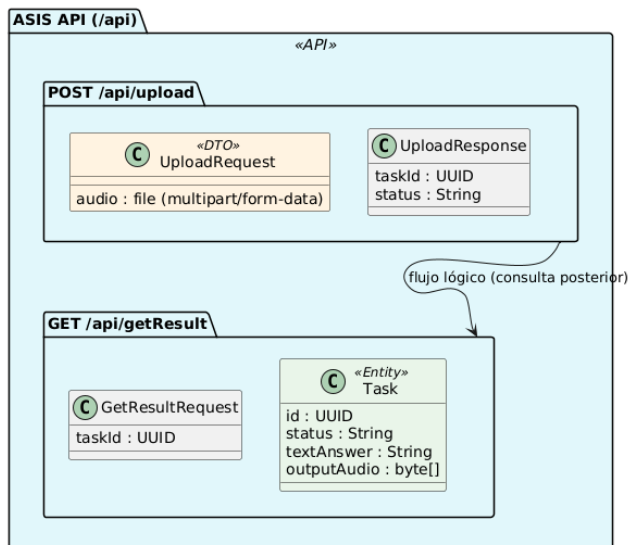


Fotografías del prototipo en fase de pruebas



Listado de comandos básicos implementados

Diagrama de API REST — ASIS (Quarkus)



SISTEMA DE INDUMENTARIA Y EQUIPAMIENTO PARA LA ACTIVIDAD FÍSICA EN LA NATURALEZA <SIENAT>



LÓPEZ-GARCÍA, SERGIO

ROMERO-LOBO, LUCÍA

SUÁREZ-COLLADO, LUCÍA

ELIPE-LORENZO, PABLO

TALAVERA-ESCRIBANO, MARCOS

DEL ARCO-CHÁS, CARMEN

DIEZ-FERNÁNDEZ, PELAYO

RESUMEN

El proyecto SIENAT nace como una solución a la creciente demanda de soluciones tecnológicas que faciliten la práctica segura de actividades físicas en entornos naturales. Su eje vertebrador es una innovadora aplicación móvil que ayuda a los usuarios en la selección de indumentaria y equipamiento deportivo, considerando variables como el tipo de actividad, la intensidad del esfuerzo, las condiciones climáticas y las características personales de cada usuario. La aplicación se fundamenta en un sistema de vestimenta de tres capas, modelo ampliamente validado en la literatura científica y en la práctica deportiva, que optimiza la protección térmica y la adaptación a factores ambientales cambiantes.

Además de ofrecer recomendaciones sobre de ropa y calzado, SIENAT incluye funciones de prevención de riesgos y gestión de emergencias, haciendo de esta herramienta un recurso integral para deportistas experimentados como para prin-

cipiantes. Así, ayuda a disminuir la incidencia de accidentes, lesiones y problemas de salud asociados a una preparación inadecuada, como la hipotermia o la deshidratación. A la vez, promueve la comodidad y el disfrute de las experiencias en la naturaleza, consolidándose como un apoyo esencial para quienes buscan practicar deporte de manera responsable.

En términos de impacto social, el proyecto se relaciona estrechamente con el fomento de hábitos de vida activos y saludables, reforzando la idea de la naturaleza como el escenario ideal para la mejora del bienestar físico, mental y emocional. Desde un punto de vista económico, el modelo abre oportunidades para generar sinergias con sectores estratégicos como el textil, el turismo deportivo y la tecnología aplicada a la salud, lo que eleva su potencial de transferencia y escalabilidad. Finalmente, SIENAT se alinea con los Objetivos de Desarrollo Sostenible (ODS), ayudando a cuidar el entorno natural al fomentar un uso responsable de los recursos.

1. CONTEXTUALIZACIÓN DEL PROYECTO

La práctica de Actividad Física en el Medio Natural (AFMN) se ha convertido en una de las prácticas más populares en el ámbito deportivo y recreativo en los últimos años (Hurtado Barroso, 2024). Este tipo de actividades proporcionan beneficios físicos, psicológicos y sociales, además de que refuerzan el vínculo con el entorno natural (Baena-Extremera & Granero-Gallegos, 2015; Gutiérrez et al, 2018; Olmedilla-Zafra, et al., 2016; Sánchez-Valle et al., 2019). En España, más de la mitad de la población practica deporte de forma regular, y un número creciente lo hace en entornos naturales como montañas, senderos, playas y áreas rurales. Este incremento en la demanda viene acompañado de la necesidad de garantizar experiencias seguras y gratificantes.

La problemática surge cuando los usuarios, especialmente aquellos con poca experiencia, no cuentan con el conocimiento suficiente sobre las características técnicas de la indumentaria o no saben cómo adaptarse a condiciones cambiantes del medio. Una incorrecta preparación puede traducirse en incomodidad, disminución del rendimiento o incluso en situaciones de riesgo que comprometan la salud. Es por ello, que el acceso a información clara, precisa y adaptada se convierte en una herramienta clave para mejorar la experiencia deportiva en la naturaleza.

El entorno digital actual brinda una oportunidad idónea para atender a esta necesidad. Las aplicaciones móviles orientadas con el deporte y la salud ya son parte de la rutina diaria de millones de personas, y el sector de las tecnologías aplicadas al "outdoor" está en continuo crecimiento. No obstante, la mayoría de las soluciones actuales se enfocan en la monitorización del rendimiento físico o en la planificación de rutas, ignorando un aspecto crucial, la preparación en términos de indumentaria y seguridad preventiva. Es precisamente en esta carencia donde SIENAT encuentra

su nicho, ofreciendo un enfoque único que combina personalización, sostenibilidad y transferencia de conocimiento entre diferentes disciplinas.

2. JUSTIFICACIÓN Y OBJETIVOS

La razón de ser del proyecto SIENAT se basa en la necesidad de abordar un vacío identificado en la preparación para actividades deportivas en espacios naturales. Elegir de manera inadecuada la vestimenta o calzado puede derivar en consecuencias negativas que van más allá de la falta de comodidad, convirtiéndose en un riesgo para la salud y la seguridad del practicante. Ejemplos de complicaciones que se pueden prevenir con una adecuada preparación incluyen la hipotermia, deshidratación, lesiones musculares o traumatismos. Aunque existe en el mercado una gran variedad de indumentaria, no todas las prendas son indicadas para cada actividad ni están diseñadas para condiciones específicas, lo que genera confusión entre los consumidores.

Desde esta perspectiva, SIENAT ofrece un modelo innovador de transferencia de conocimiento entre distintas áreas, tecnología, ciencias del deporte y salud, para desarrollar una herramienta práctica, accesible y personalizada. La aplicación no solo pretende solucionar un problema individual, sino también aspira a mejorar la salud pública promoviendo hábitos deportivos seguros y responsables. Además, incentiva un uso sostenible de los entornos naturales, alineándose con las políticas de conservación ambiental y con los Objetivos de Desarrollo Sostenible.

El valor añadido del proyecto radica en su carácter multidisciplinar y escalable, lo que posibilita la colaboración con empresas del sector textil, tecnológico y turístico, generando un impacto económico positivo. Asimismo, la personalización de sus recomendaciones y el enfoque en la educación práctica proporcionan un aspecto diferenciador frente a otras soluciones existentes en el mercado.

El objetivo principal es:

- Brindar apoyo a deportistas, aficionados y personas interesadas en iniciarse en actividades deportivas en entornos naturales, facilitando la identificación de la indumentaria y el equipamiento adecuados para cada práctica, en función de las condiciones ambientales, disciplinas deportivas y del perfil del usuario.

Los objetivos específicos son:

- Promover una práctica deportiva segura y responsable mediante la elección correcta del equipo e indumentaria adaptados a cada actividad y entorno natural.
- Orientar a los usuarios sobre las características técnicas y funcionales de la indumentaria deportiva, fomentando un aprendizaje práctico y accesible.
- Integrar criterios de sostenibilidad en la recomendación de productos, promoviendo materiales respetuosos con el medio ambiente.

- Favorecer la transferencia interdisciplinar de conocimiento entre el ámbito deportivo, tecnológico y de la salud, garantizando recomendaciones precisas y actualizadas.
- Contribuir a la consolidación del turismo activo y sostenible en entornos naturales, reforzando el valor del patrimonio natural y cultural.

3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN

3.1. Diseño de la solución

El proyecto SIENAT se concibe como una aplicación móvil destinada a acompañar a las personas que realizan actividad física en entornos naturales, tanto a nivel recreativo como deportivo. Su propósito no es únicamente recomendar la indumentaria más adecuada, sino también orientar al usuario en la preparación integral de su práctica, favoreciendo la seguridad, reduciendo riesgos y mejorando la experiencia en contacto con la naturaleza.

El diseño conceptual se apoya en el sistema de vestimenta por capas. Este esquema contempla una primera capa de contacto con la piel, que regula la temperatura corporal y evacua la humedad; una segunda capa de aislamiento destinada a conservar el calor; y una tercera capa externa que protege frente a factores ambientales como el viento, la lluvia o la nieve. A partir de esta base, la aplicación ofrece recomendaciones personalizadas teniendo en cuenta la disciplina deportiva, la intensidad del esfuerzo, las condiciones ambientales y del terreno, así como las características individuales de cada usuario. De esta manera, SIENAT no se limita a transmitir información general, sino que proporciona orientaciones específicas y comprensibles para diferentes perfiles y contextos.

3.2. Desarrollo tecnológico

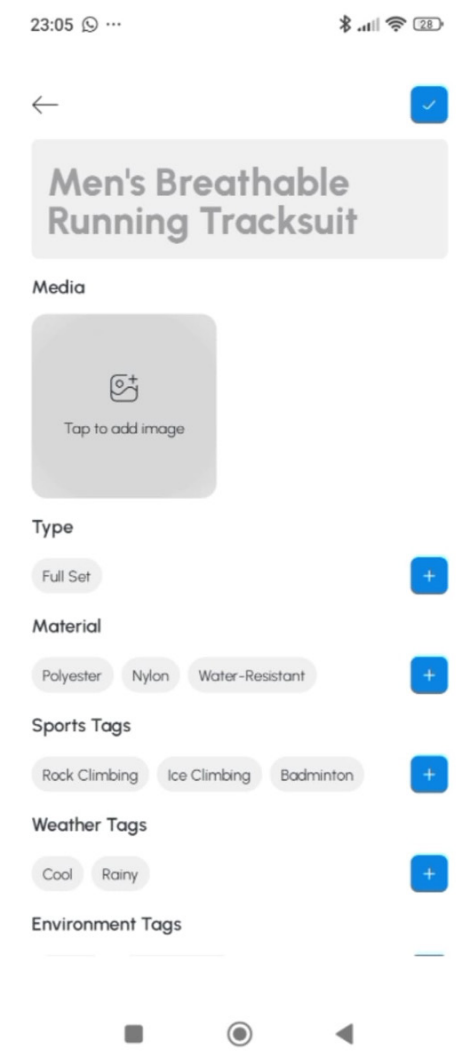
El desarrollo de SIENAT se ha planteado con un enfoque robusto y escalable. La aplicación se implementa mediante tecnologías multiplataforma, como React Native o Flutter, lo que garantiza compatibilidad en dispositivos Android e iOS y optimiza los tiempos y costes de programación. La gestión de datos se lleva a cabo a través de sistemas fiables como PostgreSQL o Firebase, que permiten almacenar información relativa a climatología, equipamiento, inventario y perfiles de usuario.

La posible integración de APIs meteorológicas y de geolocalización dota a la aplicación de capacidad para actualizar las recomendaciones en tiempo real, ofreciendo un servicio dinámico y adaptado a las condiciones concretas del entorno. Asimismo, el uso de servicios en la nube, mediante plataformas como AWS, Google Cloud o Microsoft Azure, asegura la escalabilidad del sistema, el almacenamiento seguro de

la información y una gestión eficiente de usuarios concurrentes. Todo ello se complementa con la implementación de protocolos de seguridad avanzados, entre ellos cifrado de extremo a extremo, control de accesos y cumplimiento con la normativa RGPD, lo que garantiza la protección de los datos personales y refuerza la confianza de los usuarios.

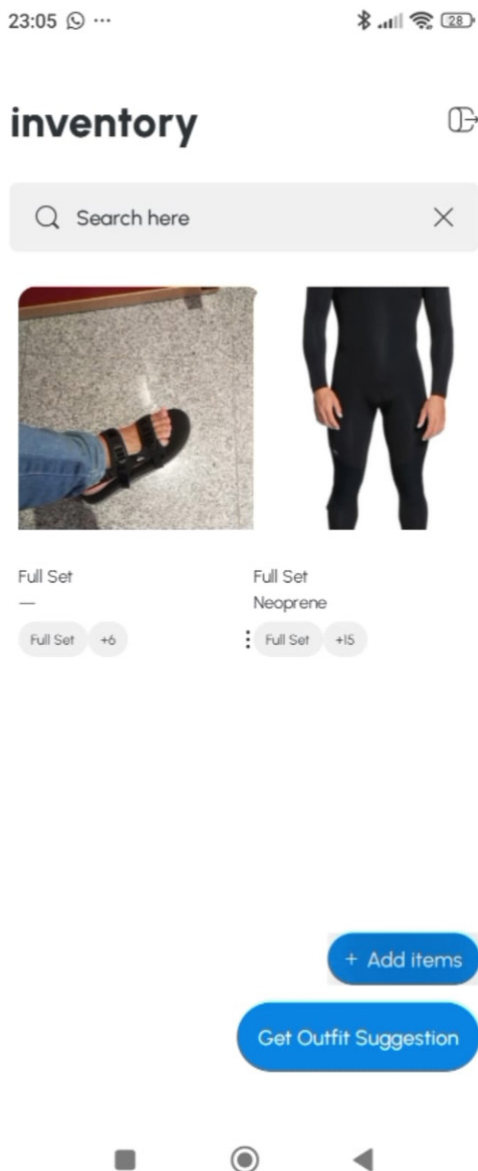
Asimismo, en la figura 1 se puede ver la posibilidad de incorporar manualmente en la app la prenda de ropa con sus características específicas (material, tipo de deporte, meteorología...).

Figura 1. Incorporación manual de materiales.



Por otro lado, en la figura 2, se observa el armario/inventario de materiales que tenemos incorporados en nuestra app y a partir de ahí, seleccionar los que mejor se adaptan a las actividades deportivas que vamos a realizar.

Figura 2. Inventario



Elaboracion propia.

3.3. Implementación

La implementación del proyecto se ha organizado en fases sucesivas que han permitido avanzar de forma progresiva y coordinada. En primer lugar, se realizó una revisión exhaustiva del estado del arte con el fin de identificar aplicaciones con funciones similares y analizar sus limitaciones. Posteriormente, se diseñó una propuesta inicial en la que se combinaron criterios técnicos con principios de simplicidad y usabilidad.

Sobre esta base se desarrolló un prototipo mínimo viable, que fue sometido a pruebas piloto con profesionales del ámbito deportivo y salud. Estas pruebas fueron fundamentales para validar la utilidad de la aplicación, detectar áreas de mejora y perfeccionar el sistema a través de iteraciones sucesivas. Finalmente, el siguiente paso previsto en el proceso de implementación es la fase de transferencia de resultados, dirigida tanto a la difusión de la herramienta como a la creación de sinergias con sectores estratégicos como el textil, el turístico y el tecnológico.

Este itinerario refleja el carácter interdisciplinar del proyecto y su vocación aplicada. La colaboración entre especialistas de distintos campos ha sido clave para desarrollar una herramienta que combina solidez científica, viabilidad técnica y un lenguaje accesible, garantizando la transferencia efectiva del conocimiento a los usuarios finales.

4. CONCLUSIONES Y REFLEXIONES FINALES SOBRE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO

El proyecto SIENAT propone una aplicación móvil para recomendar la indumentaria más adecuada para la práctica de actividades físicas en el medio natural, aborda una necesidad fundamental: garantizar la seguridad, confortabilidad y el disfrute de los usuarios al practicar deportes al aire libre en diferentes condiciones.

Sin embargo, como en cualquier proyecto de innovación tecnológica, existen limitaciones inherentes en cuanto a la transferencia del conocimiento involucrado a su puesta en práctica en el medio natural real, estas deben ser tenidas en cuenta para asegurar el éxito y la adopción de la herramienta de una manera fiable y que se acerque a la realidad de dicha actividad. A continuación, se presentan algunas conclusiones y reflexiones clave sobre algunas de estas limitaciones.

El principal desafío en la transferencia de conocimiento radica en la diversidad de perfiles de los usuarios. Si bien la aplicación está diseñada para ofrecer recomendaciones personalizadas, existe la posibilidad de que algunas personas, especialmente aquellas sin experiencia previa en actividades al aire libre, no comprendan el vocabulario técnico completamente, así como algunos de los beneficios, por ejemplo, el sistema de tres capas y cómo este se adapta a las condiciones climáticas cambiantes.

El conocimiento técnico que sustenta estas recomendaciones debe ser lo más accesible y comprensible para los usuarios independientemente del nivel, lo cual puede requerir un esfuerzo adicional en la interfaz de usuario y en la explicación de las razones detrás de las sugerencias ofrecidas por la aplicación.

Por otro lado, el proyecto SIENAT integra diversas disciplinas: ciencia deportiva, tecnología y salud. Si bien esta integración es un aspecto diferencial y valioso, puede resultar en dificultades para la transferencia efectiva del conocimiento entre disciplinas.

Otro reto importante en la transferencia de conocimiento está relacionado con la adaptación a contextos locales y climáticos específicos. Si bien la app ofrece recomendaciones personalizadas, los usuarios de diferentes regiones pueden enfrentar condiciones ambientales muy distintas, y la aplicación debe ser capaz de adaptar sus sugerencias a situaciones diferentes entendidas desde contextos dispares.

Esto plantea un desafío en cuanto a la precisión y relevancia de las recomendaciones en función de los diferentes contextos geográficos y climáticos, lo que requiere de un proceso constante de validación y ajuste en función de las nuevas condiciones actualizadas a momentos estacionales que varían dependiendo de la época del año y de las condiciones en las que se vaya a dar.

Aunque el proyecto SIENAT está diseñado para superar las barreras tradicionales en la selección de indumentaria para actividades físicas en el medio natural, la transferencia del conocimiento sigue siendo un proceso complejo que involucra múltiples capas de interacción entre los usuarios, el conocimiento experto y la tecnología. Asimismo, la personalización es uno de los puntos fuertes de la app, pero la efectividad de esta personalización dependerá en gran medida de la habilidad de la aplicación para adaptarse a los diferentes perfiles de usuario y a las condiciones cambiantes del entorno.

Finalmente, una de las mayores oportunidades del proyecto radica en su capacidad para contribuir a un estilo de vida más saludable y seguro para los usuarios, mientras promueve el uso responsable de los recursos naturales. Para lograr popularidad y una transferencia eficaz del conocimiento, será necesario superar los obstáculos relacionados con la resistencia al cambio, la accesibilidad tecnológica y la personalización del contenido. Solo mediante una estrategia que considere estos aspectos se podrá asegurar el éxito y la sostenibilidad del proyecto a largo plazo.

BIBLIOGRAFÍA

- Baena-Extremera, A, & Granero-Gallegos, A. Efectos de las actividades en la naturaleza en la predicción de la satisfacción de la Educación Física. *Retos*, 20015; 28: 9–14.
- Gutiérrez, M, Tomás, JM, Romero, I, *et al.* Apoyo social percibido, implicación escolar y satisfacción con la escuela. *Revista de Psicodidáctica*, 2017; 22(2): 111–117.

Hurtado Barroso, M, González Melero, E, Párraga Montilla, JA *et al.* La formación docente en actividades físicas en el medio natural en la Educación Física en España: Un análisis de los planes de estudio universitarios. *Retos*, 2024; 56: 681-689.

Olmedilla-Zafra, A, Navas-Luque, M, y Vilar-López, R. Programa de educación y aventura en el medio natural: efectos en el desarrollo socioemocional y académico en estudiantes de educación primaria. *Revista Iberoamericana de Psicología del Ejercicio y el Deporte*, 2016; 11(1): 77-84.

Sánchez- Valle, M, Fernández-Martínez, E y López-Sánchez, GF. Impacto de un programa de educación y aventura en el medio natural sobre la autoestima y el autoconcepto de los adolescentes. *Revista de Educación*, 2019; 380: 226-250.

COILAPP



CARLOS CAÑO GÓMEZ

ALBERTO PEDRERO ESTEBAN

ANA LUGONES HOYA

RESUMEN

COILApp es una aplicación web diseñada para optimizar la gestión de participación en proyectos COIL. Permite a los alumnos registrar sus datos de forma autónoma y proporciona a la UPSA un sistema centralizado para administrar, consultar y certificar estas experiencias colaborativas de manera más segura, reduciendo la carga administrativa del profesorado.

1. CONTEXTUALIZACIÓN DEL PROYECTO

La internacionalización es un proceso enfocado en el desarrollo y adquisición de distintas destrezas a partir de la interacción internacional e intercultural, es precisamente por dichos beneficios por lo que se mantiene vigente en las instituciones de educación superior. En la actualidad, aunque se mantiene su metodología tradicional de movilidad Erasmus, han surgido nuevas problemáticas y con ello nuevos paradigmas. Estos paradigmas responden a las necesidades y posibilidades del estudiantado interesado en expandir sus conocimientos. Entre estas innovadoras propuestas se encuentra la “Internacionalización en Casa” que surge a partir de una problemática interna y se convierte en un fenómeno educativo.

Entre los pilares que sustentan este nuevo enfoque “en casa” está una mayor flexibilidad, que permite a estudiantes y docentes desarrollarse como profesionales y

adquirir mayores competencias sin importar las dificultades de movilidad o distancia. Esta nueva metodología educativa fue de gran ayuda en el período marcado por el COVID (Fuentes Huerta, 2021) como una opción ante las restricciones de movimiento y la necesidad de retomar la integración internacional. Dicha adaptabilidad marca un punto de inflexión en el presente y ayuda a vislumbrar el futuro de la internacionalización aún en momentos de incertidumbre.

El Aprendizaje Colaborativo Internacional en Línea (COIL) (SUNY Online, n.d.) es una novedosa metodología educativa cimentada en el intercambio virtual estructurado y la capacidad transformativa. A través de instituciones como SUNY COIL Center, The Stevens Initiative, UNICollaboration o Virtual Exchange Coalition, entre otras, y redes internacionales como LatAm COIL y COIL Connect, se ha impulsado esta propuesta pedagógica que enlaza estudiantes y docentes de distintas culturas con el objetivo de promover la formación, el diálogo y la colaboración en proyectos conjuntos integrados en los programas curriculares tradicionales.

2. JUSTIFICACIÓN Y OBJETIVOS

Ilustración 1: Logotipo COILApp.



En la Universidad Pontificia de Salamanca (UPSA), la implementación de proyectos COIL ha demostrado ser una iniciativa con gran valor pedagógico. Sin embargo, su éxito ha puesto de manifiesto un desafío operativo significativo. Actualmente, el proceso de gestión de datos para la emisión de las constancias oficiales de participación implica la recopilación manual de datos tanto personales como académicos de cada uno de los participantes. Dicha metodología es intrínsecamente ineficiente, consume un tiempo valioso, y es altamente propensa a errores humanos en la transcripción y manejo de la información.

Para abordar de manera directa los desafíos operativos identificados en la UPSA, se propuso el diseño y desarrollo de COILApp. El valor fundamental de esta aplicación reside en su capacidad para crear un ecosistema digital unificado que integra a los tres actores principales del proceso: la coordinación COIL y el personal de Relaciones In-

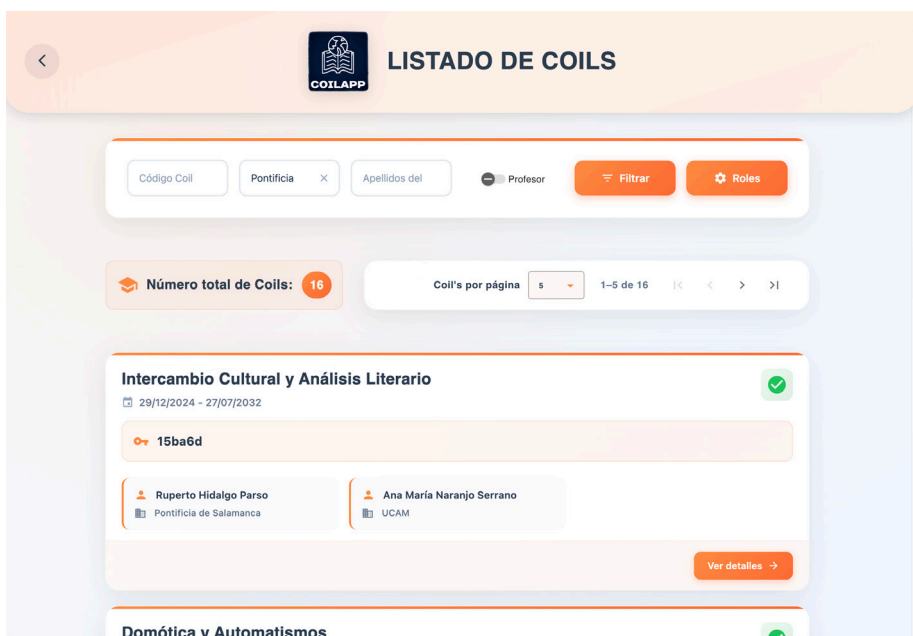
ternacionales (RRII), el profesorado y el alumnado. La aplicación está diseñada para centralizar y automatizar el ciclo de vida completo de la gestión de datos, desde el registro inicial de los participantes y la creación de proyectos hasta la recopilación segura de información y la facilitación de su exportación para la posterior emisión de certificados, además de mantener un diseño intuitivo y fácil de comprender.

Tomando en consideración aquellas necesidades y requerimientos previamente desglosados, este proyecto siempre ha tenido como pilar la simplicidad, la adaptabilidad y la practicidad. Por ello, los objetivos que guían el desarrollo de este proyecto son los siguientes:

- Desarrollar una aplicación que simplifique y optimice el proceso de gestión y certificación de los proyectos COIL, asegurando una trazabilidad clara y segura de la información, reduciendo de esta manera la carga administrativa sobre coordinadores y profesores.
- Centralizar la gestión de datos de alumnos y docentes en una plataforma digital.
- Facilitar el acceso y seguimiento histórico de los proyectos COIL.
- Optimizar la exportación de datos de los proyectos COIL.

3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN

Ilustración 2: Ejemplo de pantalla del listado de COILS en COILApp.



La aplicación web consta de dos partes fundamentales: la parte del cliente o *Frontend* y la parte del servidor o *Backend*. Para la primera, se ha utilizado el framework de desarrollo Angular (basado en JavaScript). Para la segunda se ha utilizado Node, junto con el framework Express.js. La parte de almacenamiento de datos que se lleva a cabo desde el servidor se ha implementado a través de una base de datos MongoDB.

La comunicación entre la parte cliente y servidor/BBDD se lleva a cabo de la siguiente manera:

1. El usuario interactúa con la interfaz implementada en Angular;
2. A través de Angular se utilizan servicios HTTP para realizar solicitudes al *Backend*, enviando y recibiendo datos en formato JSON;
3. Firebase se comunica con Angular para poder almacenar los usuarios y contraseñas de forma encriptada y los ficheros de tipo pdf, doc, jpeg, etc de evidencias;
4. Mediante el *framework* de Express para el entorno de ejecución NodeJS, en el *Back-end*, se reciben las solicitudes y se redirigen a las rutas definidas;
5. Se ejecutan operaciones en el *Backend* definidas en los controladores, que incluye acceso a la base de datos utilizando Mongoose;
6. Se accederá a la base de datos MongoDB para realizar operaciones CRUD;
7. El *Backend* enviará respuestas al *Frontend* en Angular, en formato JSON;
8. Por último, en Angular se actualiza la interfaz de usuario según las respuestas del *Backend*, lo que puede incluir mostrar datos de COIL, alumnos o realizar otro tipo de acciones.

3.1. Frontend

En la parte de *Frontend* se crea la interfaz de usuario, responsable no solo de la interacción; sino de definir la estructura de contenidos que se mostrará en pantalla y su estilo. El objetivo es que el usuario tenga la mejor experiencia posible de usabilidad, accesibilidad, rendimiento (velocidad de la web), compatibilidad en navegadores, etc.

Para generar los estilos de la interfaz base se emplean ficheros CSS, no obstante, con el objetivo de caracterizar la aplicación con un estilo profesional y moderno se recurrió a la librería de Angular Material. Del mismo modo, se utilizó Bootstrap para la creación e implementación de iconos.

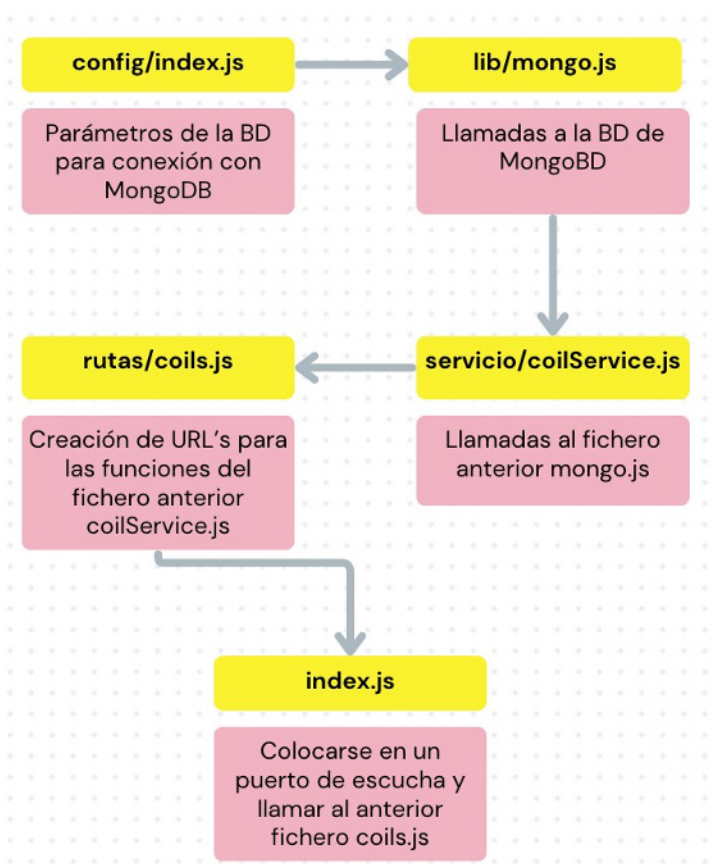
Como se ha indicado anteriormente, para la parte de *Frontend* se ha utilizado Angular, que es un *framework* desarrollado por Google y que está escrito en el lenguaje de programación TypeScript. Entre sus principales características destaca el uso de componentes que definen vistas en HTML, construidas por conjuntos de elementos en la interfaz.

La comunicación de la parte cliente con el servidor se realiza a través de servicios. Un servicio es un proveedor de datos que mantiene la lógica de acceso a los mismos. El uso del servicio en COILApp se utiliza para definir funciones que comunican el cliente con el *Backend* para obtener o enviar datos. De esta manera, se encapsula la lógica en la cual se realizan llamadas HTTP. Gracias a esto es posible reutilizar diversas funciones dentro de la aplicación en varios componentes haciéndola modular y más fácil de mantener.

El uso de enrutamiento permite que en la aplicación se muestren diferentes vistas al usuario en función de la URL, como si fuera un sistema de navegación. Asimismo, se hace uso de guardas, que son funciones utilizadas para proteger determinadas rutas.

3.2. Backend

Ilustración 3: Esquema básico de ficheros *Backend*.



El *Backend* es el otro componente principal de la aplicación. Esta parte está -que no es visible para el usuario- se encarga de la lógica de la aplicación, es decir, la manipulación con sentido de los datos, así como de la comunicación entre la base de datos y el *Frontend*. Para su implementación, se ha utilizado Node y el framework Express.js.

Como ya se ha indicado, los datos son almacenados en una BBDD MongoDB. Para facilitar la comunicación entre el servidor y la base de datos, COILApp utiliza un controlador llamado Mongoose.

Mongoose facilita la interacción entre MongoDB y NodeJS. Esto permite trabajar con objetos en vez de con consultas, así como definir relaciones entre colecciones de documentos en MongoDB. “NodeJS es concebido como un entorno de ejecución JavaScript asíncrono basado en eventos, Node.js está diseñado para construir aplicaciones de red escalables” (Node.js, n.d.)

Para la comunicación con el *Frontend* se definen unas rutas utilizando Express para manejar las solicitudes HTTP realizadas por el cliente. Estas rutas son las que ejecutarán la lógica de la aplicación y accederán a la base de datos. ExpressJS es uno de los marcos Modelo-Vista-Controlador (MVC) más utilizados para crear aplicaciones web y API con NodeJS. Lo anteriormente señalado proporciona soporte para enrutamiento y métodos HTTP.

3.3. Base de datos

Para poder funcionar correctamente, COILApp hace uso de dos bases de datos: por un lado, MongoDB, donde se guardan los datos de todo tipo de registros en la aplicación (todo lo que tiene que ver con los proyectos COIL); por otro lado, las contraseñas para inicios de sesión o registros como usuario se almacenan en una base de datos de Firebase.

Mientras que “MongoDB es una base de datos de documentos que ofrece una gran escalabilidad y flexibilidad, y un modelo de consultas e indexación avanzado” (MongoDB, n.d.), Firebase de Google es una plataforma en la nube para el desarrollo de aplicaciones web y móvil. Se utiliza Firebase para establecer una comunicación a través de Angular y así almacenar a los usuarios de forma encriptada, donde solo se podrá visualizar los correos electrónicos de estos y no las contraseñas, además de permitir el inicio de sesión a través de terceros como Google. También se emplea para almacenar ficheros pdf, word, imágenes, etc., en la nube que estén relacionados con cada COIL.

4. CONCLUSIONES Y REFLEXIONES FINALES DE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO

En el contexto actual, debido al desarrollo y avance progresivo de la información y los datos, la mayoría de las consultas suelen centrarse en sitios o aplicaciones web

con bases datos, sean locales o en la nube. Dichos elementos son herramientas que han contribuido al cumplimiento de un amplio abanico de objetivos propuestos por grupos de diferentes dimensiones.

Este tipo de aplicaciones está pensado, entre otras funciones, a que sea más gratificante su uso al tener una mejor usabilidad y/o interfaz adaptada al tipo de usuario específico y de esta manera estos simpatizasen en el entorno, para que no solo instalen o prueben estas aplicaciones, sino que las llegan a utilizar.

Algunas de las funciones para las cuales este tipo de aplicaciones están proyectadas son: la obtención de una mayor gratificación al momento del manejo promovido por una mejor funcionalidad, una interfaz adaptada a un tipo específico de usuarios y simpatizante con el entorno, con el objetivo de superar las etapas de instalación y prueba, llegando finalmente a una utilización plena diaria y práctica de las aplicaciones.

No obstante, desde una perspectiva aplicada el principal objetivo de estas aplicaciones sería la capacidad de minimizar la carga de procesos para un usuario. En relación con esta consideración se sitúa el caso de la aplicación trabajada a lo largo de este proyecto, que ha sido desarrollada para la creación de COIL y sus desencadenantes.

Se ha logrado centralizar la gestión de datos de todos los actores implicados en una plataforma única, segura y estructurada, eliminando la fragmentación de la información y reduciendo el riesgo de errores; permitiendo, de esta manera, un menor uso de recursos en los procesos de revisión y corrección, y haciendo posible centrar la atención en los estudiantes y sus experiencias educativas.

Además, se ha establecido la base para un repositorio histórico fiable de las experiencias COIL que facilita el seguimiento y la evaluación estratégica del programa a largo plazo y crea, de esta manera, históricos referenciales que colaboran a la replicabilidad y promoción de proyectos de colaboración interuniversitaria internacional.

En conclusión, se implementó con éxito la aplicación, incluyendo autenticación segura, consulta filtrada de COIL y participantes, validación de alumnos, generación automatizada de datos del COIL en formato Excel o CSV y almacenamiento de los datos de forma persistente en una base de datos.

BIBLIOGRAFÍA

Fuentes Huerta MC. ¿Qué significa internacionalización en casa? [Internet]. 2021 [citado 11 de agosto 2025]. Disponible en:

<https://www.urp.edu.pe/pdf/id/28033/n/internacionalizacion-en-casa-vf.pdf>

MongoDB. What is MongoDB? [Internet]. [citado 11 agosto 2025]. Disponible en:

<https://www.mongodb.com/es/company/what-is-mongodb>

Node.js. About Node.js [Internet]. [citado 11 agosto 2025]. Disponible en:

<https://nodejs.org/es/about>

SUNY Online. What is COIL? [Internet]. s. f. [citado 11 agosto 2025]. Disponible en:

<https://online.suny.edu/introtocoil/que-es-coil/>

SMARTSKIN: INTELIGENCIA ARTIFICIAL APLICADA A LA DERMATOLOGÍA



ANDREA PÉREZ GALÁN

CLAUDIA MARTÍN VILLORIA

MONTSERRAT MATEOS SÁNCHEZ

RESUMEN

SmartSkin es un prototipo de apoyo al cribado y diagnóstico temprano de lesiones dermatológicas a partir de imágenes clínicas. Integra un modelo de aprendizaje profundo basado en una arquitectura tipo Vision Transformer (MedFormer; Chen et al., 2022) entrenado con más de 10.000 imágenes del conjunto HAM10000 (Tschandl et al., 2018), y una interfaz web ligera (Abid et al., 2019) que permite cargar una fotografía y obtener un top-3 de diagnósticos probables (p. ej., melanoma, carcinoma basocelular, queratosis actínica, etc.) con porcentajes de confianza. En evaluación académica, el sistema alcanzó precisiones en el rango del 70–80%, con 60% de acierto top-1 y 83% top-3 en un lote de imágenes nuevas, y tiempos de inferencia de 2–3 s por imagen en CPU. La propuesta se orienta a Atención Primaria, con el objetivo de reducir derivaciones innecesarias, priorizar casos de riesgo y contribuir a un diagnóstico más precoz, especialmente en entornos con listas de espera y déficit de especialistas.

1. CONTEXTUALIZACIÓN DEL PROYECTO

Las consultas por lesiones cutáneas constituyen un motivo frecuente en Atención Primaria, primer nivel de contacto con el sistema sanitario. La escasez de dermatólogos en determinados territorios y las listas de espera en consultas es-

pecializadas dificultan el acceso temprano a diagnósticos, con impacto en la morbimortalidad del cáncer cutáneo y en el coste del sistema. En paralelo, se han consolidado tendencias como la teledermatología, el auge de la eHealth y la IA clínica. En este contexto, SmartSkin nace como una prueba de concepto universitaria para explorar cómo un sistema de inteligencia artificial puede apoyar al médico de familia en el cribado de lesiones, agilizando decisiones de derivación y mejorando la priorización.

En el ecosistema de soluciones, coexisten aplicaciones comerciales de autocribado para población general (p. ej., SkinVision, 2025), plataformas de apoyo al diagnóstico basadas en bases de conocimiento (p. ej., VisualDx, 2025), y proyectos de investigación que evalúan CNNs (Convolutional Neural Networks) sobre imágenes dermatoscópicas. SmartSkin se diferencia por su foco en el uso profesional en consulta con flujo local, su top-3 de resultados para facilitar el diagnóstico diferencial y su orientación a una integración futura con flujos de telemedicina y la historia clínica electrónica.

2. JUSTIFICACIÓN Y OBJETIVOS

Existen varios motivos que justifican el nacimiento y la existencia del proyecto SmartSkin:

- El problema sanitario: debido a la elevada demanda de valoración dermatológica en Atención Primaria, con la consecuente saturación de especialistas y tiempos de espera prolongados.
- Oportunidad tecnológica: madurez de técnicas de aprendizaje profundo para visión por computador y disponibilidad de datasets médicos públicos (Code-lla et al., 2019; Tschandl et al., 2018) que permiten entrenar modelos robustos.
- Impacto social: una herramienta de cribado rápida y accesible puede favorecer la detección temprana de lesiones malignas, con mejoras de pronóstico y eficiencia en el uso de recursos.
- Alineación con ODS (Objetivos de Desarrollo Sostenible): contribuye a ODS 3 (Salud y Bienestar), ODS 9 (Industria, Innovación e Infraestructura) y ODS 17 (Alianzas), al promover acceso equitativo, innovación y colaboración entre academia, clínica y empresa.

Como objetivo general se plantea desarrollar una herramienta de IA capaz de clasificar automáticamente lesiones dermatológicas a partir de imágenes clínicas, presentando un top-3 de diagnósticos más probables con su confianza, como apoyo a la toma de decisiones en Atención Primaria.

En cuanto a los objetivos específicos se encuentran los siguientes:

1. Entrenar y optimizar un modelo MedFormer/ViT (Chen et al., 2022; Dosovitskiy et al., 2021) con más de 10.000 imágenes etiquetadas del dataset HAM10000 (Tschandl et al., 2018), aplicando regularización y data augmentation (incluyendo MixUp (Zhang et al., 2018) y Label Smoothing (Müller et al., 2019) para mejorar robustez y calibración.
2. Implementar una interfaz web minimalista (Abid et al., 2019) que permita cargar imágenes, ejecutar inferencia local y visualizar resultados de manera clara para profesionales no expertos en IA.
3. Evaluar el rendimiento con métricas de precisión y pérdida, y validar funcionalmente el flujo completo con un conjunto de imágenes no vistas.
4. Explorar el impacto clínico-económico potencial (reducción de derivaciones, priorización de sospechas, tiempos de respuesta) y las barreras regulatorias/éticas para su transferencia (Unión Europea, 2017).

3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN

a) Diseño conceptual del sistema

SmartSkin se diseñó como un pipeline modular con tres capas principales:

1. **Capa de datos:** curación y preprocesado de imágenes clínicas de HAM10000 (Tschandl et al., 2018). Incluye redimensionado (224×224), normalización y aumentos de datos (volteos, rotaciones moderadas, ajustes de brillo/contraste) para mejorar invariancias y prevenir sobreajuste.
2. **Capa de modelo:** un Vision Transformer adaptado al ámbito médico (MedFormer; Chen et al., 2022), partiendo de pesos preentrenados (Dosovitskiy et al., 2021) y aplicando fine-tuning. Se incorporaron técnicas de MixUp (Zhang et al., 2018) y Label Smoothing (Müller et al., 2019) para calibrar probabilidades y mejorar la generalización.
3. **Capa de interfaz:** una aplicación local (script app.py) basada en Gradio (Abid et al., 2019) que permite al usuario subir una imagen (.jpg/.png), ejecuta el mismo preprocesado y muestra el top-3 de etiquetas con sus porcentajes.

b) Flujo de datos y clases

El conjunto HAM10000 (Tschandl et al., 2018) contiene múltiples clases de lesiones cutáneas. En la configuración del prototipo, se priorizó la salida clínica en tres macro-categorías de interés (p. ej., melanoma, carcinoma basocelular y queratosis/

lesiones queratósicas), sin perder la granularidad del entrenamiento multiclase, a fin de simplificar la interpretación en consulta.

c) Entrenamiento y validación

- **Infraestructura:** equipo local con GPU RTX 3070 Ti.
- **Estrategia:** fine-tuning del (MedFormer; Chen et al., 2022) con 10 épocas por ciclo, duración aproximada 30 min/ciclo.
- **Métricas:** precisión en validación $\approx 70\%$ (rango 70–80% según experimento); seguimiento de pérdida y confianza promedio.
- **Validación funcional:** sobre 30 imágenes nuevas no vistas, el sistema procesó sin fallos; 60% acierto top-1 y 83% de casos con el diagnóstico correcto en el top-3. Casos con confianza $<30\%$ se asociaron a sombras intensas o baja nitidez.

d) Interfaz y experiencia de uso

La interfaz prioriza simplicidad y rapidez: un botón para subir imagen y un panel de resultados con tres opciones diagnósticas y sus porcentajes de confianza. El tiempo medio de inferencia en CPU fue de 2–3 segundos por imagen, aceptable para su uso en consulta. El flujo trabaja en local (privacidad) y no requiere de configuración compleja.

e) Consideraciones éticas, de seguridad y regulatorias

- **Privacidad y seguridad:** al operar en local, se minimiza el intercambio de datos sensibles. Para escenarios reales, sería necesario incorporar cifrado, control de accesos y auditoría.
- **Riesgos de sesgo:** la representatividad del dataset (tipos de piel, iluminación, dispositivos) condiciona el desempeño; se plantea ampliar datos con muestras diversas y protocolos de re-entrenamiento continuo.
- **Marcado regulatorio:** una eventual transferencia al sistema sanitario requeriría cumplir MDR (Detección y respuestas administradas) (Unión Europea, 2017) y normativas de IA confiable (evaluación de seguridad, gestión de riesgo, trazabilidad y vigilancia poscomercialización).
- **Interoperabilidad:** para maximizar su utilidad clínica, se propone diseñar conectores con la historia clínica electrónica y plataformas de teledermatología, manteniendo logs y exportables para una segunda opinión.

f) Comparativa y posicionamiento

Frente a aplicaciones de autodiagnóstico de pago orientadas a población general, SmartSkin plantea un uso profesional y asistencial con un top-3 de diagnósticos y latencia baja. En comparación con plataformas basadas en bases de conocimiento, la IA potencia la detección de patrones sutiles a nivel de imagen completa (relaciones entre parches). Frente a prototipos académicos centrados en binario maligno/benigno, el enfoque multiclase va a facilitar ese diagnóstico diferencial.

4. CONCLUSIONES Y REFLEXIONES FINALES SOBRE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO

SmartSkin demuestra la viabilidad técnica de integrar IA de visión y una interfaz usable para apoyo al cribado dermatológico en Atención Primaria. Con un rendimiento prometedor ($\approx 70\%$ de precisión y 83% top-3 en pruebas académicas) y tiempos de respuesta adecuados (2–3 s en CPU), el prototipo cumple su propósito de prueba de concepto y sienta bases para futuras mejoras. Su orientación a impacto social (detección temprana, reducción de listas de espera, acceso en zonas rurales) lo alinea con las prioridades de salud pública y transformación digital.

En cuanto a las limitaciones detectadas se encontró que el rendimiento se midió en condiciones académicas con datasets públicos; pero se requiere validación multicéntrica en entornos reales con protocolos clínicos y gold standards.

En la calibración de probabilidades también se encontraron limitaciones a pesar de que se aplicaron técnicas como MixUp (Zhang et al., 2018) y Label Smoothing (Müller et al., 2019), y se observaron intervalos de confianza $< 30\%$ en imágenes con artefactos. Fue necesario incorporar calibración posterior (p. ej., temperature scaling) y estrategias de rechazo cuando la incertidumbre fue alta.

También se encontraron sesgos de datos debido a desequilibrios de clases y falta de diversidad (fototipos, dispositivos, condiciones de iluminación) que limitan la equidad del desempeño.

Además la utilización asistencial exige evidencia clínica, marcado CE (certificado europeo) y definición de responsabilidades médico-legales. El sistema debe ser complementario y nunca sustitutivo del juicio clínico del profesional.

Para poder lograr una transferencia efectiva, se necesitan integraciones con la historia clínica electrónica y mecanismos de registro de casos y soporte en centros de salud.

BIBLIOGRAFÍA

- Abid, A., Abdalla, A., Abid, A., Khan, D., & Zou, J. (2019). *Gradio: A platform for democratizing machine learning applications*. arXiv. <https://arxiv.org/abs/1906.02569>
- Chen, L., Lu, C., Hu, X., et al. (2022). *MedFormer: Medical vision transformer for improved clinical image analysis*. arXiv. <https://arxiv.org/abs/2206.XXXX>
- Codella, N. C. F., Rotemberg, V., Tschandl, P., et al. (2019). *Skin lesion analysis toward melanoma detection: A challenge at the 2018 International Skin Imaging Collaboration*. arXiv. <https://arxiv.org/abs/1902.03368>
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. (2021). *An image is worth 16×16 words: Transformers for image recognition at scale*. arXiv. <https://arxiv.org/abs/2010.11929>
- Müller, R., Kornblith, S., & Hinton, G. E. (2019). When does label smoothing help? *Advances in Neural Information Processing Systems*, 32, 4694–4703.
- Paszke, A., Gross, S., Massa, F., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 8024–8035.
- Selvaraju, R. R., Cogswell, M., Das, A., et al. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *In Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 618–626).
- SkinVision. (2025). *SkinVision – Skin cancer screening app*. <https://www.skinvision.com>
- Tschandl, P., Rosendahl, C., & Kittler, H. (2018). The HAM10000 dataset: A large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5, Article 180161. <https://doi.org/10.1038/sdata.2018.161>
- Unión Europea. (2017). *Reglamento (UE) 2017/745 del Parlamento Europeo y del Consejo relativo a los productos sanitarios*. Diario Oficial de la Unión Europea, L117, 1–175.
- VisualDx. (2025). *VisualDx clinical decision support*. <https://www.visualdx.com>
- Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). *MixUp: Beyond empirical risk minimization*. arXiv. <https://arxiv.org/abs/1710.09412>

ANEXOS

Figura 1. Pantalla inicial de SmartSkin.

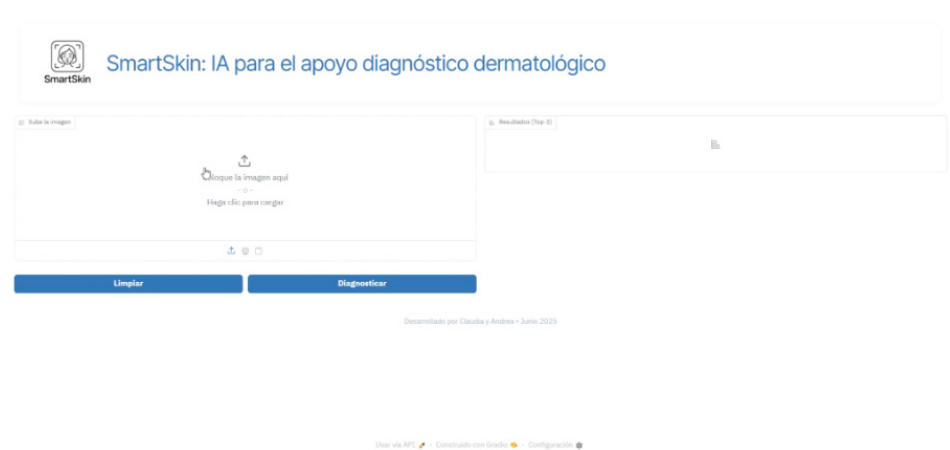
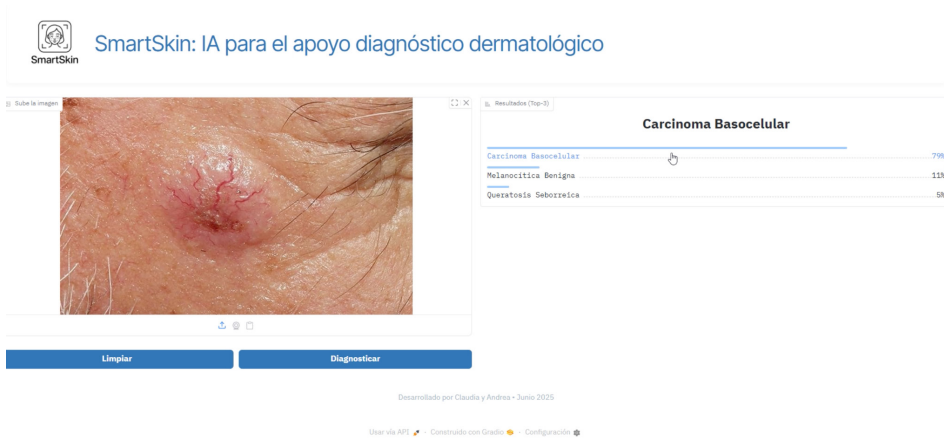


Figura 2. Funcionamiento de SmartSkin: ejemplo de salida (Top-3).



DIVA: DINERO VISUAL Y ACCESIBLE



AMPARO CASADO MELO

MARÍA JOSÉ MADRID MARTÍN

ALFREDO SÁNCHEZ-FUENTES RIVERO

RICARDO TRUJILLO ANTIVEROS

IRENE CALVO CRESPO

MAR SERRANO DEYÁ

IRIA VARGAS PÉREZ

RESUMEN

La visión es de todos los sentidos el que más información recibe, y aunque son muchos los avances en cirugía y tratamientos oftalmológicos en las últimas décadas, muchas personas conviven con la discapacidad visual o ceguera en nuestra sociedad.

Este proyecto presenta una aplicación denominada DIVA: Dinero Visual y Accesible, que con el uso de la inteligencia artificial favorece el reconocimiento de monedas y billetes para llevar a cabo algo tan cotidiano como, por ejemplo, ir a comprar, pagar en un establecimiento o realizar cualquier transacción monetaria. Y es que, aunque las personas con algún resto visual o las que no lo tienen exploran el mundo que les rodea, principalmente, con el reconocimiento táctil, en ocasiones resulta complicado identificar y discriminar algunas monedas y billetes.

Por este motivo consideramos que el desarrollo de este proyecto permite dar una mayor seguridad, rapidez y control del dinero a las personas con discapacidad visual y ceguera favoreciendo de este modo su competencia financiera.

En definitiva, DIVA integra tecnología avanzada con un propósito social significativo, impulsar la autonomía de las personas con discapacidad visual y mejorar su experiencia cotidiana al apoyarles al hacer uso del dinero de una manera segura.

PALABRAS CLAVE: Discapacidad visual, competencia financiera, calidad de vida, autonomía, inclusión.

1. CONTEXTUALIZACIÓN DEL PROYECTO

Son muchos los avances en cirugía y en los tratamientos oftalmológicos en las últimas décadas, pero convivir con la discapacidad visual o ceguera es un hecho en nuestra sociedad. Así mismo, según un informe del Instituto Clínico Quirúrgico de Oftalmología ICQO (2024) se calcula que el número de personas con discapacidad visual podría triplicarse en los próximos años debido al crecimiento de la población y a su envejecimiento. De ser así en el año 2050, según este informe, podría haber 115 millones de personas ciegas en el mundo en comparación con los 36 millones actuales.

Los trastornos visuales son un impedimento para desarrollar muchas de las actividades de la vida diaria. Cuando la patología es congénita el niño desde los primeros meses de vida debe realizar una conexión con el medio que le rodea muy diferente a cualquier otro niño vidente (ONCE, 2011). Cuando la pérdida de la vista es adquirida en cualquier momento vital, esto paraliza y obstaculiza muchos de los aprendizajes básicos como la lectura, la escritura, las posibilidades de moverse de forma autónoma y otras muchas tareas que con la vista se realizan de una forma automática.

Con respecto a la ceguera durante siglos se han mantenido muchos mitos y creencias que han estigmatizado y dañado a las personas que la padecían. Además, la falta de políticas públicas inclusivas ha sido una barrera clave en el acceso a la educación. Hasta mediados del siglo XX, la enseñanza para personas con discapacidad visual dependía principalmente de iniciativas privadas o de organizaciones religiosas, lo que generaba desigualdades significativas en la educación de esta población (Ipland y Parra, 2009; Herrero Ortín, 2015; Casado, 2022). Por este motivo, las personas carentes de visión han estado discriminadas durante siglos y vivían abocadas, en su mayoría, a la extrema pobreza. Por ejemplo, en la Edad Media se mantuvieron convicciones erróneas sobre ellas, rehuyendo su trato y se negándoseles el sustento y el derecho al trabajo. Poco a poco, se empezó a exaltar su capacidad de orientación, de memoria y la fina percepción sensorial que demostraban (Montoro, 1995).

A excepción de los más privilegiados, los privados de visión también lo estaban de la enseñanza y la educación con sus iguales. Será en 1857, cuando la Ley de Instrucción Pública recoja entre otras muchas cuestiones que la primera enseñanza se dará, con las modificaciones convenientes, a los ciegos en los establecimientos especiales que existan o que se crearán con este objeto.

Hasta principios del siglo XX, no encontramos muchas referencias de las personas con deficiencias visuales graves, porque se las consideraba ciegas y eran tratadas como tales, ya que la medicina y la óptica no proporcionaban herramientas para su mejoría y adaptación. A esto se sumaba otro problema persistente, la falta de capacitación del personal docente en estrategias pedagógicas inclusivas. Muchos maestros no contaban con formación específica para adaptar sus métodos de enseñanza a las necesidades de los estudiantes con discapacidad visual, dificultando así su integración en el aula (Ipland y Parra, 2009; Molina Ortín, 2018). La falta de sensibilización ha contribuido a que, en muchos casos, estos estudiantes hayan sido subestimados o excluidos de actividades académicas por considerarlas demasiado complejas para ellos.

La relación entre ceguera y dependencia ha sido una constante durante siglos en nuestra sociedad y a nivel internacional el panorama ha sido muy similar. Esta concepción negativista y la baja prevalencia (en números totales de la población) son las principales causas de que, hasta épocas muy recientes, no se hayan potenciado las posibilidades de las personas que la padecían. La situación actual, en el siglo XXI, es sustancialmente distinta a la presentada hasta ahora, pero aún queda mucho por hacer.

El colectivo de personas mayores con baja visión o ceguera, tienen unas necesidades muy específicas y apenas se dispone de propuestas de intervención dirigidas a aquellas personas que, debido a limitaciones funcionales, no pueden acceder a los tradicionales programas de rehabilitación (Díaz, 2008). Esto debe suponer, necesariamente, un reto para que desde instituciones como la Universidad se trabaje aportando herramientas útiles que puedan utilizar en su vida diaria.

Por todo ello, se presenta el proyecto DIVA que se justifica desde la necesidad de desarrollar una herramienta para favorecer la accesibilidad y el conocimiento del contexto de las personas con discapacidad visual, facilitando en su día a día el uso del dinero de una manera segura y autónoma. El objetivo principal de la misma ha sido diseñar y elaborar una aplicación móvil que permite discriminar los diferentes billetes y monedas de la eurozona.

Para la creación de este proyecto se han tenido en cuenta dos perspectivas imprescindibles; el enfoque educativo social y el enfoque tecnológico. La visión es considerada el sentido que más información recibe del entorno, ya que alrededor del 80% de la información que nuestro cerebro procesa es visual, permitiéndonos percibir formas, colores, distancias y movimientos. Este sentido es fundamental para la supervivencia, la interacción con el mundo y la realización de actividades diarias como trabajar, estudiar o poder comprar cualquier objeto o elemento de nuestro contexto más cercano.

DIVA se centra en la importancia de discriminar y conocer las monedas y los billetes para llevar a cabo transacciones monetarias, por ejemplo, algo tan cotidiano como ir de compras o pagar en un establecimiento. Aunque las personas con resto visual o las que carecen de él, a lo largo de su vida trabajan la estimulación háptica y el reconocimiento táctil, cuando se trata de monedas o de billetes en ocasiones esto

no es suficiente para poder determinar el valor del mismo, por estar deteriorados, carecer del suficiente tiempo para su discriminación, etc. Por eso consideramos que el desarrollo de este proyecto permite dar una mayor seguridad, rapidez y control del dinero a las personas con esta discapacidad.

Para ello, DIVA integra tecnología avanzada con un propósito social significativo, impulsando la autonomía de las personas con discapacidad visual y mejorando su experiencia cotidiana con el manejo del dinero.

2. JUSTIFICACIÓN Y OBJETIVOS

La importancia de la educación financiera es innegable en nuestra sociedad, tal y como indican diversos investigadores (Del Brío, et al. (2015)). Dentro de los contenidos de la competencia financiera establecidos por la OECD (2014) se incluye el área de dinero y transacciones, esta categoría representa el primer contenido básico de la alfabetización financiera, e incluye el conocimiento de las diferentes formas y finalidades del dinero y el manejo de transacciones monetarias sencillas como los pagos cotidianos, los gastos, el valor del dinero, etc.

También, el Real Decreto 157/2022, de 1 de marzo, por el que se establecen la ordenación y las enseñanzas mínimas de la Educación Primaria incluye entre los saberes básicos para el área de Matemáticas dentro del bloque del Sentido numérico, la Educación financiera.

Este proyecto se alinea además con los Objetivos de Desarrollo Sostenible (Naciones Unidas, s.f.), particularmente juegan un papel importante los siguientes:

- 10. Reducción de las desigualdades: el proyecto promueve la inclusión social y económica de las personas, facilitando el avance de las personas con discapacidad en la sociedad, eliminando barreras, dificultades, desigualdades y los estereotipos que puedan surgir.
- 4. Educación de calidad: es necesario centrarse en garantizar una educación de calidad, inclusiva y equitativa para todos. Por ello, es importante el planteamiento de proyectos que ayuden a la inclusión de personas con discapacidad en la sociedad.

3. OBJETIVOS

El objetivo principal de este proyecto es desarrollar e implementar la aplicación educativa DIVA mediante la identificación de las monedas y billetes de la zona euro utilizando inteligencia artificial y facilitando así la autonomía y seguridad en las transacciones monetarias de las personas con discapacidad visual.

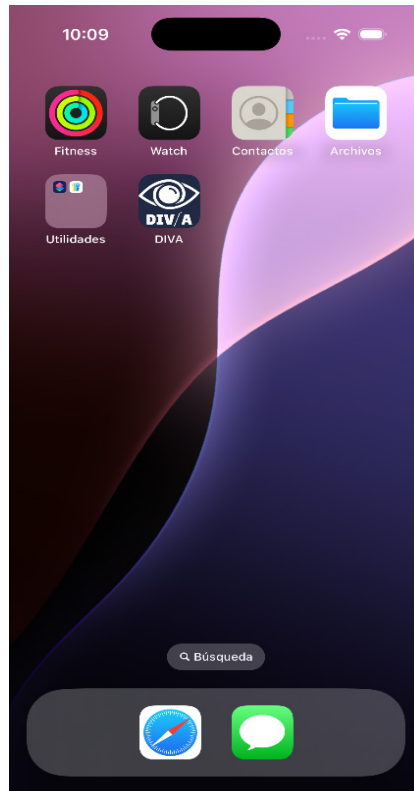
3.1. Objetivos específicos

Los objetivos específicos del proyecto llevado a cabo son:

- Desarrollar una aplicación que permita a las personas con discapacidad visual reconocer y diferenciar billetes y monedas de la zona euro de manera rápida y precisa.
- Implementar tecnologías de reconocimiento de imágenes e inteligencia artificial para garantizar un alto grado de precisión en la identificación del dinero de la zona euro.
- Garantizar la seguridad y autonomía de los usuarios, reduciendo el riesgo de confusión o fraude en transacciones monetarias.
- Promover la inclusión social, la accesibilidad y la competencia financiera, ofreciendo una herramienta para personas con discapacidad visual y ceguera.

4. DISEÑO, DESARROLLO E IMPLEMENTACIÓN

El principal resultado obtenido es la app móvil DIVA, una herramienta sencilla que nos permite el reconocimiento e identificación de monedas y billetes de la zona euro. Esta se pondrá a disposición de toda la comunidad de personas con discapacidad visual para favorecer su autonomía y accesibilidad, mediante un recurso seguro que favorezca la gestión autónoma de este aspecto de sus finanzas. La Figura 1 muestra el acceso a la app desde un Smartphone.

Figura 1. Acceso a DIVA.

La app se crea utilizando varias tecnologías como el reconocimiento de imágenes mediante inteligencia artificial, para obtener así un alto grado de precisión y fiabilidad.

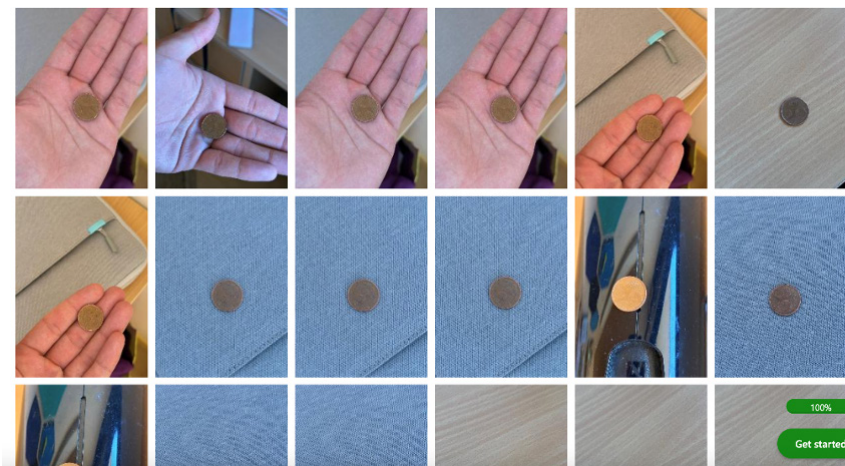
Para el diseño de la aplicación se ha pasado por varias etapas. En primer lugar, la fase de diseño se caracterizó por la simplicidad y la usabilidad. Se priorizó una interfaz de usuario minimalista y altamente intuitiva, con un enfoque en la interacción mediante un simple toque en la pantalla. Se realizaron prototipos tempranos y pruebas con usuarios de la Asociación de Retinosis Pigmentaria de Salamanca, para asegurar que la navegación y la funcionalidad fueran comprensibles y eficientes.

La compatibilidad total con VoiceOver (iOS) y TalkBack (Android) fue una consideración fundamental desde el inicio, integrando los elementos de accesibilidad de forma nativa en el diseño de la arquitectura de la aplicación.

Posteriormente, el desarrollo de DIVA se basó en el uso de tecnologías de inteligencia artificial de vanguardia para el reconocimiento de imágenes. Se entrenó un modelo de IA robusto y optimizado para la identificación precisa de billetes y monedas en diversas condiciones de iluminación y orientación. Esto implicó la reco-

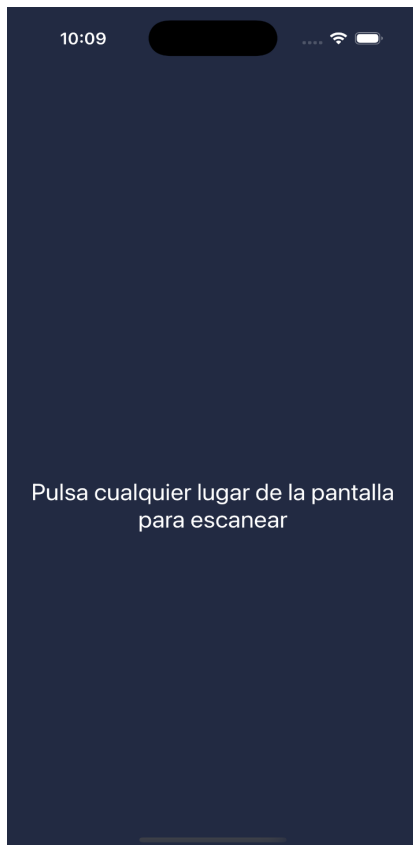
pilación y el procesamiento de un vasto conjunto de datos de imágenes de diferentes denominaciones y monedas (Figura 2). La seguridad y la privacidad de los usuarios fueron aspectos clave durante todo el desarrollo, asegurando que no se almacenará ninguna información personal ni imágenes capturadas.

Figura 2. Algunas de las imágenes incluidas para alimentar el modelo.



La tercera fase fue la de implementación de la aplicación, se llevó a cabo considerando la facilidad de acceso y distribución. La aplicación se desarrolló para plataformas iOS y Android para favorecer el mayor alcance de los futuros usuarios. La meta fue no solo lanzar una aplicación funcional, sino también establecer una plataforma que pudiera evolucionar y adaptarse a las necesidades cambiantes de la comunidad. En la Figura 3 se muestra un ejemplo de funcionamiento de la app, destacando su simplicidad y accesibilidad.

Figura 3. Captura de la pantalla que solicita la app para escanear la moneda.



La aplicación DIVA se caracteriza dentro del sector de las tecnologías accesibles por sus atributos que la posicionan como una herramienta innovadora e inclusiva para personas con discapacidad visual. A continuación, se enumeran los principales hitos:

- Reconocimiento por voz e imagen: DIVA integra tecnologías avanzadas de reconocimiento visual e inteligencia artificial, que permiten identificar de forma precisa monedas y billetes. Una vez detectada la moneda o billete, la app comunica el valor mediante un sistema de voz accesible.
- Diseño centrado en el usuario: La interfaz ha sido diseñada específicamente para personas con discapacidad visual, con un solo botón central, navegación intuitiva y compatibilidad con lectores de pantalla como VoiceOver o TalkBack.
- Enfoque en la seguridad: DIVA evita fraudes o errores durante transacciones económicas, incrementando la confianza de los usuarios en este tipo de transacciones que realizan en su vida diaria.

- Aplicación específica: A diferencia de otras aplicaciones más genéricas, DIVA se centra exclusivamente en este aspecto cotidiano clave que es el reconocimiento de dinero. Esto la convierte en una herramienta práctica, funcional y de uso frecuente.
- Alta accesibilidad: La aplicación ha sido diseñada siguiendo criterios de accesibilidad universal, garantizando el uso efectivo tanto para personas con ceguera total como con baja visión.

DIVA responde a diversas necesidades de las personas con discapacidad visual y su entorno como son:

- La identificación autónoma del dinero: facilita el reconocimiento de monedas y billetes de la eurozona, permitiendo a las personas usuarias manejar su dinero de forma segura e independiente.
- Prevención de fraudes y errores: Reduce el riesgo de ser víctimas de engaños al realizar pagos o recibir cambios de monedas y billetes.
- Incremento de la seguridad y la confianza personal: Al promover la independencia en una actividad diaria como el uso del dinero, mejora la autoestima y la seguridad en sí mismos.
- Inclusión social: Permite una participación más activa en actividades económicas y sociales, rompiendo barreras de exclusión social.
- Apoyo educativo: Puede utilizarse como herramienta de enseñanza en el aula o en casa para que niños, niñas y jóvenes con discapacidad visual aprendan a identificar monedas y billetes progresivamente.
- Mejora de la calidad de vida: Al satisfacer una necesidad básica de manera autónoma, DIVA contribuye significativamente al bienestar y la integración de las personas en su entorno.

Gracias a este uso de tecnología, consideramos que DIVA provoca un gran impacto social, ya que contribuye a una mayor calidad de vida para todas aquellas personas con discapacidad visual, promoviendo una sociedad más inclusiva y equitativa.

5. CONCLUSIONES

La aplicación DIVA ha cumplido con sus objetivos principales, creando una app sencilla y útil que permite dar accesibilidad a las personas con discapacidad visual para poder tener control y manejo del dinero.

Entre los principales resultados se destaca la aplicación que ha sido llevada a cabo mediante las tres fases del diseño, desarrollo e implementación con usuarios con baja visión y ceguera.

El uso de la inteligencia artificial con fines sociales permite apoyar en su autonomía a las personas gracias a la identificación y diferenciación de las diferentes monedas y billetes de la zona euro.

La implementación de DIVA y las demostraciones con personas con ceguera y con retinosis pigmentaria ha permitido mejorar algunas fases del diseño para facilitar a su vez la accesibilidad de la aplicación en el móvil.

Este proyecto es una muestra más de que el desarrollo que proporciona la tecnología puede ser considerado un factor revolucionario para las personas con discapacidad visual, favoreciendo de esta manera la inclusión e interacción sin dificultad en el ámbito educativo, profesional y social.

Aplicaciones como estas permiten mejorar la calidad de vida de las personas con discapacidad al favorecer la autonomía, independencia y seguridad, por este motivo reivindicamos el uso de la tecnología para abrir puertas hacia la inclusión (Zamora López, y Marín Perabá, 2021).

La aplicación DIVA puede estar considerada como una tiflotecnología avanzada y diferenciada, ya que esta se puede determinar como aquellos procedimientos y técnicas que se adaptan para poder usarlas (Doménech, 2010).

De esto modo, DIVA está considerada como una aplicación accesible e inclusiva que permite ayudar a la comunidad con discapacidad visual y favoreciendo su interacción con el entorno en su día a día.

La experiencia de poder diseñar, desarrollar e implementar una aplicación como la mostrada en este capítulo, pone de relevancia la importancia de crear propuestas tecnológicas desde un enfoque accesible e inclusivo. Más allá de las funcionalidades técnicas, se corrobora que representa un avance significativo hacia la justicia social y la mayor calidad de vida. El éxito de esta iniciativa tanto a nivel de la eficacia funcional como en las aportaciones en la sociedad, pone de manifiesto la necesidad de seguir desarrollando y haciendo uso de estas soluciones tecnológicas. Esta aplicación se determina como una herramienta clave en la construcción de una sociedad más equitativa, accesible y comprometida con la diversidad.

BIBLIOGRAFÍA

- Casado Melo, A. (2022). Los establecimientos para la educación de los niños ciegos, sordos y para la infancia anormal en España en el siglo XIX y principios del siglo XX. En A. J. Martínez González y F. J. Tejada Hernández (Eds.), *Etnicidad, identidad y ciudadanía: Las sociedades de ayer y hoy* (pp. 16-34). Dykinson.
- Del Brío, E., López, C. y Vereas, C. (2015). Educación financiera en la infancia. Propuesta didáctica en Educación Infantil. *ENSAYOS, Revista de la Facultad de Educación de Albacete*, 30(2), 99-122.

- Díaz, P. (2008). Discapacidad visual y autonomía: las posibilidades de las personas mayores. *Integración: revista sobre ceguera y discapacidad visual*, 55, 26-33.
- Doménech, X. (2010). Historia de la Tiflotecnología en España. *No solo usabilidad*, (9).
- Herrero Ortín, T. M. (2015). *La educación inclusiva del alumnado con discapacidad visual en la Comunidad Valenciana: análisis y perspectivas*. Universidad de Alicante.
- ICQO (2024). *Investigadores españoles revelan que el 80% de casos de discapacidad visual y ceguera se podrían evitar o curar*. <https://icqo.org/2024/10/10/evitar-la-discapacidad-visual-con-revision/>
- Ipland García, J., y Parra Cañadas, D. (2009). La formación de ciegos y discapacitados visuales: Visión histórica de un proceso de inclusión. En M. R. Berruero Albéniz y S. Conejero López (coords.), *El largo camino hacia una educación inclusiva: la educación especial y social del siglo XIX a nuestros días: XV Coloquio de Historia de la Educación, Pamplona-Iruñea, 29, 30 de junio y 1 de julio de 2009* (pp. 453-462). Universidad Pública de Navarra= Nafarroako Unibertsitate Publikoa.
- Ley de Instrucción pública autorizada por el Gobierno para que rija desde su publicación en la Península é Islas adyacentes, lo que se cita. *Gaceta de Madrid, núm. 1710, de 10 de septiembre de 1857*.
- Molina Ortín, J. L. (2018). *Historia y evolución de la educación para personas con discapacidad visual en España*. Editorial Académica Española.
- Montoro, J. (1995). *Los ciegos en la historia: Vol. IV*. Organización Nacional de Ciegos Españoles.
- Naciones Unidas. (s. f.) *Objetivos de Desarrollo Sostenible*. <https://www.un.org/sustainabledevelopment/es/>
- OECD. (2014). *PISA 2012 results: Students and money (volume VI)*. OECD.
- ONCE. (2011). *Discapacidad visual y autonomía personal Enfoque práctico de la rehabilitación*. Organización Nacional de Ciegos Españoles (ONCE).
- Real Decreto 157/2022, de 1 de marzo, por el que se establecen la ordenación y las enseñanzas mínimas de la Educación Primaria. *BOE*, 52, de 2 de marzo de 2022, 24386-24504.
- Zamora López, P. y Marín Perabá, C. (2021). Tiflotecnologías para el alumnado con discapacidad visual. *Academo*, 8(1), 109-118.

PSYCHE ASISTENTE VIRTUAL PARA EL PSICÓLOGO BASADO EN INTELIGENCIA ARTIFICIAL



DIEGO TABERNERO MONTERO

ESTEBAN RISTORI FERNÁNDEZ

CARLOS MARTÍN VELÁZQUEZ

MIGUEL GUTIÉRREZ HERNÁNDEZ

MANUEL MARTÍN-MERINO ACERA

RESUMEN

El proyecto PSYCHE consiste en el desarrollo de una herramienta de apoyo para psicólogos que integra tecnologías de inteligencia artificial para automatizar y optimizar tareas clínicas fundamentales. La aplicación web, desarrollada con Gradio, permite la transcripción automática de sesiones terapéuticas, la generación de diarios de sesión estructurados, el análisis cronológico de historias clínicas y la implementación de un sistema de triaje automático para la clasificación de pacientes según su nivel de urgencia. Utilizando modelos de lenguaje avanzados como Google Gemini y tecnologías de reconocimiento de voz, PSYCHE logra una precisión del 84,85% en la transcripción de audio y proporciona un sistema completo de gestión clínica que reduce significativamente la carga administrativa del profesional de la psicología mientras mejora la calidad y precisión de la documentación clínica.

1. CONTEXTUALIZACIÓN DEL PROYECTO

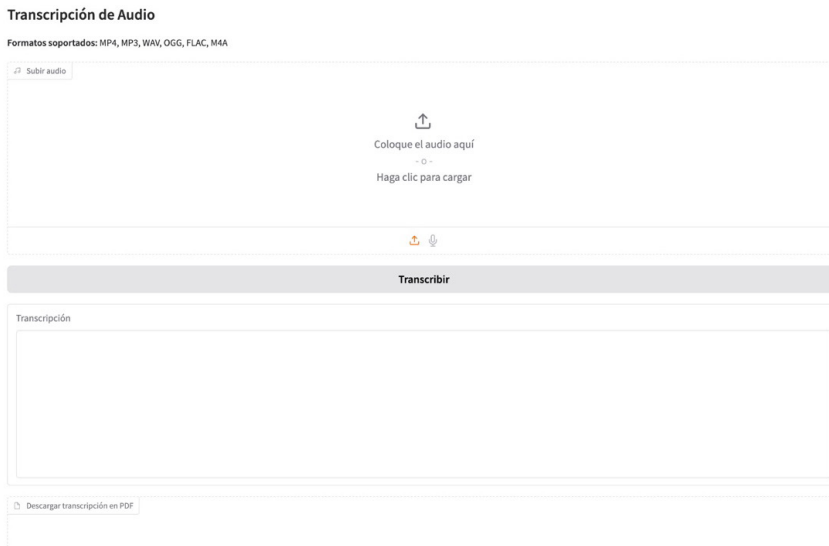
El proyecto PSYCHE se desarrolló durante el curso académico 24 25 en el marco del Club Universitario de Innovación CUI) de la Universidad Pontificia de Salaman-

ca, dentro de la convocatoria TALENT. El equipo multidisciplinar estuvo formado por estudiantes de ingeniería informática bajo la supervisión del profesor Manuel Martín-Merino Acera.

La contextualización de este proyecto surge de la identificación de necesidades reales en el ámbito de la psicología clínica, donde los profesionales dedican una cantidad considerable de tiempo a tareas administrativas y de documentación que, aunque esenciales, reducen el tiempo disponible para la atención directa al paciente. La carga de trabajo relacionada con la elaboración de informes, diarios de sesión, historias clínicas y procesos de triaje representa un desafío significativo en la práctica psicológica contemporánea.

En este contexto, la aplicación de tecnologías de inteligencia artificial en el ámbito sanitario ha demostrado un potencial transformador, especialmente en la automatización de procesos repetitivos y en la asistencia a la toma de decisiones clínicas. El proyecto se alinea con las tendencias actuales de digitalización sanitaria y la búsqueda de soluciones tecnológicas que mejoren tanto la eficiencia operativa como la calidad asistencial.

Ilustración 1. Interfaz de transcripción de audio de PSYCHE



2. JUSTIFICACIÓN Y OBJETIVOS

2.1. Justificación

La justificación del proyecto PSYCHE se fundamenta en tres pilares principales. En primer lugar, la **optimización del tiempo clínico** constituye una necesidad imperante en la práctica psicológica. Los profesionales invierten entre un 30-40% de su tiempo laboral en tareas administrativas, reduciendo significativamente el tiempo disponible para la intervención terapéutica directa. La automatización de estos procesos mediante inteligencia artificial permite redistribuir este tiempo hacia actividades de mayor valor clínico.

En segundo lugar, la **mejora en la precisión y consistencia de la documentación clínica** representa un beneficio crucial. La transcripción manual de sesiones y la elaboración de informes están sujetas a errores humanos, omisiones y variabilidades en el formato. Un sistema automatizado garantiza la captura sistemática de información relevante y mantiene la consistencia en la documentación a lo largo del tiempo.

Finalmente, la **estandarización del proceso de triaje** mejora la equidad y eficacia en la atención. Los sistemas de triaje automático basados en criterios clínicos establecidos reducen la subjetividad en la asignación de prioridades y garantizan que los casos más urgentes reciban atención prioritaria de manera consistente.

2.2. Objetivos

El **objetivo principal** del proyecto es desarrollar una aplicación web basada en inteligencia artificial que facilite la carga de trabajo del psicólogo relacionada con la documentación clínica y la gestión organizacional, así como la priorización de pacientes mediante un sistema de triaje automatizado.

Los **objetivos específicos** incluyen:

- **Implementar un sistema de reconocimiento de voz** para la transcripción automática de sesiones terapéuticas, con soporte para múltiples formatos de audio MP4, MP3, WAV, OGG, FLAC, M4A.
- **Desarrollar un generador de diarios de sesión** que utilice inteligencia artificial para estructurar automáticamente la información relevante de cada sesión, incluyendo temas discutidos, objetivos, reacciones del paciente y recomendaciones terapéuticas.
- **Crear un sistema de análisis cronológico** que genere historias clínicas organizadas temporalmente, identificando factores predisponentes, precipitantes y mantenedores.

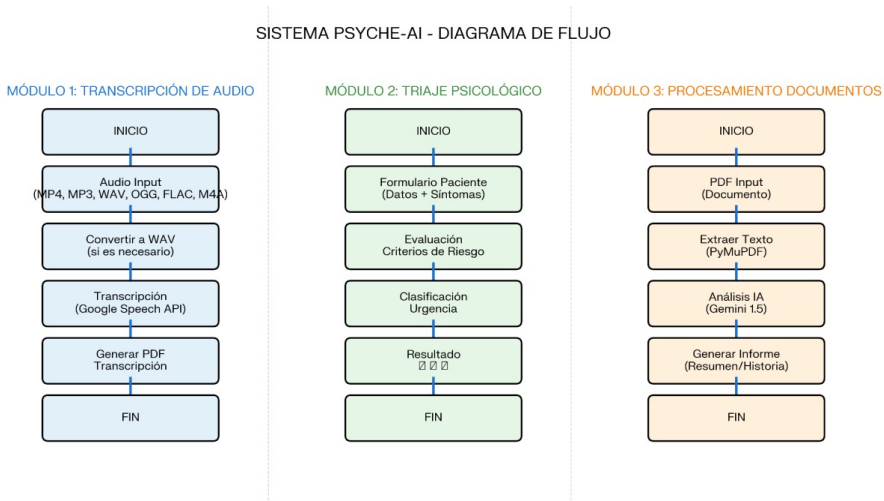
- **Diseñar un sistema de triaje automático** basado en un cuestionario estructurado que clasifique a los pacientes según su nivel de urgencia clínica mediante un sistema de colores (rojo-amarillo-verde).
- **Desarrollar una interfaz web intuitiva que permita tanto a psicólogos como a pacientes** utilizar el sistema de manera eficiente, con funcionalidades diferenciadas para cada tipo de usuario.

3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN

3.1. Arquitectura del sistema

El sistema PSYCHE se desarrolló siguiendo una arquitectura modular que separa claramente las funcionalidades en componentes especializados. La aplicación utiliza **Gradio** como framework principal para la interfaz web, proporcionando una experiencia de usuario intuitiva y responsiva. Esta elección tecnológica permite el despliegue local sin necesidad de configuraciones complejas de servidor.

Ilustración 2. Diagrama de flujo del sistema PSYCHE



La arquitectura se estructura en cinco módulos principales:

- **Módulo de interfaz principal** (interfacecosas.py): Gestiona la transcripción de audio y el procesamiento de documentos PDF.
- **Módulo de triaje** (interfacetriage.py): Implementa el sistema de clasificación de pacientes.

- **Módulo de análisis cronológico** (historiacrono.py): Genera historias clínicas estructuradas temporalmente.
- **Módulo de resúmenes** (resumen.py): Produce diarios de sesión automatizados.
- **Módulo de utilidades** (chatutils.py): Proporciona funciones auxiliares para el procesamiento conversacional.

4. IMPLEMENTACIÓN TECNOLÓGICA

4.1. Sistema de transcripción de audio

```
def convert_to_wav(audio_path):
    try:
        temp_wav = tempfile.NamedTemporaryFile(suffix='.wav', delete=False)
        temp_wav_path = temp_wav.name
        temp_wav.close()

        audio = AudioSegment.from_file(audio_path)
        audio.export(temp_wav_path, format="wav")
        return temp_wav_path
    except Exception as e:
        return None
```

El componente de transcripción utiliza **SpeechRecognition** con la API de Google Speech-to-Text, alcanzando una precisión del 84,85% en la transcripción de audio en español. El sistema incluye una función de conversión automática (`convert_to_wav`) que utiliza **pydub** para transformar archivos en formatos no compatibles MP4, M4A, AAC, FLAC, OGG) a formato WAV, garantizando la compatibilidad universal.

4.2. Sistema de inteligencia artificial

El procesamiento de lenguaje natural se basa en **Google Gemini 1.5 Flash** integrado a través de **LangChain**. La configuración del modelo incluye parámetros optimizados para el contexto clínico:

- Temperatura: 0.7 (balance entre coherencia y creatividad).
- Máximo de tokens: 1024.
- Configuración de seguridad deshabilitada para contenido clínico sensible.
- Rate limiting: 0.1 requests/second para optimizar el uso de API.
- 5.3 Sistema de triaje automático

- La implementación del triaje se basa en una lógica de clasificación jerárquica que evalúa indicadores clínicos según protocolos establecidos:
 - **Nivel Rojo Urgencia Alta):** Ideación suicida, intentos previos, síntomas psicóticos, violencia
 - **Nivel Amarillo Urgencia Media):** Depresión severa, duelo/crisis
 - **Nivel Verde Urgencia Baja):** Síntomas leves sin riesgo inmediato

Ilustración 3. Sistema de triaje psicológico con ejemplo de clasificación

Triaje Psicológico

Clasificación de urgencia psicológica con base en síntomas y riesgo

Nombre completo	Resultado del Triaje
Miguel Palomares Montero	Nombre: Miguel Palomares Montero
Edad: 21	Edad: 21 años Género: Masculino
Género: <input checked="" type="radio"/> Masculino	Primera vez en terapia: Sí
¿Es tu primera vez en terapia? <input checked="" type="radio"/> Sí	Motivo de consulta: Cansancio Duración de síntomas: 1-6 meses
Motivo principal (máximo 3 palabras clave): Cansancio	Medicación actual: Ninguno
Duración de síntomas: <input checked="" type="radio"/> 1-6 meses	Urgencia asignada: ALTA ● Rojo - Derivación urgente
¿Tomas medicación actualmente? <input checked="" type="radio"/> No	Observaciones:
Especifica el medicamento (si aplica)	Marcar
<input type="checkbox"/> ¿Me gustaría recibir artículos o recursos relacionados?	

5. FUNCIONALIDADES IMPLEMENTADAS

5.1. Generación de diarios de sesión

El módulo de resúmenes (resumen.py) utiliza prompts especializados para generar diarios estructurados que incluyen:

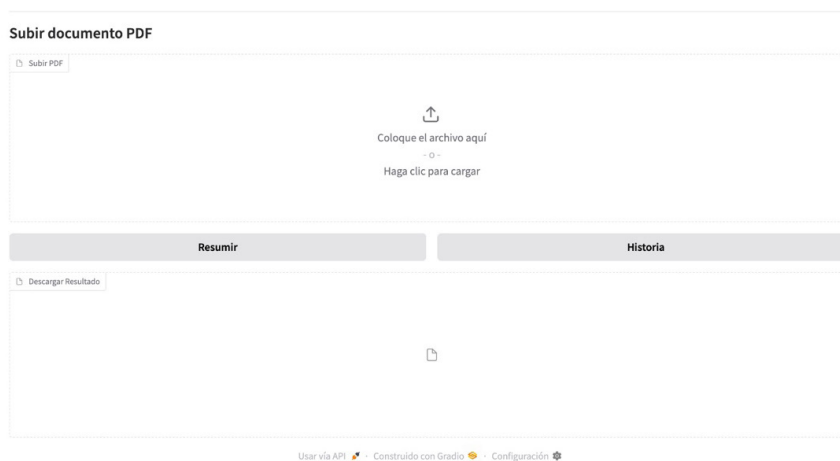
- **Temas discutidos:** Identificación automática de los temas principales abordados
- **Objetivos de la sesión:** Extracción de metas terapéuticas específicas
- **Reacciones del paciente:** Análisis de respuestas emocionales y comportamentales
- **Recomendaciones:** Compilación de sugerencias terapéuticas proporcionadas
- **Preguntas futuras:** Generación de preguntas relevantes para próximas sesiones

5.2. Análisis cronológico

El sistema de historia cronológica (historiacrono.py) analiza las transcripciones para identificar y organizar temporalmente:

- **Factores predisponentes:** Antecedentes personales o familiares relevantes
- **Factores precipitantes:** Eventos específicos que desencadenaron el problema
- **Factores mantenedores:** Condiciones actuales que perpetúan la situación
- **Cronología de eventos:** Ordenamiento temporal de acontecimientos significativos

Ilustración 4. Interfaz de procesamiento de documentos PDF



5.3. Gestión de archivos y seguridad

El sistema implementa un manejo seguro de archivos temporales para el procesamiento de audio y documentos. Los archivos se crean en ubicaciones temporales seguras y se eliminan automáticamente tras su procesamiento, garantizando la privacidad de los datos clínicos. La configuración de variables de entorno (.env) protege las claves API y otros datos sensibles.

La aplicación incluye autenticación básica (auth=(“user”;1234”)) que, aunque simple para el prototipo, establece las bases para implementar sistemas de autenticación más robustos en entornos de producción.

6. CONCLUSIONES Y REFLEXIONES FINALES DE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO

6.1. Principales logros alcanzados

El proyecto PSYCHE ha demostrado la viabilidad y efectividad de integrar tecnologías de inteligencia artificial en el ámbito de la psicología clínica. Los resultados obtenidos confirman que es posible automatizar significativamente las tareas administrativas sin comprometer la calidad de la atención clínica.

El **logro más destacado** es la consecución de una precisión del 84,85% en el reconocimiento de voz para transcripciones en español, una cifra que se sitúa dentro de rangos aceptables para uso clínico cuando se complementa con revisión profesional. Esta precisión, combinada con la capacidad de procesamiento de múltiples formatos de audio, hace que el sistema sea prácticamente aplicable en entornos clínicos reales.

La **generación automática de diarios de sesión** ha mostrado resultados prometedores en la estructuración de información clínica relevante. La capacidad del sistema para identificar temas discutidos, objetivos terapéuticos y reacciones del paciente proporciona un framework consistente para la documentación clínica, reduciendo la variabilidad entre profesionales y mejorando la calidad general de los registros.

El **sistema de triaje automático** representa una innovación significativa en la gestión de flujos de pacientes. La clasificación basada en criterios clínicos objetivos reduce la subjetividad en la priorización y garantiza que los casos más urgentes reciban atención prioritaria de manera sistemática.

6.2. Limitaciones identificadas

A pesar de los logros alcanzados, el proyecto presenta varias **limitaciones técnicas** que deben abordarse para una implementación completa. La precisión del reconocimiento de voz, aunque aceptable, requiere supervisión y corrección manual, especialmente en casos de audio con calidad deficiente o terminología técnica compleja.

Las **limitaciones en el procesamiento de lenguaje natural** incluyen la dependencia de modelos externos (Google Gemini), lo que implica costos operativos continuos y dependencia de conectividad a internet. Además, aunque el sistema maneja bien el español, podría presentar dificultades con acentos regionales marcados o jerga específica.

Desde una perspectiva **ética y de privacidad**, el proyecto requiere el desarrollo de protocolos más robustos para el manejo de datos sensibles. La implementación actual, aunque funcional para pruebas, no cumple completamente con regulaciones como el RGPD para datos sanitarios, lo que limitaría su despliegue en entornos clínicos reales.

6.3. Reflexiones sobre la transferencia del conocimiento

La **transferencia de este conocimiento al sector sanitario** enfrenta varios desafíos significativos. En primer lugar, la resistencia al cambio tecnológico en entornos clínicos tradicionales requiere estrategias de formación y acompañamiento específicas. Los profesionales de la psicología necesitan percibir el valor añadido del sistema sin sentir que su expertise clínico está siendo reemplazado.

La **integración con sistemas existentes** representa otro desafío importante. Los centros de salud y clínicas privadas utilizan diversos sistemas de gestión clínica, y la interoperabilidad se convierte en un requisito crítico para la adopción exitosa de la tecnología.

Desde una perspectiva **regulatoria**, la implementación de sistemas de IA en entornos clínicos requiere certificaciones y validaciones específicas que van más allá del alcance del prototipo actual. La necesidad de cumplir con normativas sanitarias, protección de datos y estándares de calidad implica un desarrollo adicional significativo.

6.4. Potencial de escalabilidad y continuidad

El proyecto PSYCHE presenta un **alto potencial de escalabilidad** debido a su arquitectura modular y el uso de tecnologías estándar. La posibilidad de integrar nuevos módulos especializados (gestión de citas, análisis de tendencias, alertas automáticas) hace que el sistema pueda evolucionar según las necesidades específicas de diferentes entornos clínicos.

Las **mejoras futuras** identificadas incluyen la implementación de modelos de IA especializados en terminología psicológica, el desarrollo de capacidades multilingües, la integración con dispositivos IoT para monitorización de pacientes y la incorporación de análisis predictivo para identificar patrones de riesgo.

La **sostenibilidad económica** del proyecto requiere un modelo de negocio que equilibre los costos de desarrollo y mantenimiento con la accesibilidad para profesionales independientes y centros pequeños. El modelo SaaS (Software as a Service) podría proporcionar una vía viable para la comercialización y sostenibilidad a largo plazo.

7. IMPACTO ESPERADO

El impacto potencial de PSYCHE trasciende la mera automatización de tareas administrativas. La estandarización de procesos de documentación puede contribuir significativamente a la **mejora de la calidad asistencial** al garantizar que información clínica crítica no se pierda y que los tratamientos mantengan consistencia a lo largo del tiempo.

Desde una perspectiva de **salud pública**, un sistema como PSYCHE podría facilitar la recopilación de datos epidemiológicos sobre trastornos mentales, con-

tribuyendo a una mejor comprensión de patrones de prevalencia y efectividad de tratamientos a nivel poblacional.

La **democratización del acceso** a herramientas avanzadas de gestión clínica podría beneficiar especialmente a profesionales en áreas rurales o con recursos limitados, reduciendo las disparidades en la calidad de atención psicológica disponible en diferentes contextos geográficos y socioeconómicos.

En conclusión, aunque PSYCHE presenta limitaciones que requieren desarrollo adicional para su implementación completa en entornos clínicos, el proyecto ha demostrado exitosamente la viabilidad de integrar inteligencia artificial en el flujo de trabajo psicológico, estableciendo una base sólida para futuras innovaciones en este campo crítico de la atención sanitaria.

BIBLIOGRAFÍA

- Google Cloud. Speech-to-Text API documentation. <https://cloud.google.com/speech-to-text/docs>
- LangChain Community. LangChain framework for LLM applications. https://python.langchain.com/docs/get_started/introduction
- Aboulenein A, Baranchuk A. Speech recognition technology in medicine: applications and benefits. *J Med Internet Res* 2022;24 3 e32871.
- Mitchell TM. Machine learning approaches in healthcare: opportunities and challenges. *Nat Med* 2021;27 4 518 527.
- American Psychological Association. Guidelines for computer-based tests and interpretations. 2019.
- Chen L, Zhang H, Wu J. Artificial intelligence in clinical psychology: a systematic review. *Clin Psychol Rev* 2023;89 102087.

TURNOFÁCIL. CHATBOT INTELIGENTE PARA OPTIMIZACIÓN DE RESERVAS Y CITAS



ANA MARÍA FERMOSE GARCÍA

JUAN RAMÓN MARTÍN SAN ROMÁN

PAULA ISABEL JUANES GIL

RESUMEN

El proyecto TurnoFácil presenta un asistente conversacional diseñado para optimizar la gestión de reservas y citas en organismos públicos y privados. Partiendo del caso de uso de REGTSA (Diputación de Salamanca), el sistema responde a la necesidad de reducir el elevado volumen de llamadas telefónicas destinadas exclusivamente a concertar o modificar citas. TurnoFácil combina inteligencia artificial, mensajería instantánea y bases de datos distribuidas para ofrecer al ciudadano una experiencia fluida, inclusiva y accesible a través de WhatsApp y una interfaz web complementaria.

La solución se estructura en cinco capas: interacción con el usuario, orquestación, inteligencia artificial, datos y disponibilidad, lo que facilita su modularidad y escalabilidad. El prototipo fue validado en las jornadas de proyectos TALENT, con la participación de alumnado, profesorado y personal técnico de REGTSA, confirmando su utilidad y pertinencia. Aunque su transferencia inmediata a la práctica institucional no se ha concretado, el sistema se plantea como una herramienta adaptable a múltiples sectores (sanidad, educación, servicios sociales, sector privado), con potencial de convertirse en un servicio SaaS replicable en distintos contextos.

PALABRAS CLAVE: Asistente, Gestión de citas, Inteligencia artificial, WhatsApp, SaaS.

ABSTRACT: TurnoFácil presents a conversational assistant designed to optimise the management of reservations and appointments in public and private organisa-

tions. Based on the use case of REGTSA (Salamanca Provincial Council), the system responds to the need to reduce the high volume of telephone calls made exclusively to make or change appointments. TurnoFácil combines artificial intelligence, instant messaging and distributed databases to offer citizens a fluid, inclusive and accessible experience via WhatsApp and a complementary web interface.

The solution is structured in five layers: user interaction, orchestration, artificial intelligence, data and availability, which facilitates its modularity and scalability. The prototype was validated at the TALENT project conference, with the participation of students, teachers and technical staff from REGTSA, confirming its usefulness and relevance. Although its immediate transfer to institutional practice has not yet been finalised, the system is intended to be a tool that can be adapted to multiple sectors (health, education, social services, private sector), with the potential to become a SaaS service that can be replicated in different contexts.

KEYWORDS: Assistant, Appointment management, Artificial intelligence, WhatsApp, SaaS.

1. CONTEXTUALIZACIÓN DEL PROYECTO

El proyecto TurnoFácil se enmarca en la transformación digital de la administración pública y en la mejora de la atención ciudadana en organismos con alta demanda de trámites. Su caso de uso inicial es REGTSA, Organismo Autónomo de Recaudación y Gestión Tributaria de la Diputación de Salamanca. Según trabajadores del área de atención al cliente, aunque dispone de un formulario web y un chatbot informativo, sigue recibiendo numerosas llamadas solo para gestionar citas, lo que satura a los equipos de atención e impide dedicar tiempo a casos más complejos. TurnoFácil aborda este problema al integrar, en un mismo flujo conversacional, la consulta y gestión de citas, evitando cambios de canal y tareas manuales innecesarias. Así, el ciudadano puede reservar, modificar o cancelar su cita mediante una conversación natural conectada directamente con la base de datos.

El contexto demográfico y de uso tecnológico de la provincia refuerza la importancia de un enfoque conversacional inclusivo. Según el Instituto Nacional de Estadística (INE), Castilla y León es una de las comunidades autónomas más envejecida de España, junto con Galicia y Asturias (INE, s.f.). Además, WhatsApp es una herramienta ampliamente extendida incluso entre personas mayores: el 71 % de las personas de 65+ la utiliza (Fundación Telefónica [FT], 2023), lo que avala priorizar este canal como puerta de entrada natural a los servicios públicos y facilita su uso en personas con menor alfabetización digital.

Sobre este contexto operativo y social, TurnoFácil propone un sistema multicanal (web y WhatsApp) con interacción en lenguaje natural y personalización geográfica, capaz de sugerir ubicaciones y horarios óptimos según disponibilidad, proximidad y

preferencias del usuario. El proyecto nace con apoyo institucional explícito de REGTSA y respaldo técnico de UGROUND para investigación, explotación y transferencia tecnológica, lo que facilita su pilotaje en entorno real y su posterior escalado.

Además de responder a una necesidad operativa concreta (reducir tiempos de espera y carga telefónica), la iniciativa se alinea con objetivos de política pública y agenda internacional: ODS 9 (Industria, innovación e infraestructura), ODS 10 (Reducción de las desigualdades) y ODS 16 (Paz, justicia e instituciones sólidas), al promover servicios más accesibles, eficientes y centrados en las personas, con trazabilidad y transparencia mejoradas. (Organización de las Naciones Unidas [ONU], s. f.).

Finalmente, TurnoFácil se plantea como una solución adaptable más allá de REGTSA. Su arquitectura modular y el uso de tecnologías consolidadas, como la inteligencia artificial conversacional, la orquestación mediante API y el almacenamiento en bases de datos, facilitan su aplicación en otros organismos públicos (sanidad, servicios sociales, administración local) e incluso en el sector privado con alta demanda de citas. Su enfoque tipo SaaS y la documentación técnica disponible refuerzan su capacidad de transferencia y su viabilidad en distintos entornos.

2. JUSTIFICACIÓN Y OBJETIVOS

El punto de partida del proyecto TurnoFácil es una necesidad detectada en los organismos públicos que gestionan citas para la atención presencial o telemática. Concretamente, fue detectado en REGTSA que, aunque dispone de un sistema web de cita previa, el volumen de llamadas telefónicas sigue siendo muy elevado y se centra, en gran parte, en concertar o modificar citas, lo que satura los canales de atención según el personal de atención al cliente. Esta situación provoca ineficiencias operativas, costes añadidos y frustración ciudadana, especialmente entre quienes encuentran dificultades en el uso de portales web complejos.

La elección de **WhatsApp** como canal principal responde a un criterio de inclusión digital: es una aplicación ampliamente utilizada por la población, también entre personas mayores, lo que facilita su adopción y garantiza un acceso sencillo sin necesidad de instalar nuevas herramientas ni adquirir competencias digitales avanzadas. De este modo, se promueve la igualdad de oportunidades en el acceso a los servicios públicos y se contribuye a reducir la brecha digital.

Además, la incorporación de IA conversacional y técnicas de procesamiento de lenguaje natural (PLN) permite transformar la experiencia de usuario, al posibilitar una interacción más natural, cercana y personalizada. No se trata únicamente de reservar citas, sino de acompañar al ciudadano durante el proceso, resolviendo dudas sobre el tema, recordando citas pendientes o sugiriendo horarios y localizaciones alternativas.

En este contexto, TurnoFácil se justifica como una respuesta innovadora, eficiente e inclusiva a un problema real de saturación de canales y accesibilidad desigual, con un potencial de aplicación escalable tanto en administraciones públicas (ayuntamientos, sanidad, universidades...) como en entidades privadas (clínicas, asesorías, banca...).

Existen diferentes proyectos y aplicaciones que han tratado de automatizar la gestión de citas mediante asistentes virtuales o plataformas digitales, especialmente en el ámbito sanitario y administrativo. Sistemas como la plataforma de cita previa de SACYL o los asistentes informativos de ayuntamientos como Madrid o Zaragoza, facilitan la atención al ciudadano pero presentan limitaciones: no integran mensajería instantánea, no mantienen una conversación de forma natural o no permiten realizar reservas efectivas en tiempo real (SACYL, s. f.; Ayuntamiento de Madrid, s. f.; El Referente, 2024). Es decir, en lugar de interactuar directamente con la base de datos del organismo, se limitan a ofrecer formularios o información estructurada.

En este contexto, TurnoFácil se diferencia por su enfoque inclusivo, al priorizar WhatsApp como canal de comunicación, y por su arquitectura modular basada en inteligencia artificial, orquestación ligera y bases de datos distribuidas. Estas características permiten ofrecer una experiencia de usuario más natural, accesible y escalable, superando las limitaciones de los modelos anteriores.

2.1. Objetivos

El objetivo general es desarrollar e implementar un chatbot inteligente y accesible para la gestión integral de citas y turnos mediante WhatsApp y web, que reduzca la carga de trabajo manual en los organismos, mejore la experiencia ciudadana y garantice la inclusión digital de colectivos vulnerables.

No obstante, existen otros objetivos específicos:

1. **Automatizar la gestión de citas** (creación, modificación y cancelación) reduciendo la necesidad de llamadas telefónicas y atención manual.
2. **Mejorar la experiencia de usuario** mediante interacción conversacional en lenguaje natural, cercana y clara.
3. **Garantizar accesibilidad universal** priorizando WhatsApp como canal principal, eliminando barreras de entrada tecnológicas.
4. **Optimizar recursos internos** de los organismos, liberando al personal de tareas repetitivas para centrarse en casos de mayor valor añadido.
5. **Ofrecer un sistema personalizable y escalable**, adaptable a distintos sectores (administración pública, sanidad, educación, sector privado).
6. **Integrar fuentes de datos y sistemas existentes**. Por el momento se utilizan datos simulados, pero podría integrarse con el propio sistema actual de REGTSA e incluso mejorar el prototipo son sistemas de autenticación del Estado como Cl@ve para garantizar la interoperabilidad.

7. **Fomentar la transparencia y la trazabilidad**, registrando todas las interacciones y citas en bases de datos seguras y consultables.

3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN

3.1. Arquitectura general del sistema

El sistema TurnoFácil se diseñó bajo un enfoque modular, que combina un backend ligero en Flask, un frontend conversacional en Streamlit, la integración con WhatsApp mediante Twilio y la persistencia de datos en Elasticsearch. Además, se apoya en Eazybase para el registro del historial de conversaciones y en modelos de lenguaje de OpenAI para interpretar la intención del usuario y generar respuestas en lenguaje natural backend.

La arquitectura se estructura en cinco capas: capa de interacción con el usuario, capa de orquestación (backend flask), capa de inteligencia artificial, capa de datos y capa de disponibilidad.

3.1.1. Capa de interacción con el usuario

La primera capa del sistema facilita el contacto directo entre el ciudadano y la aplicación mediante varios canales complementarios. El principal es **WhatsApp**, integrado a través de **Twilio**, que gestiona el envío y recepción de mensajes conectándolos con el backend en Flask. Esta elección responde a un criterio de accesibilidad, ya que el usuario no necesita instalar nuevas aplicaciones ni aprender a manejar herramientas distintas a las que ya usa habitualmente.

Como alternativa, se ha desarrollado una interfaz web con **Streamlit**, que permite probar el asistente en fases de desarrollo y realizar demostraciones en entornos académicos o de laboratorio, manteniendo una experiencia de conversación similar a la de WhatsApp.

La capa se completa con un módulo de calendario en **Streamlit-Calendar**, que muestra de forma gráfica las reservas, permite filtrarlas por teléfono y eliminarlas si es necesario. Esta herramienta facilita tanto al ciudadano como a los gestores una supervisión clara e intuitiva de las citas.

3.1.2. Capa de orquestación (Flask backend)

La capa de orquestación constituye el eje central del sistema, pues coordina la comunicación entre los distintos componentes. Está implementada con **Flask**, un microframework ligero que permite desplegar un servidor flexible y exponer los endpoints REST necesarios para la gestión de citas: creación (/create_appointment),

registro (/record_appointment), modificación (/update_appointment) y eliminación (/delete_appointment).

Más allá de ofrecer estos puntos de acceso, el backend concentra la lógica de negocio que da coherencia al sistema. Entre sus funciones destacan: registrar en **Eazybase** los mensajes intercambiados con el ciudadano para mantener un historial completo; validar la información recibida antes de confirmar o modificar una reserva; coordinar el flujo de datos entre el motor de IA, la base de datos en **Elasticsearch** y los servicios de mensajería de Twilio/WhatsApp; y ejecutar operaciones CRUD sobre las citas almacenadas.

Gracias a esta capa, TurnoFácil consigue integrar de forma armónica tecnologías diversas (procesamiento de lenguaje natural, mensajería instantánea y almacenamiento distribuido) en un flujo único y fiable. Esto garantiza la coherencia del sistema y su fiabilidad.

3.1.3. Capa de inteligencia artificial

La capa de inteligencia artificial es el motor de comprensión y generación de lenguaje del sistema. Su papel es interpretar lo que solicita el usuario y devolver una respuesta coherente y contextualizada en español. Está formada por dos componentes complementarios: el **Router** y el módulo de **Contestation**.

El Router (router.py) analiza los mensajes y los transforma en un JSON estructurado, clasificando las peticiones según su propósito: crear, cerrar, modificar o cancelar una cita. Destaca su capacidad para interpretar expresiones temporales habituales (“mañana”, “la semana que viene”) y convertirlas en instrucciones claras para el sistema.

El módulo de Contestation (answer.py) genera la respuesta final al usuario, aplicando reglas estrictas: confirmar antes de cerrar una reserva, proponer horarios alternativos o validar datos como modalidad y localización. Para ello, consulta tanto la disponibilidad de turnos como el historial de reservas, lo que permite personalizar cada interacción.

Gracias a esta capa, TurnoFácil puede comprender el lenguaje natural en diferentes contextos y responder de manera clara y útil, lo que lo convierte en una herramienta accesible y eficaz para la gestión de citas.

3.1.4. Capa de datos

La capa de datos constituye uno de los pilares fundamentales de TurnoFácil, ya que garantiza que toda la información generada quede registrada y pueda consultarse de manera rápida y fiable. Está formada por dos sistemas que cumplen funciones distintas pero complementarias: **Elasticsearch** y **Eazybase**.

Por un lado, Elasticsearch actúa como base de datos principal para las reservas de citas. Este motor de búsqueda y análisis está pensado para manejar grandes volúmenes de información con eficiencia, por lo que resulta adecuado para un sistema que gestiona muchas operaciones simultáneas. En TurnoFácil, se encarga de crear, consultar, actualizar y eliminar datos, además de diferenciar entre citas futuras y pasadas. Gracias a ello, el sistema puede recordar citas pendientes, mostrar historiales y permitir modificaciones o cancelaciones. También ofrece consultas avanzadas, como filtrar reservas por teléfono o por fecha, lo que facilita el trabajo de usuarios y gestores.

Por otro lado, Eazybase se centra en el registro completo de las conversaciones que el ciudadano mantiene con el asistente. Cada mensaje enviado o recibido queda almacenado junto con la información asociada al perfil, de modo que es posible reconstruir en cualquier momento el hilo de la interacción. Este registro asegura trazabilidad y transparencia, y además aporta un valor añadido al convertirse en una fuente de datos para el análisis de patrones de uso. Gracias a ello es posible identificar dificultades de comprensión, mejorar las respuestas del asistente y optimizar la experiencia de interacción.

En conjunto, esta estrategia dual de almacenamiento dota al sistema de una memoria sólida y confiable, que no solo respalda la operativa diaria, sino que también permite extraer conclusiones de valor para la mejora continua y la evolución futura del proyecto.

3.1.5. Capa de disponibilidad

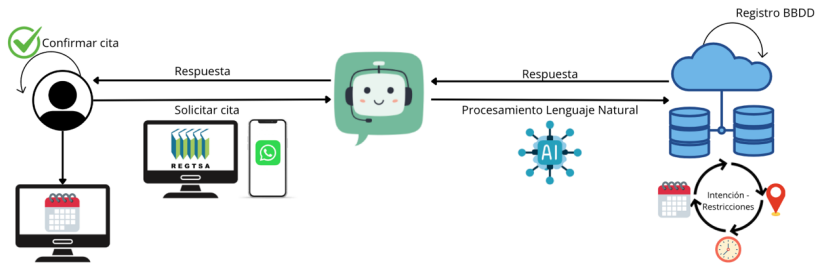
El núcleo de esta capa se compone de dos elementos principales: el script *crear_availability_map.py* y el archivo de configuración *availability_julio2025.json*. El primero genera el mapa de disponibilidad a partir de parámetros de calendario, definiendo qué días y franjas horarias están abiertas al público en cada localidad. Se contemplan tanto las oficinas fijas, con atención recurrente, como las móviles, que siguen un calendario rotatorio en distintos municipios, reflejando así la realidad operativa de REGTSA.

El archivo JSON recoge esta planificación en un formato estructurado, incluyendo también la modalidad online. De este modo, el asistente puede consultar en tiempo real los huecos libres y proponerlos al usuario, reduciendo solapamientos y errores de agenda.

Su principal ventaja es la adaptabilidad: basta con ajustar el script o regenerar el archivo para incorporar festivos, cambios de horario o ampliaciones de servicio, sin modificar el núcleo del sistema. Esto convierte la gestión de citas en un proceso flexible y dinámico, ajustado a las necesidades de cada institución.

3.2. Flujo de trabajo del sistema

Figura 1. Flujo de trabajo del sistema - TurnoFácil



1. El usuario envía un mensaje vía **WhatsApp** o **chat web**.
2. El **backend Flask** recibe el mensaje y lo registra en **Eazybase** junto con la respuesta generada.
3. El módulo **Router** (IA) procesa el mensaje, identifica la intención (crear, modificar, cancelar) y devuelve un JSON estructurado.
4. El módulo **Contestation** genera una respuesta en español, proponiendo citas disponibles, confirmando operaciones o solicitando información faltante.
5. Según el caso:
 - Si se confirma una cita, se registra en **Elasticsearch**.
 - Si se modifica o cancela, se actualiza o elimina el registro correspondiente.
6. La respuesta final se envía al usuario por WhatsApp (Twilio) o se muestra en la interfaz web.
7. Opcionalmente, el usuario o administrador puede consultar el **calendario** visual en Streamlit, filtrando citas y eliminándolas con un clic.

3.3. Desarrollo y tecnologías empleadas

El desarrollo de TurnoFácil se ha llevado a cabo en Python 3.10 o superior, lo que garantiza compatibilidad con las librerías más recientes y estabilidad en el entorno de ejecución. Para la construcción del sistema se han empleado dos *frameworks* principales: Flask, encargado de implementar el backend REST y gestionar los distintos endpoints del servicio, y Streamlit, que ha permitido desarrollar tanto la interfaz web de interacción con el chatbot como el calendario visual de reservas.

La inteligencia artificial conversacional se apoya en la API de OpenAI, utilizando el modelo *gpt-4o-mini*, que ofrece un buen equilibrio entre capacidad de comprensión en lenguaje natural y eficiencia de procesamiento. En lo que respecta a la mensajería, la integración con WhatsApp se ha resuelto mediante el uso de Twilio, que actúa como puente entre los usuarios y el backend del sistema.

Para la gestión de datos se ha optado por Elasticsearch 8.x, configurado con índices específicos como `reservation_upsa` y `te_reservation`, que permiten almacenar, consultar y manipular de manera eficiente la información relativa a las citas. La configuración del entorno se gestiona a través de ficheros `.env`, implementados con la librería `python-dotenv`, lo que facilita mantener seguras las credenciales y parámetros de conexión sin exponerlos en el código. Finalmente, la capa de disponibilidad se resuelve mediante archivos JSON dinámicos, generados automáticamente a partir de scripts de calendario y horarios, que definen los turnos y franjas horarias en cada oficina o modalidad de atención.

4. RESULTADOS Y VALIDACIÓN DEL SISTEMA

El desarrollo de TurnoFácil dio como resultado un prototipo funcional, capaz de gestionar automáticamente la creación, modificación y cancelación de citas a través de dos canales: WhatsApp y una interfaz web en Streamlit. Todas las capas del sistema fueron implementadas y probadas, garantizando la coherencia entre backend, inteligencia artificial, base de datos y calendario.

Uno de los principales logros fue la automatización completa del flujo de reservas, desde la interpretación del mensaje en lenguaje natural hasta el registro de la cita en Elasticsearch, con confirmación inmediata al usuario. El sistema demostró además su capacidad para detectar distintas intenciones y manejar expresiones temporales habituales como “la próxima semana” o “el viernes que viene”.

La validación tuvo lugar en las jornadas de proyectos TALENT, donde participaron alumnos, profesorado y representantes de entidades colaboradoras. Destacó la asistencia de Vega Cortés Pérez, técnica de REGTSA, cuyo feedback confirmó la utilidad del prototipo para la problemática real de gestión de citas. Los asistentes subrayaron también la facilidad de uso de WhatsApp y la claridad del calendario visual como herramienta de apoyo.

En conjunto, los resultados muestran que TurnoFácil es una alternativa viable y eficiente para mejorar la gestión de citas y reducir la carga telefónica. Aunque se trata de un prototipo, las pruebas evidencian su potencial de escalabilidad y adaptación a otros organismos públicos y privados con necesidades similares.

Figura 2. Ejemplo funcionamiento prototipo via WhatsApp - solicitar cita

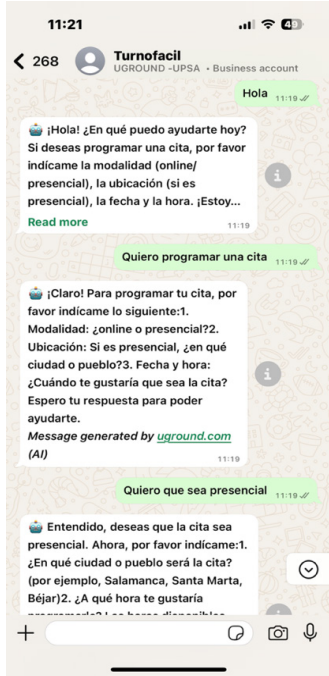
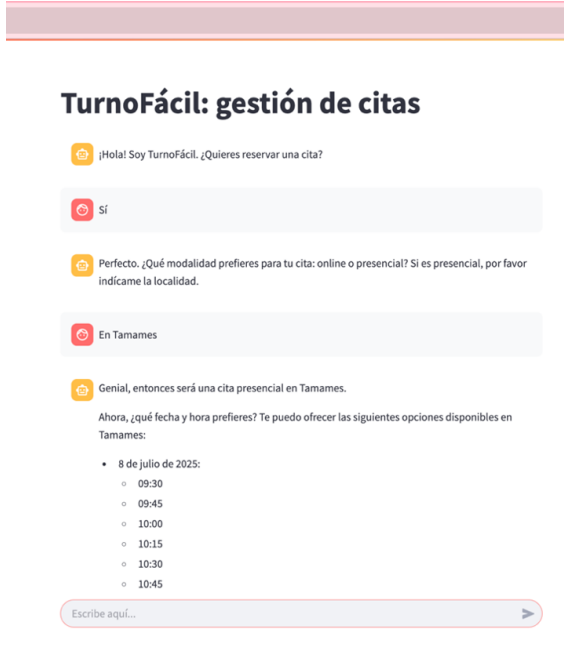
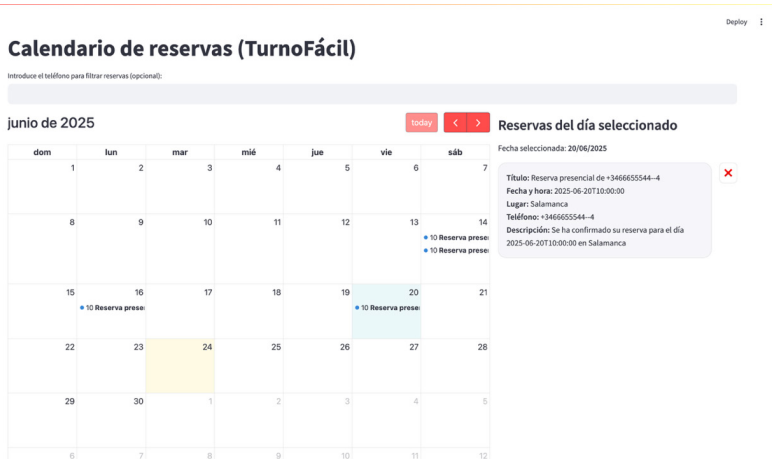


Figura 3. Ejemplo funcionamiento prototipo vía web - solicitar cita



A continuación, se muestran algunos ejemplos de funcionamiento del prototipo:

Figura 4. Ejemplo funcionamiento prototipo vía web - calendario



5. CONCLUSIONES Y REFLEXIONES FINALES SOBRE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO

El desarrollo de TurnoFácil ha permitido demostrar que es posible construir un asistente conversacional eficaz para la gestión de citas, combinando la inmediatez de WhatsApp con la flexibilidad de una arquitectura modular. El prototipo ha validado de forma práctica que un sistema de este tipo puede reducir la carga de llamadas telefónicas, mejorar la experiencia ciudadana y ofrecer un servicio inclusivo y accesible para diferentes colectivos.

Tras la presentación en las jornadas de proyectos TALENT y las posteriores reuniones mantenidas con REGTSA y UGROUND, se confirmó el interés en la propuesta y su alineación con las necesidades reales de las administraciones públicas. Sin embargo, las particularidades del ecosistema tecnológico actual de REGTSA no han facilitado la implantación inmediata del sistema en dicho organismo. Este hecho no resta valor al trabajo realizado, sino que señala un nuevo reto: explorar otras vías de transferencia y búsqueda de clientes potenciales que puedan beneficiarse directamente de la solución desarrollada.

En este sentido, la naturaleza del proyecto, concebido como un servicio SaaS adaptable, abre la puerta a su aplicación en distintos contextos. Además de las diputaciones y organismos tributarios, TurnoFácil puede resultar útil en centros sanitarios, universidades, servicios sociales, ayuntamientos de menor tamaño o incluso entidades privadas que gestionen un alto volumen de citas (clínicas, asesorías, despachos profesionales, entre otros). La versatilidad del sistema, junto con el respaldo institucional y académico recibido, refuerza su potencial de crecimiento.

Por tanto, la principal conclusión que se desprende del trabajo no es únicamente la viabilidad técnica del prototipo, sino también la necesidad de seguir impulsando contactos y alianzas con instituciones interesadas en modernizar su atención al ciudadano mediante herramientas de inteligencia artificial. TurnoFácil sienta bases sólidas para futuras colaboraciones y proyectos, contribuyendo al objetivo compartido de fomentar una administración más accesible, eficiente e inclusiva.

BIBLIOGRAFÍA Y REFERENCIAS ELECTRÓNICAS

Ayuntamiento de Madrid. s.f. *Asistente Virtual de Línea Madrid*. Recuperado de: <https://www.madrid.es/portales/munimadrid/es/Inicio/El-Ayuntamiento/Atencion-a-la-ciudadania/Asistente-Virtual/Asistente-Virtual-de-Linea-Madrid/?vgnnextfmt=default&vgnnextoid=c6ca14e8451ff610VgnVCM2000001f4a900aRCRD&vgnnextchannel=498a14e8451ff610VgnVCM2000001f4a900aRCRD>

Clara Labs. (2022). *Clara.ai Virtual Assistant for Scheduling*. Recuperado de <https://claralabs.com>

- El Referente. (2024, 6 de septiembre). *Zaragoza impulsa un asistente virtual de IA para atender a los ciudadanos*. Recuperado de <https://elreferente.es/actualidad/zaragoza-impulsa-asistente-virtual-ia-atender-a-ciudadanos/>
- Fundación Telefónica. (2023). *Sociedad Digital en España 2023*. Fundación Telefónica. Recuperado de: https://publiadmin.fundaciontelefonica.com/media/publicaciones/780/Sociedad_Digital_en_Espana_2023.pdf
- Instituto Nacional de Estadística (INE). (s. f.). *Padrón municipal. Personas por sexo*. Recuperado de https://www.ine.es/jaxiT3/Datos.htm?t=1452#_tabs-mapa
- Organización de las Naciones Unidas (ONU). (s. f.). *Objetivos de Desarrollo Sostenible*. Naciones Unidas. Recuperado de <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>
- SACYL (Junta de Castilla y León). (s. f.). *Cita Previa con su Centro de Salud*. Recuperado de <https://www.saludcastillayleon.es/es/citaprevia>

SISTEMA DE NUTRICIÓN AVANZADO Y PERSONALIZADO (SNAP)



ALEJANDRO LÓPEZ BARRANTES

ROBERTO BERJÓN GALLINA

MARÍA ENCARNACIÓN BEATO GUTIÉRREZ

RESUMEN

El proyecto SNAP (Sistema de Nutrición Avanzado y Personalizado) consiste en el desarrollo de una aplicación móvil inteligente enfocada en la planificación nutricional personalizada mediante el sistema de intercambios alimentarios. A través de tecnologías modernas como Flutter/Dart (frontend), PHP (backend) y MySQL alojado en Hostalia, SNAP permite al usuario generar planes nutricionales semanales equilibrados, adaptados a sus características físicas, objetivos y hábitos. Su innovación principal radica en el uso del **método de intercambios**, que agrupa los alimentos por su valor nutricional y permite al sistema sustituir automáticamente unos por otros, garantizando variedad sin alterar el equilibrio calórico y nutricional. La aplicación incluye, además, una herramienta de inteligencia artificial (Google Vision AI) capaz de reconocer alimentos mediante fotografías, facilitando la identificación y sustitución dentro de cada grupo de intercambio. Actualmente, la primera fase del proyecto está finalizada, abarcando tanto la estructura técnica como la lógica de generación de menús. El siguiente paso es la validación profesional del sistema junto a un nutricionista acreditado, con el objetivo de transferir la aplicación al ámbito real de la nutrición personalizada.

1. CONTEXTUALIZACIÓN DEL PROYECTO

El aumento de la preocupación por la alimentación saludable y los estilos de vida equilibrados ha impulsado la aparición de múltiples aplicaciones de nutrición y con-

trol dietético. Sin embargo, la mayoría de ellas se limitan a calcular calorías o registrar comidas sin ofrecer un modelo verdaderamente adaptativo, educativo y científicamente fundamentado.

El sistema de intercambios alimentarios, utilizado desde hace décadas por nutricionistas, constituye una herramienta eficaz para diseñar planes alimentarios flexibles. Este método permite sustituir alimentos pertenecientes al mismo grupo —como lácteos, cereales, frutas o proteínas— sin modificar el valor nutricional total del plan. SNAP digitaliza y automatiza este sistema tradicional, generando menús adaptados al usuario de forma dinámica.

La relevancia del proyecto se enmarca en un contexto donde la educación nutricional digital y la personalización de la dieta son aspectos esenciales para promover la salud pública. La aplicación busca democratizar el acceso a herramientas profesionales, permitiendo que cualquier persona pueda seguir un plan alimentario equilibrado sin necesidad de conocimientos técnicos previos. Además, SNAP está diseñado para que profesionales de la nutrición puedan utilizarlo como apoyo clínico, ya que ofrece datos estructurados y configurables según las necesidades de cada paciente.

En un momento en el que la inteligencia artificial y las tecnologías móviles están transformando el sector sanitario, SNAP se posiciona como una herramienta que une ciencia, tecnología y bienestar, aplicando los avances tecnológicos de forma práctica y ética.

2. JUSTIFICACIÓN Y OBJETIVOS

El principal problema que SNAP aborda es la falta de personalización real y flexibilidad en las aplicaciones de nutrición existentes. El método de intercambios representa un enfoque más humano, dinámico y adaptativo, que permite a los usuarios modificar su alimentación diaria sin salir del plan prescrito.

La aplicación ofrece una solución tecnológica a una necesidad práctica: cómo seguir una dieta equilibrada sin rigidez, adaptándola al día a día del usuario. Al integrar inteligencia artificial, bases de datos nutricionales y automatización, SNAP convierte un proceso complejo en una experiencia sencilla e intuitiva.

Objetivo general: Desarrollar una aplicación móvil capaz de generar y gestionar planes nutricionales semanales basados en intercambios alimentarios, personalizados según las características y objetivos de cada usuario, integrando inteligencia artificial para el reconocimiento y sustitución automática de alimentos.

Objetivos específicos:

- Implementar el sistema de intercambios como base para la planificación de menús semanales equilibrados.

- Automatizar el cálculo energético y la distribución de macronutrientes mediante fórmulas científicas.
- Incorporar Google Vision AI para identificar alimentos por imagen y sugerir sustituciones equivalentes.
- Crear una interfaz atractiva y sencilla, que motive al usuario y mejore la adherencia a su plan.
- Permitir la participación de profesionales de la nutrición en la personalización de los planes.

3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN

El desarrollo del **Sistema de Nutrición Avanzado y Personalizado (SNAP)** se llevó a cabo siguiendo un proceso estructurado y planificado desde las primeras etapas. Antes de comenzar la programación, se realizó una fase de **análisis y diseño** en la que se definieron los requisitos funcionales, casos de uso y flujo general de la aplicación.

Para una correcta gestión del proyecto se utilizaron herramientas de planificación y diseño ampliamente empleadas en el entorno profesional:

- **Trello:**
Se empleó como sistema de organización de tareas, permitiendo estructurar el desarrollo en diferentes fases (*backlog, en proceso, en revisión y hecho*). Cada módulo —como el registro de usuarios, el cálculo energético o la generación de menús por intercambios— se gestionó mediante tarjetas y listas, lo que facilitó el control del progreso y la priorización de tareas.
- **Figma:**
Se utilizó para la creación de los **bocetos iniciales (wireframes)** y el diseño visual preliminar de la interfaz de usuario. Esta herramienta permitió visualizar la disposición de los elementos principales de la aplicación, establecer la identidad visual de SNAP y asegurar una experiencia de usuario coherente e intuitiva antes de su implementación en Flutter.

Posteriormente, se pasó a la fase de desarrollo técnico, aplicando un enfoque modular y escalable que combina distintas tecnologías para garantizar rendimiento y mantenibilidad:

- **Frontend (Flutter/Dart):**
La interfaz fue implementada en Flutter, permitiendo compatibilidad multiplataforma y una navegación fluida. Se siguió el patrón **MVC**, separando la lógica de negocio de la presentación. Los menús generados se muestran al

usuario con la posibilidad de intercambiar alimentos dentro de su grupo nutricional.

- **Backend (PHP):**

Los servicios RESTful desarrollados en PHP se encargan de la comunicación con la base de datos, gestionando peticiones para el registro, cálculo energético, creación de menús y sustitución de alimentos. El código fue estructurado en módulos independientes para facilitar su mantenimiento y escalabilidad.

- **Base de datos (MySQL en Hostalia):**

Se diseñó un modelo relacional con tablas dedicadas a usuarios, alimentos, grupos de intercambio, menús y planes semanales. La información alimentaria utilizada en SNAP fue extraída y normalizada a partir de la base de datos BECDA (Base de Datos Española de Composición de Alimentos). Esta fuente proporciona datos oficiales sobre la composición nutricional de los alimentos consumidos en España, lo que garantiza la fiabilidad y precisión científica del sistema. A partir de estos datos, se clasificaron los alimentos por grupos de intercambio, asignando a cada uno su aporte calórico y nutricional (hidratos de carbono, proteínas, grasas y micronutrientes).

- **Integración con Google Vision AI:**

SNAP emplea la API de Google Vision para reconocer alimentos por imagen. Una vez identificado el alimento, el sistema lo asocia a su grupo de intercambio y sugiere sustituciones nutricionalmente equivalentes, haciendo posible que el usuario realice cambios sin perder el equilibrio del plan.

- **Pruebas y seguridad:**

El sistema se probó tanto en emuladores como en dispositivos reales, evaluando su rendimiento, estabilidad y compatibilidad. Se aplicaron medidas de seguridad básicas como validaciones en servidor, cifrado de contraseñas y peticiones HTTP, además de una gestión controlada de errores para garantizar la integridad de los datos.

4. CONCLUSIONES Y REFLEXIONES FINALES

El desarrollo de SNAP ha sido un proceso de crecimiento tanto técnico como personal. Afrontar un proyecto de esta magnitud en solitario ha requerido planificación, disciplina y constancia, así como la capacidad de resolver imprevistos en distintas áreas tecnológicas.

Durante su desarrollo se constató la importancia de un análisis previo profundo, que permita anticipar errores y reducir contratiempos en las fases críticas del proyecto. Las herramientas de organización como Trello y de diseño como Figma resultaron fundamentales para mantener una visión global y ordenada del proceso.

La primera fase, que culmina con la implementación funcional del sistema de intercambios, demuestra que la automatización de principios nutricionales tradicionales mediante tecnología moderna es viable y útil. SNAP no solo cumple con los objetivos técnicos planteados, sino que sienta las bases de un producto con potencial real de aplicación profesional.

El siguiente paso consistirá en la colaboración con un profesional de la nutrición, para validar el modelo con usuarios reales, ajustar los parámetros de cálculo y mejorar la experiencia de uso. En el futuro, se prevé incluir nuevas funciones como el seguimiento de progreso del usuario, notificaciones inteligentes, recomendaciones basadas en IA y conexión con dispositivos de salud (smartwatches, básculas, etc.).

SNAP representa la fusión entre ciencia, tecnología y salud, y demuestra cómo un proyecto individual puede convertirse en una herramienta de valor social. Más allá de su aspecto técnico, este desarrollo ha supuesto una experiencia formativa que ha fortalecido mis capacidades como programador, diseñador y gestor de proyectos, así como mi comprensión de la importancia del rigor científico en el ámbito de la nutrición.

BIBLIOGRAFÍA

Organización Mundial de la Salud (OMS). *Guías sobre dieta saludable*. 2020.

BECDA (Base de Datos Española de Composición de Alimentos). Agencia Española de Seguridad Alimentaria y Nutrición (AESAN).

Harris, J. A., & Benedict, F. G. (1919). *A Biometric Study of Human Basal Metabolism*.

Google Developers. *Cloud Vision API Documentation*. 2024.

PHP.net. *PHP Manual*. 2024.

Flutter.dev. *Flutter Documentation*. 2024.

Figura 1



Figura 2



Figura 3



Figura 4



Figura 5



Figura 6



SMART TRACKING FRAMEWORK FOR IOT – GATEWAI



ROBERTO BERJÓN GALLINAS

MARÍA ENCARNACIÓN BEATO GUTIÉRREZ

MONTSERRAT MATEOS SÁNCHEZ

ANA MARÍA FERMOSE GARCÍA

RESUMEN

Las aplicaciones *IoT* (Internet of Things) se diseñan empleando arquitecturas basadas en microservicios. Este tipo de arquitecturas aportan numerosas ventajas: la fragmentación de las aplicaciones en pequeños componentes facilita su desarrollo, la utilización de tecnologías de contenedores estandariza sus despliegues lo que, por otra parte, no sólo permite su portabilidad entre distintos proveedores *cloud*, además garantiza la escalabilidad de la aplicación. Sin embargo, el uso de microservicios también tiene sus desventajas ya que conlleva una excesiva complejidad a la hora de interactuar con ellos. Para simplificar el acceso a los mismos, las aplicaciones incorporan un componente denominado genéricamente *gateway*. Su función es la actuar como único punto de entrada en la aplicación, centralizando todas las peticiones de los usuarios y encargándose de enrutarlas a los microservicios correspondientes.

Actualmente, los *gateways* comerciales como *NGINX*, *Kong Gateway*, *Traefik* y otros cumplen estas funciones de enrutamiento, pero pocos tienen la capacidad de agregación de servicios. Esta característica avanzada permite combinar respuestas de varios microservicios para responder a peticiones complejas de los usuarios. Los *gateways* comerciales con esta característica están ligados a proveedores *cloud* (*AWS*, *Azure*, ...) Estos requieren, para su correcto funcionamiento, el uso de otras herra-

mientas proporcionadas por dichos proveedores, lo que añade una complicación adicional y limita su portabilidad en los despliegues.

Ante esta problemática, se propone el desarrollo de *SCIFI Gateway*, un *gateway* con capacidades avanzadas centrado en la agregación de servicios mediante Inteligencia Artificial (IA) y modelos de lenguaje de gran escala (LLM). La complejidad derivada de la agregación de servicios se resolverá describiendo en lenguaje natural cómo debe realizarse ésta, por lo que ya no será preciso la programación de herramientas adicionales. A partir de dicha descripción, la IA generará el flujo con las peticiones a los distintos microservicios y combinará los resultados obtenidos para formar la respuesta que finalmente se envíe al usuario.

1. CONTEXTUALIZACIÓN

Hoy en día el desarrollo de aplicaciones *IoT* o *cloud* se realiza utilizando arquitecturas basadas en microservicios. Esto supone descomponer la aplicación en múltiples servicios donde cada uno tiene su propia responsabilidad o funcionalidad. Gracias a que cada microservicio es un módulo autónomo, los equipos de desarrollo pueden utilizar las tecnologías y lenguajes de programación que mejor se adapten a las necesidades específicas de cada uno, optimizando de esta forma el rendimiento de cada componente. Por otra parte, y con el propósito de estandarizar el despliegue de las aplicaciones, la implementación de los microservicios se realiza mediante contenedores, lo que garantiza su portabilidad entre diferentes proveedores *cloud*, redundando de esta forma la capacidad que ahora tienen las aplicaciones de realizar despliegues híbridos donde se pueda combinar recursos *on-premise* (aquellos que se alojan y gestionan físicamente en las instalaciones de la empresa) con recursos *cloud* alojados además en distintos proveedores.

Hay que resaltar que el uso de arquitecturas basadas en microservicios permite que las aplicaciones sean escalables, es decir, tengan la capacidad de adaptarse a fluctuaciones en la carga de trabajo (peticiones que realizan los usuarios). Esta escalabilidad supone añadir instancias de los contenedores en los que están alojados los microservicios y así repartir en ellos la carga de trabajo, por lo que no hay una pérdida en cuanto al rendimiento y calidad de servicio. Lógicamente, si se produjese una disminución de la carga de trabajo también se podría eliminar instancias de contenedores con el propósito de ahorrar recursos.

Desde el punto de vista del cliente, independientemente que se trate de un cliente web, un dispositivo móvil o un componente de Internet de las Cosas (*IoT*), el acceso a la aplicación puede llegar a convertirse en un proceso complejo, ya que no se sabe cuáles son, en cada instante, los contenedores a los que habría que conectarse para acceder a determinados servicios. Por este motivo es esencial incluir en la arquitectura de las aplicaciones un nuevo componente denominado genéricamente

API Gateway. Actúa como único punto de entrada de la aplicación siendo su función la de centralizar todas las peticiones de los usuarios y enrutarlas a los servicios adecuados gestionando además el balanceo de carga entre las distintas instancias disponibles. Esto permite reducir la complejidad en la interacción entre el cliente y el conjunto de microservicios. Además del enrutamiento, un *API Gateway* de forma generalizada incluye más características: seguridad centralizada (en lugar de que cada microservicio implemente sus propias medidas de seguridad, incluir en el Gateway tanto la autenticación como la autorización por lo que se asegura un acceso uniforme y controlado), gestión de políticas con las que controlar el tráfico (permite definir de límites de velocidad, cuotas de acceso y facilita protección contra ataques de denegación de servicios), monitorización (recopila y analiza métricas y eventos en tiempo real para evaluar el rendimiento general) y trazabilidad (realiza un seguimiento del camino que recorre cada solicitud a través de los diferentes microservicios analizando los tiempos de respuesta a fin de identificar problemas específicos y cuellos de botella sobre los que actuar para optimizar el rendimiento general).

Aunque una arquitectura de microservicios aporta múltiples ventajas en el desarrollo de aplicaciones, también introduce una complejidad adicional: cuando un usuario realiza una solicitud compleja, será necesario interactuar con múltiples microservicios para obtener toda la información requerida. Además, habría que tener en cuenta que la comunicación entre los microservicios no tiene que ser homogénea, es decir, no tienen por qué utilizar el mismo protocolo de comunicación. Para solucionar esto, sería necesario que el *gateway* dispusiera de una característica avanzada: la agregación o combinación de servicios. Esta característica facultaría al *gateway* a realizar múltiples peticiones a distintos microservicios para combinar los resultados obtenidos y así generar la respuesta que el cliente pudiese procesar. De esta forma se lograría minimizar el número de interacciones entre el cliente y los servicios lo que supondría una optimización en la eficiencia (ya que se reduciría la latencia -los tiempos de espera-) y mejoraría en la experiencia del usuario.

La agregación de servicios es una característica que no está lo suficientemente extendida en los *gateways* analizados. Sólo está soportado por *AWS API Gateway*, *Apigee* y *Azure API Management* que son, respectivamente, los *gateway* proporcionados por los proveedores cloud *AWS*, *Google Cloud* y *Azure* respectivamente. El inconveniente es que en estas soluciones sólo pueden emplearse en los entornos de sus correspondientes proveedores y además requieren la utilización de otras herramientas, también proporcionadas por los proveedores, para su implementación: funciones *Lambda* en el caso de *AWS API Gateway*, scripts definidos en *JavaScript* o *Python* en el caso de *Apigee* y *Azure Functions* o *Logic Apps* en el caso de *API Management*.

Analizada esta problemática, el objetivo que se pretende alcanzar en este proyecto es el diseño y desarrollo de un *gateway* (SCIFI GatewAi) especializado en la agregación de servicios. A diferencia de lo que sucede en los *gateways* descritos anteriormente donde es imprescindible utilizar otra serie de herramientas para realizar su configuración, en la solución que se propone sólo será necesario una descripción

en lenguaje natural de cómo debe realizarse dicha agregación. Para ello se hará uso de Inteligencia Artificial Generativa (GenAi) a través de modelos LLM (Large Language Model). A partir de dicha descripción, la iA (inteligencia artificial) generará el flujo con las peticiones a los distintos microservicios, así como combinará las respuestas obtenidas para formar la respuesta que finalmente se envíe al usuario.

2. DISEÑO E IMPLEMENTACIÓN

Un agente iA es una entidad software autónoma que observa el entorno, dispone de una serie de funcionalidades a través de las que realizar una serie de tareas y tiene la capacidad de reaccionar de forma inteligente cuando detecta cualquier cambio en el entorno. En base a dicho cambio, el agente iA ejecuta de forma autónoma las funcionalidades que considere adecuadas con el propósito, por ejemplo, de garantizar que el entorno se mantenga en un estado predefinido.

En este trabajo hemos desarrollado un tipo de agente iA que utiliza un LLM como “motor de razonamiento”. Su principal característica es que permite generar la lógica a seguir cuando se detecta un cambio, es decir, el LLM genera un flujo de trabajo, descrito mediante la especificación *AWS States Language*, que secuencia las funcionalidades de la aplicación (*task pool*) que habría que ejecutar. Este flujo de trabajo se ejecutaría posteriormente a través de un orquestador de *workflows*.

Este agente iA se ha utilizado en el desarrollado del *gateway* inteligente SCIFI *GatewayAi*. De forma dinámica se genera una implementación de una API REST descrita a partir de la especificación OpenAPI. Esta implementación hace uso del Agente iA que reaccionará cuando los clientes accedan a las distintas operaciones que se publican a través del API REST. La respuesta que obtendrán los clientes será el resultado de procesar el flujo de trabajo que ha generado el LLM para cada operación. Mediante este flujo se podrá no sólo redirigir las peticiones de los clientes a los servicios correspondientes, sino que también será capaz de combinar la acción de distintos servicios para realizar tareas complejas.

2.1. Definición de las funcionalidades de la aplicación o *task pool*

Es necesario definir el *task pool* conteniendo todas las tareas que podrán ser invocadas por el sistema. Estas tareas deben ser descritas a través de clases decoradas con la anotación `@Task`. Esta anotación posee distintos elementos relacionados con los distintos tipos de tareas soportadas:

- a. **inout**: se corresponde con tareas asíncronas que reciben un dato JSON desde un determinado *channel* de entrada y devuelven otro dato JSON por otro *channel* de salida. A cada tarea hay que asignarle: **name** (actúa como identi-

ficador de forma que será utilizado posteriormente para hacer referencia a ella en los *workflows*), **description** (descripción detallada del cometido de la tarea) así como los identificadores de los *channel* de entrada y salida (**inputChannel** y **outputChannel**).

Código 1. Definición de tareas *inout*

```
@Tasks(inout={
    @InOutTask(name="uppercase",
               inputChannel="to-uppercase",
               outputChannel="from-uppercase",
               description="Cambia el nombre de un alumno a mayúsculas"
            )
})
@ApplicationScoped
@Unremovable
public class UppercaseTasksDescriptor
{
    @Inject
    @TaskChannel(description = "to-uppercase: Recibe un objeto de tipo Alumno")
    @Channel("to-uppercase")
    MutinyEmitter<JsonNode> toUppercaseEmitter;

    @Inject
    @TaskChannel(description = "from-uppercase: Devuelve un objeto de tipo Alumno")
    @Channel("from-uppercase")
    Multi<Message<JsonNode>> fromUppercaseMulti;
}
```

- b. **rest**: se corresponde con tareas vinculadas con invocaciones asíncronas a operaciones de una API REST. Por cada tarea hay que indicar en una anotación **@ResTask** los elementos: **name** y **description** (sus significados son idénticos a los descritos en las tareas definidas en el elemento **inout**) así como el tipo de dato que hay que enviar y que se espera recibir tras la ejecución de la tarea (**inputChannel** y **outputChannel**) Para describir cada uno de estos tipos de dato se utiliza una anotación **@TaskChannel** en donde se debe indicar una descripción del dato (**description**) y las coordenadas **namespace**, **type** y **version** que permitan localizar dentro de un registro de schemas JSON la estructura del dato. La URL a través de la que localizar el schema se obtendrá a través de una URL que, en cada caso, se obtendrá a partir de la plantilla definida en la propiedad de configuración **scifi.workflows.default.builtin-task-urn-format**. Su valor por defecto es:

`http://<host>:<port>/apis/registry/v2/groups/%1$s/artifacts/%2$s/versions/%3$s`

donde **host** y **port** hacen referencia respectivamente a la dirección IP (o nombre DNS) y puerto del registro de schemas (sus valores por defecto se especifican a través de las propiedades de configuración **scifi.workflows.default.schema-registry.host** y **scifi.workflows.default.schema-registry.port**) y **%1\$s**, **%2\$s** y **%3\$s** tomarán, respectivamente, los valores definidos en **namespace**, **type** y **version**.

Por último, por cada tarea hay que incluir a través del elemento **parameters** una anotación **@RestTaskParameters** con la que especificar una serie de parámetros:

- *method*: valor de la enumeración `RestBuiltInTask.Method` que representa el verbo HTTP utilizado GET, POST, PUT, DELETE, PATCH, HEAD, OPTIONS (elemento **method**).
- *endpoint*: URL que se correspondería con el valor `server` de la descripción OpenAPI de la API REST. Esta URL se puede especificar de forma estática a través de un literal o bien de forma dinámica mediante una expresión que utilice *intrinsic functions*. Según la forma en la que se vaya a especificar habría que utilizar o bien el elemento **endpoint** (utilizado cuando la URL se indique a través de un literal) o bien el elemento **endpoint\$**. Así por ejemplo **endpoint\$="Scifi.Config('alumnos.url')"** indicaría que la url se obtendría de la propiedad de configuración `alumnos.url` o bien de la variable de entorno `ALUMNOS_URL`.
- *path*: representa el *path* de la *operation* que se concatenaría al *endpoint* para obtener la url final. Este path puede incluir *path params* mediante la inclusión de llaves {}. Hay dos elementos **path** y **path\$** que se emplearán para indicar el *path* bien a través de un literal (**path**) o bien de forma dinámica a través de una expresión *intrinsic functions* (**path\$**).
- *pathParams*: solo en el caso de que el *path* anterior incluyese *path params*, habría que indicar a través del elemento **pathParams** el valor de cada uno de ellos. **pathParams** representa un array de anotaciones **@Param** donde cada una representa un *path param*. Sus elementos **name** (o su versión **name\$**) y **value** representan, respectivamente, el nombre del *path param* representado a través de un literal (o bien mediante una expresión *intrinsic functions*) y su valor (representado por una *JsonPath expression* que se aplicaría al input de la tarea descrito en *inputChannel*).

Así, por ejemplo, asumiendo que existiese una variable de entorno `ALUMNOS_URL` cuyo valor fuese `http://ms-alumnos:8080`, la descripción de la siguiente tarea `findMatriculacionAlumnoByExpediente` haría referencia a una petición `GET` a un recurso con el *path templating* `http://ms-alumnos:8080/alumnos/{expediente}/matriculas`. De esta forma, si al invocar a dicha tarea se le pasase como dato de entrada el objeto JSON `{"expediente": "a44"}`, la URL final a la que se accedería sería: `http://ms-alumnos:8080/alumnos/a44/matriculas`.

Código 2. Definición de task de tipo rest

```

@Tasks(rest={
    @RestTask(name="findMatriculacionAlumnoByExpediente",
        description="Se obtiene el 'codigo' y 'calificacion' de las asignaturas
        en las que se ha matriculado el alumno identificado por
        'expediente'."
        inputChannel=@TaskChannel(namespace = "scifi",
            type = "IdAlumno",
            version = "1.0.0",
            description="Objeto Json conteniendo el
            expediente del alumno"),
        outputChannel=@TaskChannel(namespace="scifi",
            type="MatriculacionAlumno",
            version="1.0.0",
            description="Describe de qué asignaturas se ha
            matriculado el alumno así como
            las calificaciones obtenidas"),
        parameters=@RestTaskParameters(method=RestBuiltInTask.Method.GET,
            endpoint$="Scifi.Config('alumnos.url')",
            path="alumnos/{expediente}/matriculas",
            pathParams={
                @Param(name="expediente",
                    value="$.expediente"
                )
            },
        )
    )
}
)

```

- c. **stateMachine**: se corresponde con tareas vinculadas a un *workflow*. El código de dicho *workflow* estará codificado, bien de forma explícita o bien de manera indirecta, a través *SCIFI Workflows DSL*. En el caso de que se optase por codificar el *workflow* de forma indirecta, lo que habría que proporcionar sería una descripción en lenguaje natural del cometido del *workflow* para que, posteriormente, la iA generase el código correspondiente a partir de dicha descripción.

Por cada tarea se debe incluir una anotación **@StateMachineTask** en la que sus elementos **name**, **description**, **inputChannel** y **outputChannel** son equivalentes a los elementos homónimos de las anotaciones **@RestTask** analizadas anteriormente. Adicionalmente hay que incluir el elemento **stateMachine** que o bien incluye explícitamente el código DSL correspondiente al *workflow* o bien la descripción en lenguaje natural del cometido de este. Será el elemento **generatorType** el que especifique cómo debe interpretarse finalmente el valor de **stateMachine**: si **generatorType** tomase el valor **StateMachineGenerator.Type.EXPLICIT**, se esperará que **stateMachine** contendrá el código *SCIFI Workflows DSL* del *workflow*. Si por el contrario **generatorType** tomase el valor **StateMachineGenerator.Type.PROMPT** se esperará que **stateMachine** represente el *prompt* que habría que pasarle a la iA para que ésta generase el código *SCIFI Workflows DSL* del *workflow*.

Código 3. Definición de una tarea de tipo *stateMachine*

```

@Tasks
(
  stateMachine={
    @StateMachineTask
    (
      name="mapMatriculaToMatriculaFull",
      description="Transforma Matricula en MatriculaFull",
      inputChannel=@TaskChannel(namespace="scifi",
        type="Matricula",
        version="1.0.0",
        description="Objeto de tipo
          Matricula"
      ),
      outputChannel=@TaskChannel(namespace="scifi",
        type="MatriculaFull",
        version="1.0.0",
        description="Objeto de tipo MatriculaFull"
      ),
      generatorType=StateMachineGenerator.Type.PROMPT,
      stateMachine = """
        El input es un objeto Matricula.
        Debes secuenciar los siguientes pasos:
        1. Obtener la Asignatura identificada por el atributo
        'codigo' del input.
        2. Crea un objeto MatriculaFull a partir de los objetos
        Matricula (el input) y Asignatura (obtenido anteriormente)
        """
    )
  )
)

```

2.2. Recopilación de la información de tareas

En este proceso, se recopila la información definida en todas las tareas del pool a fin de tenerlas almacenadas en un registro. En primer lugar, se obtiene la información de las tareas “estáticas” (las definidas en *inout* seguido de las incluidas en *rest*) y posteriormente de las tareas “dinámicas” (las descritas en *stateMachine*). Se hace así puesto que las tareas “dinámicas” pueden necesitar utilizar tareas “estáticas” a la hora de generar sus *workflows*.

Cuando más adelante se solicite a la iA generar el código de un *workflow*, será necesario especificar en el *prompt* la funcionalidad que se espera de él. Para que el *workflow* generado pueda invocar a las tareas del *task pool*, la iA debe conocerlas, por lo que será necesario proporcionar en el contexto la información de dichas tareas. Para ser más rigurosos, será preciso enriquecer el contexto únicamente con la información de aquellas tareas que, potencialmente, se necesiten ejecutar para alcanzar el objetivo del *workflow*. Para ello, será necesario aplicar técnicas de RAG (*Retrieval-Augmented Generation*) que permiten realizar búsquedas semánticas y recuperar las tareas más relacionadas con lo descrito en el *prompt*.

Por este motivo, la recopilación de información de las tareas también implica su almacenamiento en un *embedding store* de forma vectorizada.

SCIFI GatewAi utiliza Redis como *embedding store* y el modelo de *embeddings* *text-embedding-ada-002* que genera vectores de 1536 dimensiones. Además de los *embeddings* también se almacenan como metadatos *name*, *description* y *resource* de cada tarea. El *resource* de una tarea representa la *URN* que habría que especificar

en *SCIFI Workflow DSL* para referenciarla; esta *URN* tiene la forma: `urn:scifi:workflows:tasks:<name>` donde `<name>` corresponde al valor del name de la tarea.

2.3. Definición del API REST del *gateway* a partir de su especificación OpenAPI

SCIFI GatewAi expone una API REST que unifica el acceso a las APIs internas facilitando además la agregación de servicios. Para definir esta API REST se utiliza la especificación OpenAPI (OAS). Esta especificación es un estándar abierto cuyo propósito es describir APIs HTTP/REST de manera formal e independiente de cualquier lenguaje de programación. Su principal ventaja radica en que permite definir de forma precisa la interfaz de comunicación antes de comenzar con la implementación del software, lo que se conoce como el enfoque *API-first*.

SCIFI GatewAi no soporta toda la especificación sino únicamente un subconjunto de ella. A continuación, se describe los componentes de la especificación que se utilizan:

A nivel superior suele incluir estos campos: `openapi` (versión), `info`, `paths`, `components` y `tags`.

Código 4. Estructura de OpenAPI

```

openapi: 3.1.0
info:
  title: Ejemplo de API
  version: "1.0.0"
  description: API de ejemplo para ilustrar OpenAPI 3.1
paths: {}           # Aquí se incluye el core de la especificación
components:
  schemas: {}      # Esquemas JSON Schema reutilizables
tags: []           # (Opcional) taxonomía de operaciones

```

El objeto `paths` representa el grueso de la especificación. Describe todos los *endpoints* de la API. En sí `paths` es un mapa cuyas claves son rutas relativas que deben empezar por `/`. Se soporta *path templating*, esto es la capacidad de que las rutas contengan llaves `{}` para representar parámetros que de definirán posteriormente a nivel de operación (p. ej., `/users/{id}`).

Cada entrada de `paths` contiene un *path item object*, donde únicamente se debe declarar la operación HTTP: `get`, `put`, `post` o `delete`. Esta operación es un *operation object* donde, de forma obligatoria, se debe incluir `operationId`, `description` y `responses` que permiten definir, respectivamente, el identificador de la operación, la descripción de ésta y la respuesta que debe proporcionar. El elemento `description` es fundamental ya que, a partir de él, la IA generará el *workflow* a través del que se conseguirá implementar la operación. En función de las necesidades de la operación, el *operation object* opcionalmente puede incluir `parameters` y `requestBody`.

Código 5. Esquema de operation object

```

{
  "type": "object",
  "properties": {
    "operationId": { "type": "string" },
    "description": { "type": "string" },
    "parameters": {
      "type": "array", "items": { "$ref": "#/$defs/parameter" }
    },
    "requestBody": { "$ref": "#/$defs/request-body" },
    "responses": { "$ref": "#/$defs/responses" },
  },
  "required": ["operationId", "description", "responses"],
  "additionalProperties": false
}

```

En `parameters` se define un array conteniendo la especificación de cada parámetro. Los tipos de parámetros soportados son: `path`, `query` y `header`. Para describirlos se debe especificar un *parameter object* con la siguiente estructura:

Código 6. Esquema de parameter object

```

{
  "$comment": "https://spec.openapis.org/oas/v3.1#parameter-object",
  "type": "object",
  "required": ["name", "in"],
  "properties": {
    "name": {"type": "string"},
    "in": {"type": "string", "enum": ["query", "header", "path"]},
    "description": {"type": "string"},
    "required": {"type": "boolean", "default": false},
    "style": {"type": "string", "enum": ["matrix", "label", "form", "simple"]},
    "explode": {
      "type": "boolean",
      "description": "Controla si se repite la clave para arrays/objetos"
    },
  },
  "schema": {
    "$ref": "https://json-schema.org/draft/2020-12/schema",
    "description": "Esquema JSON con la estructura del parámetro"
  }
}

```

El valor del elemento `in` se emplea para describir el tipo del parámetro y condicionar el valor del atributo `style`:

Tabla 1. Valores de *in* y *style* en *parameter object*

in	style	
	Valor por defecto	Opciones permitidas
path	simple	simple, label, matrix
query	form	form
header	simple	simple

Conviene resaltar que los parámetros de tipo `path` (los que aparecen en la URL entre `{}`) tienen reglas estrictas:

- Cada expresión {param} en la ruta debe corresponderse con un *parameter object*.
- Dicho *parameter object* debe marcarse con: in: path y required: true

El valor de los parámetros podrá ser simple o complejo (un array o un objeto).

2.3.1. Parámetros simples

Código 7. Definición de parámetros simples de tipo path param: GET /alumnos/a33

```

/alumnos/{expediente}:
  get:
    description: |
      Obtén un objeto Alumno que represente el alumno identificado por el parámetro
      'expediente'. Devuelve únicamente el objeto Alumno.
    operationId: getAlumnoByExpediente
    parameters:
      - name: expediente
        in: path
        required: true
        description: Expediente del alumno
        style: simple
        explode: false
        schema:
          type: string
    responses:
      '200':
        description: Información del alumno
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Alumno'

```

Código 8. Definición de parámetros simples de tipo query param: GET /alumnos?expediente=a33

```

/alumnos/{expedientes}:
  get:
    description: |
      Obtén un objeto Alumno que representa el alumno identificado por el parámetro
      expediente. Devuelve únicamente el objeto Alumno.
    operationId: getAlumnoByExpediente
    parameters:
      - name: expediente
        in: query
        required: true
        description: Expediente del alumno
        style: form
        explode: true
        schema:
          type: string
    responses:
      '200':
        description: Información del alumno
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Alumno'

```

2.3.2. Parámetros complejos:

- a. Parámetros cuyo valor es un Array

Tabla 2. Valores de style y explode en path param de tipo Array

Atributo	style	explode	URL resultante
Valor	matrix	true	/alumnos/;p=123;p=456;p=789
		false	/alumnos/;p=123,456,789
	label	true	/alumnos/.123.456.789
		false	/alumnos/.123,456,789

Tabla 3. Valores de style y explode en query param de tipo Array

Atributo	style	explode	URL resultante
Valor	form	true	/alumnos?expedientes=a33&expedientes=a44
		false	/alumnos?expedientes=a33,a44,a55

- b. Parámetros cuyo valor representa un objeto

```
Objeto:  {
          "ap1": "Soto",
          "ap2": "Pozo"
        }
```

Tabla 4. Valores de style y explode en path param que representen objetos

Atributo	style	explode	URL resultante
Valor	matrix	true	/alumnos/;ap1=Soto;ap2=Pozo
		false	/alumnos/;apellidos=ap1,Soto,ap2,Pozo
	label	true	/alumnos/.ap1=Soto.ap2=Pozo
		false	/alumnos/.ap1,Soto,ap2,Pozo

Tabla 5. Valores de *style* y *explode* en *query param* que representen objetos

Atributo	style	explode	URL resultante
Valor	form	true	/alumnos?ap1=Soto;ap2=Pozo
		false	/alumnos?apellidos=ap1,Soto,ap2,Pozo

En lo que respecta a *requestBody* indicar que únicamente se admite los *media type* *application/json*, *application/x-www-form-urlencoded* y *text/plain*.

2.4. Implementación del API Rest

Por cada *operation* descrita en la especificación OpenAPI se creará un *workflow* que represente el flujo de trabajo a ejecutar cada vez que un cliente realice una petición a dicha *operation*. El JSON obtenido tras la ejecución del *workflow* será la respuesta que se le devolverá al cliente.

La generación del *workflow* lo realiza la iA en base a un *prompt* que incluye:

- el valor del elemento *description* de la *operation*. Este texto actuará a modo de requisito funcional del *workflow*;
- una descripción del dato que recibirá como *input* el *workflow*. Este dato será un objeto JSON con el siguiente JSON Schema:

Código 9. JSON Schema con la descripción del dato que recibe como *input* el *workflow*

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "Objeto con los parámetros y requestBody enviados en la request",
  "type": "object",
  "properties": {
    "parameters": {
      "type": "object",
      "description": "Objeto con tantos atributos como parámetros incluidos en la request",
      "additionalProperties": {
        "description": "Valor de cada atributo, puede ser cualquier tipo de dato",
        "type": ["string", "number", "integer", "boolean", "array", "object"]
      }
    },
    "requestBody": {
      "type": "object",
      "description": "Objeto JSON que representa el cuerpo de la petición",
      "additionalProperties": true
    }
  },
  "additionalProperties": false
}
```

- un *yml* con la información de tareas que se considera útiles. Para obtener los datos se emplean técnicas *RAG*, de forma que se obtiene del *embedding store* la información con las tareas que se estima pertinentes en base al contexto definido por la *description* de la *operation*;

- un JSON conteniendo los *JSON schemas* correspondientes a los datos de entrada y salida de cada tarea, así como otros *JSON schema* a los que se pudiera hacer referencia.

Por ejemplo, dada la siguiente *API REST* (Código 10) y asumiendo que se tiene definida la tarea `findMatriculacionAlumnoByExpediente`, se confeccionaría el *prompt* que se muestra en Código 11.

Código 10. OpenAPI del Gateway

```

openapi: 3.1.0
info:
  title: API de Ejemplo con path params
  version: 1.0.0
  description: API de ejemplo.
paths:
  /alumnos/{expediente}/matriculas:
    get:
      description: |
        Obten el objeto MatriculacionAlumno que representa los codigos de las asignaturas de que está
        matriculado el alumno identificado por el parámetro 'expediente' así como la calificación obtenida en
        cada una. Devuelve únicamente el objeto MatriculacionAlumno.
      operationId: getMatriculacionByExpediente
      parameters:
        - name: expediente
          in: path
          required: true
          description: Expediente del alumno
          schema:
            type: string
      responses:
        '200':
          description: Matriculación del alumno
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/MatriculacionAlumno'

```

Código 11. Prompt enviado a la iA

```

task pool
@Tasks(rest={
    @RestTask(name="findMatriculacionAlumnoByExpediente",
        description=""
        Se obtiene el 'codigo' y 'calificacion' de las asignaturas en las que se
        ha matriculado el alumno identificado por 'expediente'.
        """"
        inputChannel=@TaskChannel(namespace="scifi", type="IdAlumno", version = "1.0.0",
            description="Objeto conteniendo el expediente del alumno"),
        outputChannel=@TaskChannel(namespace="scifi", type="MatriculacionAlumno",
            version="1.0.0",
            description=""
            Objeto que describe de qué asignaturas se ha
            matriculado el alumno así como las
            calificaciones obtenidas
            """"
        ),
        parameters=@RestTaskParameters(method=RestBuiltInTask.Method.GET,
            endpoint$="Scifi.Config('alumnos.url')",
            path="alumnos/{expediente}/matriculas",
            pathParams={
                @Param(name="expediente",
                    value="$expediente"
                )
            }
        )
    )
})

Prompt
requisitos:
Obten el objeto MatriculacionAlumno que representa los codigos de las asignaturas de que está matriculado
el alumno identificado por el parámetro 'expediente' así como la calificacion obtenida en cada una.
Devuelve unicamente el objeto MatriculacionAlumno.

El input de la máquina de estados es un objeto JSON RestRequest con los siguientes atributos:
- 'parameters': Objeto JSON cuyos atributos representan los parámetros enviados por el cliente.
    El objeto puede estar vacío.
- 'requestBody': Objeto JSON con el dato enviado por el cliente. El objeto puede estar vacío.

tasks:
- name: "findMatriculacionAlumnoByExpediente"
  description: |
    Se obtiene el 'codigo' y 'calificacion' de las asignaturas en las que se ha
    matriculado el alumno identificado por 'expediente'."
  resource: urn:scifi:workflows:tasks:findMatriculacionAlumnoByExpediente
  input:
    name: http://apicurio:8091/apis/registry/v2/groups/scifi/artifacts/IdAlumno/versions/1.0.0
    description: Objeto Json conteniendo el expediente del alumno
    namespace: scifi
    type: IdAlumno
    version: 1.0.0
  output:
    name: http://apicurio:8091/apis/registry/v2/groups/scifi/artifacts/MatriculacionAlumno/versions/1.0.0
    description: |
      Objeto que describe de qué asignaturas se ha matriculado el alumno así como las
      calificaciones obtenidas
    namespace: scifi
    type: MatriculacionAlumno
    version: 1.0.0

schemas:
{
  "http://apicurio:8091/apis/registry/v2/groups/scifi/artifacts/IdAlumno/versions/1.0.0":
  {
    "$id": "http://apicurio:8091/apis/registry/v2/groups/scifi/artifacts/IdAlumno/versions/1.0.0",
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "IdAlumno",
    "type": "object",
    "properties": {
      "expediente": {"type": "string"},
      "formatAlumno": {
        "type": "string",
        "enum": ["uppercase", "lowercase", "doctor", "unmodified"],
        "default": "unmodified",
      }
    },
    "required": ["expediente"]
  },
  "http://localhost:8091/apis/registry/v2/groups/scifi/artifacts/MatriculacionAlumno/versions/1.0.0":
  {
    "$id": "http://localhost:8091/apis/registry/v2/groups/scifi/artifacts/MatriculacionAlumno/versions/1.0.0",
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "MatriculacionAlumno",
  }
}

```

```

"type": "object",
"properties":
{
  "expediente": {"type": "string"},
  "matriculas":
  {
    "type": "array",
    "items":
    {
      "$ref": "http://localhost:8091/apis/registry/v2/groups/scifi/artifacts/Matricula/versions/1.0.0"
    }
  }
},
"required": ["expediente", "matriculas"]
},
"http://localhost:8091/apis/registry/v2/groups/scifi/artifacts/Matricula/versions/1.0.0":
{
  "$id": "http://localhost:8091/apis/registry/v2/groups/scifi/artifacts/Matricula/versions/1.0.0",
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Matricula",
  "type": "object",
  "properties": {
    "codigo": {"type": "string"},
    "calificacion": {"type": "number", "format": "float"}
  },
  "required": ["codigo", "calificacion"]
}
}

```

2.5. Ejecución de peticiones que implican secuenciación de tareas y agregación de datos

Con el propósito de verificar la agregación de servicios, se ha diseñado el siguiente ejemplo. En él se dispone de un *task pool* en el que se incluye tareas que representan APIs REST (Código 12, Código 13 y Código 14) mediante las que se obtiene información de alumnos y asignaturas concretos referenciados por sus respectivos identificadores.

Código 12. Task findAlumnoByExpediente de tipo API Rest

```

@RestTask(
  name="findAlumnoByExpediente",
  description="Obtiene el alumno identificado por su expediente.",
  inputChannel=@TaskChannel(namespace = "scifi", type = "IdAlumno", version = "1.0.0",
    description = "Objeto con el expediente del alumno"
  ),
  outputChannel=@TaskChannel(namespace = "scifi", type = "Alumno", version = "1.0.0",
    description = "Información del alumno"),
  parameters=@RestTaskParameters(endpoint$ = "Scifi.Config('alumnos.url')",
    path="alumnos/{expediente}",
    pathParams=@Param(name$="expediente", value="$expediente"),
    method=RestBuiltInTask.Method.GET
  )
)

```

path template	Ejemplo	Respuesta
/alumnos/{expediente}	/alumnos/1111	{ "expediente": "1111", "nombre": "Carlos Pérez" "fechaNacimiento": "2002-02-11" }

Código 13. Task findAsignaturaByCodigo de tipo API Rest

```

@RestTask(name="findAsignaturaByCodigo",
  description="Obtiene la asignatura identificada por su codigo.",
  inputChannel=@TaskChannel(namespace="scifi", type="IdAsignatura", version="1.0.0",
    description = "Objeto con el codigo de la asignatura"
  ),
  outputChannel=@TaskChannel(namespace="scifi", type="Asignatura", version="1.0.0",
    description="Información de la asignatura"
  ),
  parameters=@RestTaskParameters(endpoint$="Scifi.Config('asignaturas.url')",
    path="asignaturas/{codigo}",
    pathParams=@Param(name$="codigo", value="$codigo"),
    method = RestBuiltInTask.Method.GET
  )
)

```

path template	Ejemplo	Respuesta
/asignaturas/{codigo}	/asignaturas/bbbb	{ "codigo": "bbbb", "nombre": "Sistemas BBDD-II", "creditos": 6, "profesor": "Roberto" }

Código 14. Task findMatriculacionByExpediente de tipo API Rest

```

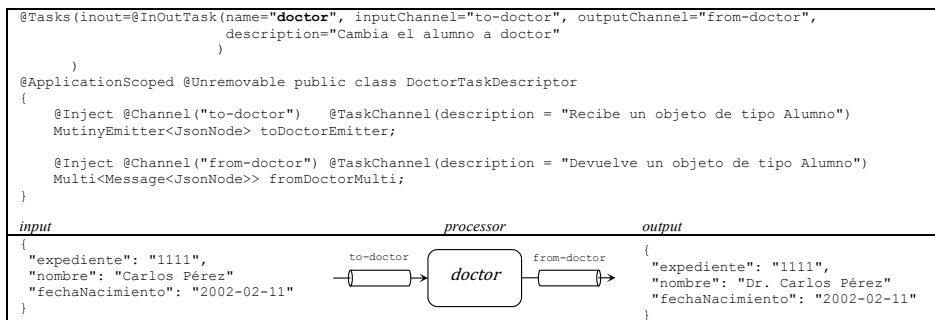
@RestTask(name="findMatriculacionByExpediente",
  description=""
  Obtene las codigos de las asignaturas en la que se ha matriculado un alumno
  identificado por su 'expediente'. Además por cada asignatura se indica la
  calificación que ha obtenido el alumno.
  "",
  inputChannel=@TaskChannel(namespace="scifi", type="IdAlumno", version="1.0.0",
    description = "Objeto con el expediente del alumno"
  ),
  outputChannel=@TaskChannel(namespace="scifi", type="MatriculacionAlumno", version="1.0.0",
    description = ""
    Objeto con las asignaturas se ha matriculado el alumno
    y las calificaciones obtenidas
    ""
  ),
  parameters=@RestTaskParameters(endpoint$="Scifi.Config('alumnos.url')",
    path="alumnos/{expediente}/matriculas",
    pathParams=@Param(name$="expediente", value="$expediente"),
    method=RestBuiltInTask.Method.GET
  )
)

```

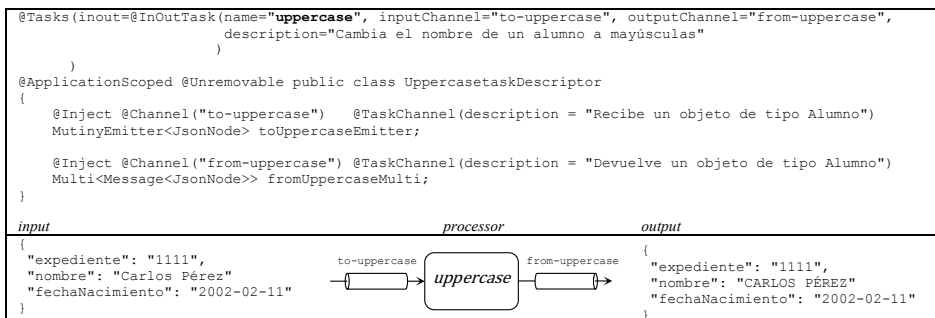
Path template	Ejemplo	Respuesta
/alumnos/{expediente}/matriculas	/alumnos/1111/matriculas	{ "expediente": "1111", "matriculas": [{ "codigo": "bbbb", "calificación": 7.5 }, { "codigo": "cccc", "calificación": 6 }] }

Además, se dispone también de otras tareas (Código 15, Código 16 y Código 17) de tipo inout (que representan *processors Kafka*) que realizan modificaciones respecto a los datos de un alumno.

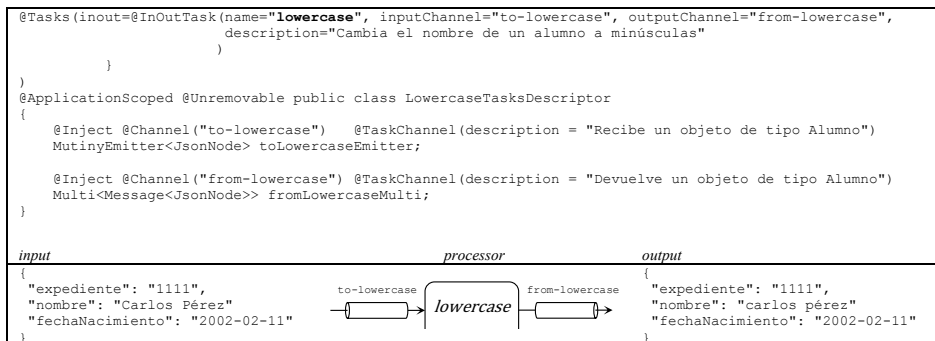
Código 15. Task doctor de tipo inout basadas en processors Kafka



Código 16. Task uppercase de tipo inout basadas en processors Kafka



Código 17. Task lowercase de tipo inout basadas en processors Kafka



En base a estas tareas, se quiere que el *gateway* incluya una funcionalidad que permita obtener los datos completos referentes a la matriculación de un alumno. En lugar de obtener sólo los datos proporcionados por la tarea `findMatriculacionByExpediente` se quiere que estén agregados junto con los proporcionados por las tareas `findAlumnoByExpediente` y `findAsignaturaByCodigo`. Además, también se busca transformar el alumno a través de las tareas `doctor`, `uppercase` o `lowercase`.

Funcionalidad <code>findMatriculacionByExpediente</code>	Funcionalidad buscada a través del <i>gateway</i>
<pre> { "expediente": "1111", "matriculas": [{ "codigo": "bbbb", "calificación": 7.5 }, { "codigo": "cccc", "calificación": 6 }] } </pre>	<pre> { "alumno": { "expediente": "1111", "nombre": "Dr. nombre alumno 1" "fechaNacimiento": "2002-02-11" }, "matriculas": [{ "asignatura": { "codigo": "bbbb", "nombre": "Sistemas BBDD-II", "creditos": 6, "profesor": "Roberto" }, "calificación": 7.5 }, { "asignatura": { "codigo": "cccc", "nombre": "Desarrollo SSI", "creditos": 6, "profesor": "Roberto" }, "calificación": 6 }] } </pre>

Para ello se define el OpenAPI del recurso del *gateway* (Código 18) en el que el recurso `/alumnos/{expediente}/matriculasFull?format=[uppercase|lowercase|doctor|unmodified]` en el que como parámetros se especifica el identificador del alumno (*path param* `expediente`) y el formato en al que se desea transformar los datos de éste (*query param* `format` con valores `doctor`, `uppercase` o `lowercase`).

Código 18. OpenAPI de la operation `/alumnos/{expediente}/matriculasFull`

```

openapi: 3.1.0
info:
  title: API de Ejemplo con path params
  version: 1.0.0
  description: API de ejemplo.
paths:
  /alumnos/{expediente}/matriculasFull:
    get:
      description: |
        Obten el objeto MatriculacionFullAlumno que representa las asignaturas de que está matriculado el alumno identificado por el parametro 'expediente' así como las calificaciones obtenidas en cada una. Devuelve únicamente el objeto MatriculacionFullAlumno.
      summary: |
        Obten el objeto Alumno correspondiente al alumno identificado por el parámetro expediente
      operationId: getMatriculacionFullByExpediente
      parameters:
        - name: expediente
          in: path
          required: true
          description: Expediente del alumno
          schema:
            type: string
        - name: formatAlumno
          in: query
          description: Formato en el que se desea obtener el alumno
          required: false
          schema:
            type: string
            enum: [uppercase, lowercase, doctor, unmodified]
            default: unmodified
      responses:
        "200":
          description: Matriculación del alumno
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/MatriculacionFullAlumno'

```


Código 20. Tarea `mapMatriculacionToMatriculasFull` de tipo `stateMachine`

<pre> @StateMachineTask (name="mapMatriculacionToMatriculasFull", description="Transforma un objeto de tipo MatriculacionAlumno en un objeto MatriculasFull", inputChannel=@TaskChannel(namespace="scifi", type="MatriculacionAlumno", version="1.0.0", description="Objeto MatriculacionAlumno"), outputChannel=@TaskChannel(namespace="scifi", type="MatriculasFull", version="1.0.0", description="Objeto MatriculasFull"), stateMachine = """ El input es un objeto MatriculacionAlumno Debes secuenciar los siguientes pasos: 1. A través de un estado de tipo Map debes iterar por cada elemento del array del atributo "matriculas" del input de forma que: 1.a. Su "ItemsPath" debe ser "\$.matriculas" 1.b. A través de "ItemSelector" se define un objeto Matricula con los atributos: - 'codigo' con el valor "\$\$.Map.Item.Value.codigo". - 'calificacion' con el valor "\$\$.Map.Item.Value.calificacion". 1.c. Debes transformar cada objeto Matricula en un objeto MatriculaFull. 1.d. El ResultPath debe ser \$.arrMatriculasFull 2. A continuación, a través un estado Pass crea y devuelve un objeto MatriculasFull: 2.a. A través de Parameters crea un objeto que contenga como atributo: - 'matriculas' con el valor de \$.arrMatriculasFull 2.b. El ResultPath debe ser \$ """, generatorType = StateMachineGenerator.Type.PROMPT) </pre>		
<p>Workflow generado:</p> <pre> { "Comment": "StateMachine para transformar un array de Matricula en un array de MatriculaFull y devolver un objeto MatriculasFull", "StartAt": "MapearMatriculas", "States": { "MapearMatriculas": { "Type": "Map", "ItemsPath": "\$.matriculas", "ItemSelector": { "codigo.\$": "\$\$.Map.Item.Value.codigo", "calificacion.\$": "\$\$.Map.Item.Value.calificacion" }, "ItemProcessor": { "ProcessorConfig": { "Mode": "INLINE" }, "StartAt": "TransformarMatriculaEnMatriculaFull", "States": { "TransformarMatriculaEnMatriculaFull": { "Type": "Task", "Resource": "urn:scifi:workflows:tasks:mapMatriculaToMatriculaFull", "End": true } } }, "ResultPath": "\$.arrMatriculasFull", "Next": "ConstruirMatriculasFull" }, "ConstruirMatriculasFull": { "Type": "Pass", "Parameters": { "matriculas.\$": "\$.arrMatriculasFull" }, "ResultPath": "\$", "End": true } } } </pre>		
input	processor	output
<pre> { "expediente": "1111", "matriculas": [{ "codigo": "bbbb", "calificación": 7.5 }, { "codigo": "cccc", "calificación": 6 }] } </pre>		<pre> { "matriculas": [{ "asignatura":{ "codigo": "bbbb", "nombre": "Sistemas BBDD II", "creditos": 6, "profesor": "Roberto" }, "calificación": 7.5 }, { "asignatura":{ "codigo": "cccc", "nombre": "Desarrollo SSI ", "creditos": 6, "profesor": "Roberto" }, "calificación": 6 }] } </pre>

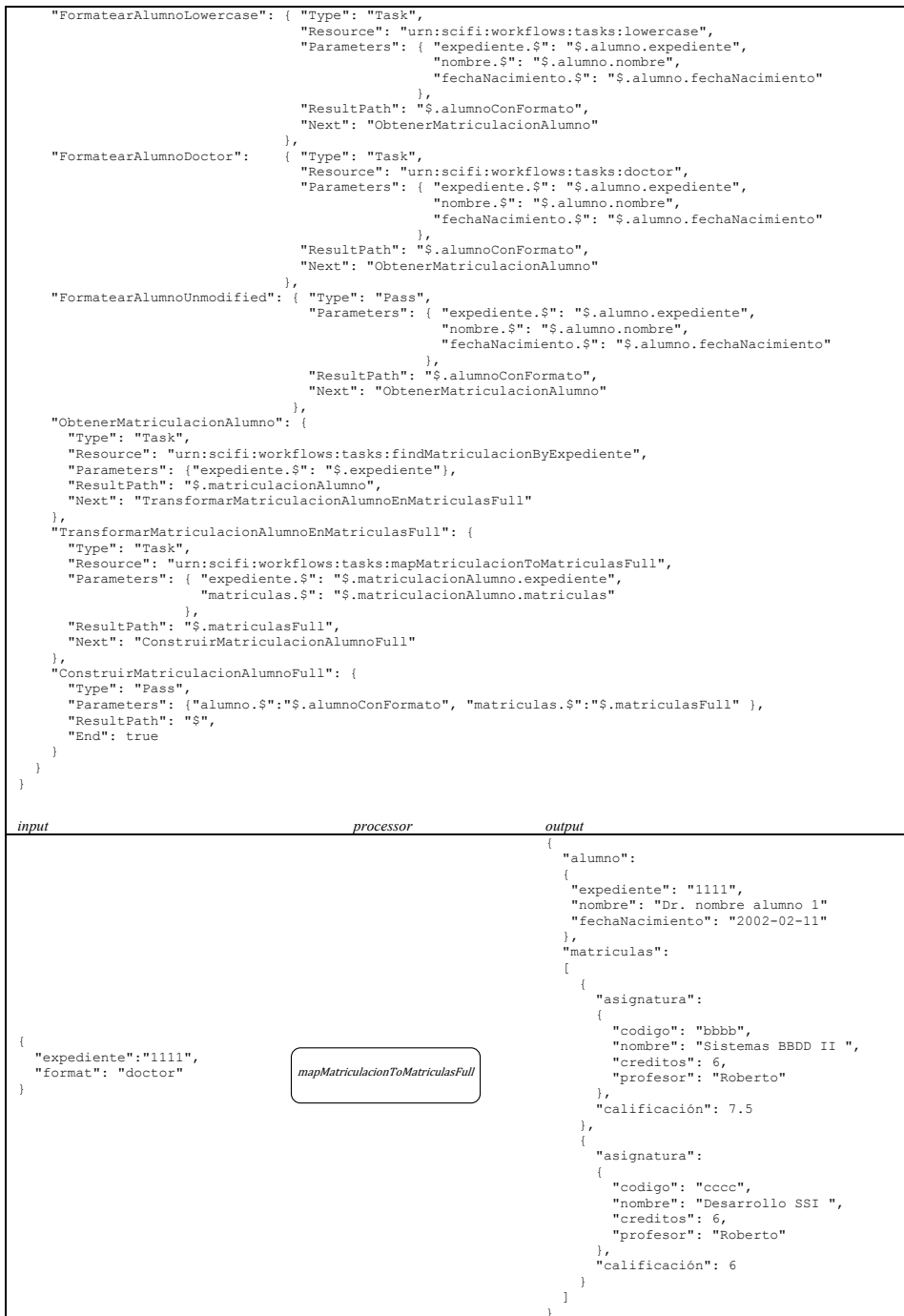
Código 21. Tarea findMatriculacionFullByExpediente de tipo stateMachine

```
@StateMachineTask
(
  name="findMatriculacionFullByExpediente",
  description="""
    Obtiene el objeto MatriculacionFullAlumno que representa las asignaturas de que está
    matriculado el alumno identificado por el atributo 'expediente' de su input.
    """
  inputChannel=@TaskChannel(namespace="scifi", type="IdAlumno", version="1.0.0",
    description = "Objeto que identifica a un alumno"
  ),
  outputChannel=@TaskChannel(namespace="scifi", type="MatriculacionFullAlumno", version="1.0.0",
    description = "Objeto MatriculacionFullAlumno"
  ),
  stateMachine="""
    El input es un objeto IdAlumno.
    Debes secuenciar las siguientes tareas:
    1. Ejecuta una tarea que recupere el Alumno identificado por su expediente.
    Para ello:
    1.a. A través de Parameters crea un objeto que contenga como atributo:
    - 'expediente': con el valor del atributo 'expediente' del input.
    1.b. El ResultPath debe ser $.alumno
    2. A continuación ejecuta una tarea Choice en la que debes comprobar el valor del
    atributo 'formatAlumno' del input de forma que:
    2.a Si es igual al literal 'uppercase' debes ejecutar una tarea a través de
    la que se cambie el nombre del alumno a mayúsculas en la que:
    2.a.1. En Parameters crea un objeto Alumno igual al Alumno $.alumno.
    2.a.2. El ResultPath debe ser $.alumnoConFormato.
    2.b Si es igual al literal 'lowercase' debes ejecutar una tarea a través de
    la que se cambie el nombre del alumno a minúsculas en la que:
    2.a.1. En Parameters crea un objeto Alumno igual al Alumno $.alumno.
    2.a.2. El ResultPath debe ser $.alumnoConFormato.
    2.c Si es igual al literal 'doctor' debes ejecutar una tarea a través de la
    que se cambie el alumno a doctor en la que:
    2.c.1. En Parameters crea un objeto Alumno igual al Alumno $.alumno.
    2.c.2. El ResultPath debe ser $.alumnoConFormato.
    2.d En cualquier otro caso, debes ejecutar una tarea Pass en la que:
    2.d.1. En Parameters crea un objeto Alumno igual al Alumno $.alumno.
    2.d.2. El ResultPath debe ser $.alumnoConFormato.
    3. A continuación, recupera los códigos de las asignaturas de la que se ha matriculado un
    alumno así como las calificaciones obtenidas en cada una. Para ello:
    3.a. A través de Parameters crea un objeto que contenga como atributo:
    - 'expediente': cuyo valor es el valor del atributo 'expediente' del input
    3.b. El ResultPath debe ser $.matriculacionAlumno
    4. A continuación, ejecuta una tarea que transforme el objeto MatriculacionAlumno en un
    objeto MatriculasFull. Para ello:
    4.a. A través de Parameters crea un objeto que contenga como atributos:
    - 'expediente': con el valor del atributo '$.matriculacionAlumno.expediente'.
    - 'matriculas': con el valor del atributo '$.matriculacionAlumno.matriculas'.
    4.b. El ResultPath debe ser $.matriculasFull
    5. Finalmente, ejecuta un Pass que cree un objeto MatriculacionAlumnoFull. Para ello:
    5.a. A través de Parameters crea un objeto que contenga como atributos:
    - 'alumno': cuyo valor debe ser $.alumnoConFormato
    - 'matriculas': cuyo valor debe ser $.matriculasFull
    5.b. El ResultPath debe ser $
    """
  generatorType = StateMachineGenerator.Type.PROMPT
)

```

Workflow generado:

```
{
  "Comment": "StateMachine para obtener información de un alumno, formatear su nombre según el input,
  obtener su matriculación y devolver un objeto MatriculacionAlumnoFull",
  "StartAt": "ObtenerAlumnoPorExpediente",
  "States": {
    "ObtenerAlumnoPorExpediente": { "Type": "Task",
      "Resource": "urn:scifi:workflows:tasks:findAlumnoByExpediente",
      "Parameters": { "expediente.$": "$.expediente" },
      "ResultPath": "$.alumno",
      "Next": "FormatearAlumnoChoice"
    },
    "FormatearAlumnoChoice": {
      "Type": "Choice",
      "Choices": [ { "Variable": "$.formatAlumno", "StringEquals": "uppercase",
        "Next": "FormatearAlumnoUppercase"
      },
        { "Variable": "$.formatAlumno", "StringEquals": "lowercase",
        "Next": "FormatearAlumnoLowercase"
      },
        { "Variable": "$.formatAlumno", "StringEquals": "doctor",
        "Next": "FormatearAlumnoDoctor"
      }
    ],
    "Default": "FormatearAlumnoUnmodified"
  },
  "FormatearAlumnoUppercase": { "Type": "Task",
    "Resource": "urn:scifi:workflows:tasks:uppercase",
    "Parameters": { "expediente.$": "$.alumno.expediente",
      "nombre.$": "$.alumno.nombre",
      "fechaNacimiento.$": "$.alumno.fechaNacimiento"
    },
    "ResultPath": "$.alumnoConFormato",
    "Next": "ObtenerMatriculacionAlumno"
  },
}
```



En base a este *task pool*, y teniendo en cuenta la descripción de la *operation* /alumnos/{expediente}/matriculasFull?format, el *workflow* generado por la iA se muestra en Código 22.

Código 22. Workflow generado para la operation /alumnos/{expediente}/matriculasFull?format=

```
{
  "Comment": "StateMachine para obtener el objeto MatriculacionFullAlumno de un alumno identificado por los parámetros 'expediente' y 'formatAlumno' recibidos en RestRequest.parameters.",
  "StartAt": "PrepararInputIdAlumno",
  "States": {
    "PrepararInputIdAlumno": {
      "Type": "Pass",
      "Parameters": {
        "expediente.$": "$.parameters.expediente",
        "formatAlumno.$": "$.parameters.formatAlumno"
      },
      "ResultPath": "$.IdAlumno",
      "Next": "ObtenerMatriculacionFullAlumno"
    },
    "ObtenerMatriculacionFullAlumno": {
      "Type": "Task",
      "Resource": "urn:scifi:workflows:tasks:findMatriculacionFullByExpediente",
      "Parameters": {
        "expediente.$": "$.IdAlumno.expediente",
        "formatAlumno.$": "$.IdAlumno.formatAlumno"
      },
      "ResultPath": "$.MatriculacionFullAlumno",
      "OutputPath": "$.MatriculacionFullAlumno",
      "End": true
    }
  }
}
```

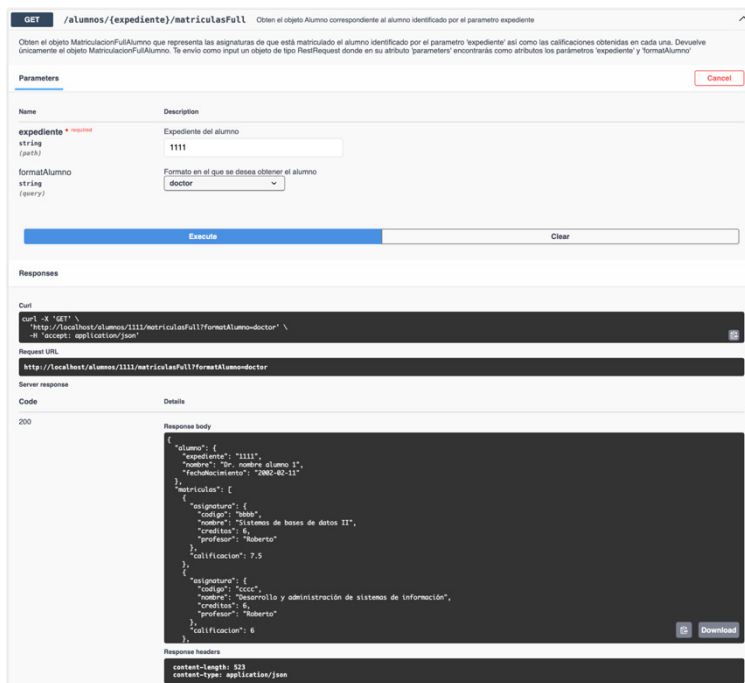
Una petición a /alumnos/1111/matriculasFull?format=doctor desemboca en la ejecución del *workflow* anterior al que como input se le pasa:

Código 23. Input correspondiente a la petición /alumnos/1111/matriculasFull?format=doctor

```
{
  "parameters": {
    "expediente": "1111",
    "format": "doctor"
  },
  "requestBody": {}
}
```

La ejecución de esta petición produce el resultado mostrado en la Figura 1.

Figura 1. Ejecución de la operation del gateway /alumnos/{expediente}/matriculasFull



3. CONCLUSIÓN

Se ha desarrollado SCIFI GatewAi, un prototipo funcional de un *gateway* inteligente orientado a la agregación de servicios en arquitecturas basadas en microservicios. A diferencia de los *gateways* tradicionales, esta solución integra técnicas de Inteligencia Artificial Generativa y modelos de lenguaje de gran escala (LLM) que posibilitan la generación automática de flujos de trabajo a partir de descripciones en lenguaje natural. De este modo, se elimina la necesidad de programar herramientas adicionales, lo que supone un avance significativo en cuanto a la simplificación del diseño, implementación, despliegue y mantenimiento de aplicaciones *IoT* y *cloud*.

Los resultados obtenidos demuestran que SCIFI GatewAi no solo es capaz de centralizar y enrutar peticiones, sino también de combinar dinámicamente respuestas de distintos microservicios para atender solicitudes con respuestas complejas. Esto reduce la latencia, mejora la experiencia de usuario y potencia la escalabilidad de las aplicaciones. Además, la utilización de estándares como OpenAPI disminuye la curva de aprendizaje lo que favorece su utilización.

Por tanto, SCIFI GatewAi representa una innovación a la hora de definir un *gateway* aportando ventajas en términos de eficiencia, adaptabilidad y facilidad de uso. Su carácter extensible y su capacidad de evolucionar con nuevos requisitos funcionales lo convierten en una solución con un gran potencial de aplicación en escenarios reales.

REFERENCIAS BIBLIOGRÁFICAS

Amazon states language. 2025. <https://states-language.net/spec.html>.

AWS Step Functions. 2025. <https://aws.amazon.com/es/step-functions/>.

Apache Kafka. <https://kafka.apache.org/>.

Fowler M. Event Sourcing. 2021.

<https://martinfowler.com/eaaDev/EventSourcing.html>.

JSONPath working group. JSONPath: query expressions for json. 2023. <https://ietf-wg-jsonpath.github.io/draft-ietf-jsonpath-base/draft-ietf-jsonpath-base.html>.

Kong Gateway. 2025. <https://konghq.com/products/kong-gateway>. 2025.

LangChain4j. 2025. <https://docs.langchain4j.dev/>

Lewis P, Perez E, Piktus A *et al*. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* 2020, 33, 9459-9474.

Nginx. 2025. <https://nginx.org>.

Quarkus LangChain4j. 2025. <https://docs.quarkiverse.io/quarkus-langchain4j/dev/index.html>

Red Hat Inc. Apicurio. 2025. <https://www.apicur.io/registry>.

Red Hat, Inc. Quarkus. 2025. <https://quarkus.io/>.

Roper J, Escoffier C, Hutchison G. Microprofile reactive messaging specification. 2022. <https://download.eclipse.org/microprofile/microprofile-reactive-messaging-3.0/microprofile-reactive-messaging-spec-3.0.pdf>. 2025.

Traefik. 2025. <https://traefik.io>.

Vaswani A, Shazeer N, Parmar, N *et al*. Attention is all you need. *Advances in neural information processing systems* 2017; 30.

SISTEMA PARA LA GESTIÓN EFICIENTE DE RECURSOS MATERIALES DEPORTIVOS (SISGER)



PELAYO DÍEZ FERNÁNDEZ

ESTHER SÁNCHEZ MORO

PABLO ELIPE LORENZO

DAVID REVESADO CARBALLARES

SERGIO LÓPEZ GARCÍA

RESUMEN

El proyecto “Sistema para la Gestión Eficiente de Recursos Materiales Deportivos” (SISGER) pretende desarrollar una aplicación informática que permita optimizar la gestión de materiales deportivos de la Facultad de Educación de la Universidad Pontificia de Salamanca. Para ello, los usuarios deberán solicitar materiales deportivos con al menos 48 horas de antelación, priorizando las necesidades docentes sobre las de investigación. Las solicitudes serán validadas por la Coordinación de Recursos e Infraestructuras de la Facultad. Tras efectuar la solicitud, los usuarios recibirán una notificación en la aplicación y en su email acerca de la disponibilidad del material y la fecha y lugar de recogida. La aplicación también se encargará de gestionar las distintas incidencias que puedan girar en torno a este material, pudiendo penalizar a los usuarios que no respeten el protocolo establecido desde la institución. Además, el sistema incluirá un control exhaustivo sobre la entrega y devolución de los materiales, asegurando su localización y evitando extravíos, un hecho que contribuirá a un uso eficiente y responsable de los recursos deportivos.

PALABRAS CLAVE: instalaciones; recursos; reserva; software

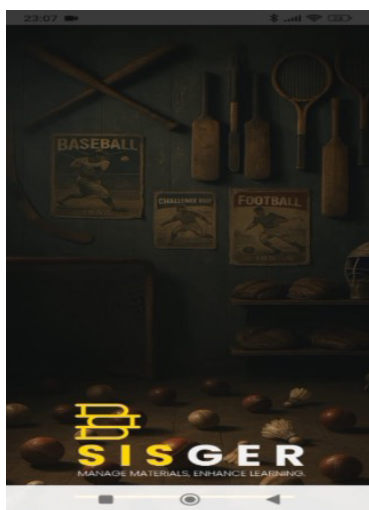
1. CONTEXTUALIZACIÓN DEL PROYECTO

Sistema para la Gestión Eficiente de Recursos Materiales Deportivos (SISGER) se trata de un proyecto presentado en la convocatoria del programa TALENT PRO-2024-2025 de la Universidad Pontificia de Salamanca. Esta iniciativa, está enmarcada dentro del Plan de Transferencia de Conocimiento Universidad-Empresa (Plan TCUE) 2024-2027, que cuenta con el apoyo de la Fundación Universidades y Enseñanzas Superiores de Castilla y León (FUESCYL), siendo un claro ejemplo de innovación y repercusión científica en el seno de las instituciones de educación superior.

Además, el proyecto se encuentra alineado con los intereses de sectores estratégicos identificados en la RIS3 2021-2027 de Castilla y León, lo que asegura su pertinencia y utilidad en ámbitos clave para el desarrollo económico y social de la región. Concretamente, SISGER está relacionado con el ámbito número 4 (Tecnologías y Ciberseguridad), ya que se centra en el desarrollo de una aplicación tecnológica para la gestión eficiente de recursos materiales deportivos. Este enfoque implica el uso de herramientas digitales avanzadas, gestión de datos y procesos innovadores, alineándose con la promoción de soluciones tecnológicas dentro de la estrategia RIS3. También podríamos vincularlo de manera indirecta con el ámbito número 3 (Salud y atención social), dado que fomenta el uso adecuado de recursos deportivos que, a su vez, facilitan el desarrollo de actividades que promueven la salud y el bienestar en entornos de carácter educativo, deportivo y recreativo (Cabello y Cabra, 2006).

La propuesta que hemos desarrollado es una aplicación informática que mejora la gestión eficiente de los recursos materiales de la Facultad de Educación de la Universidad Pontificia de Salamanca (figura 1).

Figura 1. *Aplicación SISGER*



Fuente: Elaboración propia

Se trata, sin lugar a dudas, de una medida de actuación necesaria para gestionar y planificar los distintos incidentes que se desarrollan cotidianamente en cualquier tipo de instalación deportiva (Celma, 2020). Asimismo, los futuros profesionales del sector deportivo, deben de desarrollar, además de conocimientos técnicos, competencias de liderazgo y comunicación, gestión de recursos humanos y materiales (Jiménez-Saiz et al., 2015; Tafalla et al., 2005).

En nuestro caso, debemos señalar que la Facultad de Educación cuenta actualmente con tres titulaciones de grado: Maestro de Educación Infantil, Maestro de Educación Primaria y Ciencias de la Actividad Física y del Deporte. Es esta última, por su idiosincrasia, es la que demanda un mayor volumen de recursos materiales para su actividad docente y, por tanto, la que requiere una mayor planificación de los mismos. Esta titulación ha contado durante el curso académico 2024-2025 con 385 alumnos matriculados. Al mismo tiempo, el equipo docente ascendía a un total de 36 profesores. Frente a este contexto, nuestra aplicación cobra una especial relevancia, ya que va a ayudar a la Coordinación de Recursos e Infraestructuras de la Facultad a realizar una gestión más eficaz y eficiente de los materiales deportivos.

2. JUSTIFICACIÓN Y OBJETIVOS

Nos encontramos ante una aplicación necesaria para el correcto funcionamiento de la Facultad de Educación de la Universidad Pontificia de Salamanca, ya que optimiza la reserva de instalaciones y el préstamo de material deportivo. En este sentido, debemos reconocer que la pertinencia y utilidad de SISGER se asienta sobre varios aspectos. Por un lado, permite gestionar todos los recursos materiales del centro, lo que favorece que los usuarios conozcan en tiempo real la disponibilidad de los mismos. Por otro lado, incrementa la autonomía del profesorado, que dispondrá de mayores facilidades para planificar sus sesiones, en base al material disponible. Y, por último, ya que permite agilizar las labores de mantenimiento, abriendo un nuevo canal de comunicación donde cualquier usuario podrá informar sobre el estado de un material, posibilitando que, desde la Coordinación de Recursos, se lleve a cabo una reparación oportuna de este recurso.

Además, nos gustaría resaltar que se trata de un proyecto viable desde el punto de vista técnico y económico. Técnicamente, ya que se basa en el desarrollo de una aplicación que se valdrá de tecnologías ampliamente utilizadas, lo que garantizará un funcionamiento eficiente. Y desde el punto de vista económico, dado que no se trata de una herramienta especialmente costosa. La aplicación te ofrece multitud de servicios con una inversión mínima, lo que ayudará a un desarrollo e implementación de la misma asequible.

Por otra parte, cabe reseñar que SISGER se trata de una herramienta con una gran capacidad innovadora ya que permite digitalizar y optimizar el préstamo de materia-

les deportivos. Y es que, en la actualidad, la Facultad de Educación no cuenta con una herramienta específica que gestione de manera integral estos recursos, teniendo que emplear una metodología más convencional. De este modo, la aplicación introduce una solución digitalizada que permite centralizar y automatizar estas tareas, mejorando la planificación y el uso de los recursos.

Para finalizar, la aplicación se adapta a las necesidades de los usuarios, ya que no solo permite reservar materiales, sino que también ofrece funciones personalizadas como la validación automática de solicitudes, notificaciones personalizadas y control de devoluciones. Del mismo modo, permite notificar daños o pérdidas del material en tiempo real, además de ofrecer un seguimiento detallado del mismo, reduciendo la probabilidad de pérdidas y mejorando la trazabilidad de los recursos. De esta manera los recursos estarán siempre localizados y en un estado óptimo, algo fundamental en un entorno educativo donde la disponibilidad y el estado del material impactan directamente en la calidad de los procesos de enseñanza y aprendizaje.

Por todo lo expuesto, nos gustaría destacar que el objetivo principal que se pretende alcanzar para el desarrollo de esta propuesta no es otro que el de diseñar y desarrollar una aplicación que permita optimizar la gestión eficiente de materiales deportivos de la Facultad de Educación de la Universidad Pontificia de Salamanca. Para alcanzarlo, se plantean los siguientes objetivos secundarios:

- Digitalizar el proceso de reserva de los materiales eliminando el uso de metodologías convencionales.
- Mejorar la localización, planificación y disponibilidad de los materiales deportivos.
- Fomentar el uso responsable de estos deportivos.

3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN

El diseño de la aplicación está orientado a que los usuarios puedan solicitar un material deportivo para una fecha determinada. En este sentido, y tal y como establece el protocolo de reserva de materiales de la Facultad, prevalecerá la solicitud de recursos para actividad docente sobre la investigación. Se demanda que las reservas se efectúen con un mínimo de 48 horas de antelación, para que la Coordinación de Recursos pueda validar en tiempo y forma esta solicitud. Para efectuar este proceso es necesario que todos los usuarios de la aplicación accedan a través de su propio *login* (figura 2).

Figura 2. Acceso a la aplicación a través del login



Fuente: Elaboración propia

Una vez efectuada esta validación, el usuario recibirá un correo electrónico, así como una notificación en la propia aplicación, donde se indicará el lugar y el horario de recogida de ese material. Con esta medida se pretende que los usuarios no interfieran en el desarrollo de las sesiones, ya que los materiales deportivos suelen encontrarse en espacios destinados para fines docentes. Para facilitar la organización y acceso de los distintos materiales deportivos, la aplicación cuenta con un inventario individualizado del material disponible en cada instalación deportiva, esto es: Pabellón Luis Vives, Aulario Santa María, Pabellón Guadalupe, Pista Multideportiva de Campus Champagnat.

Figura 3. Inventarios material instalaciones deportivas

Item	Quantity
Pelotas hockey	13
Pelotas Tennis-padel en bolsa	35
Colchonetas gruesas	4
Colchonetas finas	6
Raquetas padel artengo blancas PR7...	10
Raquetas padel Wilson STING (azules)	6
Fundas Raquetas de padel	8
Aros Medianos	22
Aros Medianos - algo más grandes	11

Fuente: Elaboración propia

Del mismo modo, la aplicación contará con la posibilidad de que el usuario pueda notificar a la Coordinación de Recursos e Infraestructuras que se ha efectuado la devolución en el espacio previsto. De esta manera existirá un control sobre el material que se entrega y se devuelve en los almacenes deportivos, impidiendo que este se extravíe o no esté localizado, evitando la adquisición excesiva de material.

Adicionalmente, la aplicación permitirá bloquear solicitudes de aquellos usuarios que no hayan seguido el protocolo de la Facultad. De esta manera, las devoluciones que no se hayan realizado de acuerdo a las indicaciones de la Coordinación de Recursos e Infraestructuras, traerán consigo una penalización al usuario que le impedirá realizar reservas de material deportivo, que variarán en función de la infracción realizada. Así, por cada día de retraso existirá una penalización de dos días sin la posibilidad de utilizar material deportivo.

Otra de las potencialidades que ofrece la aplicación, y a la que ya hemos aludido, es la posibilidad de que los usuarios notifiquen el deterioro de los materiales deportivos o incluso, la ausencia del mismo. Evidentemente, nos encontramos ante recursos

con una vida útil muy breve que deben ser repuestos cada poco tiempo. De esta manera facilitamos que el material disponible se encuentre en un perfecto estado para la práctica deportiva y, sobre todo, que pueda ser utilizado en el futuro por el resto de los compañeros.

En el diagrama de Gantt se ilustra esquemáticamente el cronograma estimado para cada una de las fases del proyecto, cuya duración aproximada ha sido de un año (tabla 1). En este sentido, para alcanzar el objetivo principal de la investigación, se ha dividido el proyecto en seis fases:

1. Revisión del estado del arte (literatura y tecnología).
2. Presentación y resolución de la convocatoria TALENT PRO24.25.
3. Contacto con profesionales del sector.
4. Desarrollo y prueba piloto por parte del equipo informático.
5. Evaluación de viabilidad.
6. Tratamiento e interpretación de resultados.
7. Transferencia de los resultados obtenidos

Tabla 1. Fases para el desarrollo del proyecto SISGER

FASES SISGER 2024-2025	O	N	D	E	F	M	A	M	J	
1. Revisión del estado de la cuestión y planteamiento inicial de la propuesta.	[Barra gris]									
2. Presentación y resolución del proyecto.		[Barra roja]								
3. Contacto con profesionales.				[Barra marrón]						
4. Desarrollo/prueba piloto					[Barra verde]					
5. Análisis de viabilidad						[Barra azul]				
6. Tratamiento e interpretación de resultados								[Barra verde]		
7. Transferencia de resultados									[Barra azul]	

Fuente: Elaboración propia

4. CONCLUSIONES Y REFLEXIONES FINALES DE LAS LIMITACIONES A LA TRANSFERENCIA DEL CONOCIMIENTO

Este proyecto se enfoca en la creación de una aplicación que permite reservar y utilizar recursos materiales deportivos de la Facultad de Educación de la Universidad Pontificia de Salamanca.

Gracias a ella, todos los profesores deberán seguir el mismo protocolo y, por lo tanto, tendrán las mismas oportunidades a la hora de reservar los diferentes recursos con los que cuenta la Facultad, lo que favorece una mayor y mejor rotación de material, ayudando a aumentar el número de horas de uso de cada uno de los recursos. De esta forma, también se evitaría la compra innecesaria de material (uno de los grandes problemas que presentaba la Facultad hasta el momento) ya que, mediante una adecuada planificación, se optimizarían las horas de utilización de cada elemento. Esta medida se encuentra alienada con el ODS 12 (producción y consumo responsable).

Aunque aún es pronto para valorar el impacto de la aplicación, lo cierto es que su irrupción ha sido positiva, ya que creemos que minimiza muchos de los problemas que la Facultad de Educación presentaba en la gestión de materiales y en la reserva de instalaciones, dos actividades que se desarrollan de manera ordinaria y que, en ocasiones, dificultaban el quehacer diario de los docentes.

Concebido como una solución innovadora, SISGER optimiza el uso de materiales deportivos, reduciendo la necesidad de nuevas adquisiciones mediante una gestión eficiente y sostenible (Pérez, Gude, Magraner et al., 2012). Además, su implementación puede expandirse a otras áreas de la universidad (Servicio de Deportes), e incluso fuera de ella. Y es que, nos encontramos ante una herramienta con un gran potencial, ya que su aplicabilidad podría extenderse a la gestión de recursos deportivos en otros centros e instituciones, tales como colegios, gimnasios, centros y clubes deportivos, etc. De esta manera, podremos incrementar su impacto y alcance, a la vez que se abren nuevas oportunidades de ingresos a través de su comercialización, consolidándolo como una solución eficaz para la gestión integral de recursos deportivos.

BIBLIOGRAFÍA

- Cabello, E y Cabra, N. Evaluación de las instalaciones deportivas escolares desde el punto de vista de la salud. *Revista Internacional de Medicina y Ciencias de la Actividad Física y el Deporte* 2006, 6 (23): 138-154 <http://cdeporte.rediris.es/revista/revista23/artinstalaciones.htm>
- Celma, J (Ed). *Dirección y gestión de instalaciones y centros deportivos*. Madrid, Federación de Asociaciones de Gestión del Deporte de España, 2020.

- Jiménez Sáiz, S. L., Lorenzo Calvo, A., Leite, N. M. C., & Gómez Ruano, M. Ángel. (2015). La relevancia de la gestión del equipo en el éxito deportivo. *KRONOS*, 14(01-02). <https://doi.org/10.64197/Kronos.14.01-02.748>
- Pérez, V, Gude, R, Magraner, L, Rosa, D, González, FM., Gil, I, Pereira, I, Montero, J. La sostenibilidad en instalaciones deportivas. *Revista biomecánica*, 2012. 58: 19-20.
- Tafalla, A. M. G., Sancho, J. A. M., & Gómez, A. M. (2005). *La importancia del gestor deportivo en el municipio* (Vol. 654). Inde.

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE APERTURA DE PUERTAS AUTOMÁTICO BASADO EN VISIÓN ARTIFICIAL Y BIOMETRÍA FACIAL Y OCULAR (VABRO)



ALFONSO JOSÉ LÓPEZ RIVERO

VIDAL ALONSO SECADES

JORGE ZAKOUR DIB

MANUEL MARTÍN-MERINO ACERA

ANTONIO FERRERAS GARCÍA

JORGE CHAMORRO SÁNCHEZ

RESUMEN

El proyecto VABRO (Visión Artificial, Biometría y Reconocimiento Ocular) desarrolla un sistema de control de acceso autónomo que integra reconocimiento facial, dactilar y ocular en un único dispositivo físico, con procesamiento embebido y funcionamiento sin conexión a internet. Su propósito es demostrar la viabilidad de un sistema de autenticación multimodal local, aplicando tecnologías de visión artificial, inteligencia artificial y electrónica de bajo consumo. El prototipo se concibió como una cerradura inteligente de triple validación, construida con microcontroladores ESP32-CAM y Arduino, junto con un módulo de procesamiento intermedio basado en ZimaBoard o Raspberry Pi. El sistema fue encapsulado en una carcasa diseñada y fabricada mediante impresión 3D, integrando además un módulo fotovoltaico y baterías recargables que permiten su funcionamiento autónomo. Desde el punto de vista académico, VABRO constituye un ejemplo práctico de transferencia tecnológica universitaria, aplicando conocimientos de electrónica, programación y seguridad informática en un producto tangible y escalable.

PALABRAS CLAVE: visión artificial, biometría, reconocimiento facial, seguridad, inteligencia artificial, IoT, transferencia universitaria.

1. CONTEXTUALIZACIÓN DEL PROYECTO

El control de acceso ha sido históricamente uno de los ámbitos más sensibles en la gestión de la seguridad física y digital. La evolución desde los sistemas mecánicos tradicionales hacia métodos electrónicos y, posteriormente, biométricos, refleja la búsqueda de mayor precisión, trazabilidad y comodidad en los procesos de autenticación personal (Jain et al., 2016). En este contexto, la biometría multimodal, que combina varios rasgos fisiológicos como el rostro, la huella o el iris, se ha consolidado como un campo de innovación aplicado tanto en entornos empresariales como domésticos (Ross & Jain, 2019).

Sin embargo, la mayoría de soluciones comerciales actuales dependen de infraestructuras en la nube, lo que introduce vulnerabilidades en términos de privacidad, latencia y dependencia de red. El proyecto VABRO surge precisamente como respuesta a esta limitación, con el propósito de desarrollar un sistema de verificación biométrica local, autónomo y reproducible, capaz de operar sin necesidad de conectividad externa. Este enfoque resulta especialmente útil en contextos donde la seguridad y la confidencialidad de los datos son prioritarias, como en instalaciones críticas o entornos rurales sin acceso estable a internet.

A nivel institucional, el proyecto se enmarca dentro de las líneas de innovación tecnológica y transferencia de conocimiento promovidas por la Universidad Pontificia de Salamanca (UPSA). Además, se alinea con la Estrategia RIS3 2021-2027 de Castilla y León, especialmente en los ámbitos de digitalización, seguridad e industria 4.0, contribuyendo a la formación práctica en tecnologías emergentes aplicadas a la seguridad inteligente.

2. JUSTIFICACIÓN Y OBJETIVOS

La motivación fundamental del proyecto radica en integrar, dentro de un mismo dispositivo, tres métodos biométricos complementarios, con un enfoque modular y de bajo coste. Frente a las soluciones centralizadas que dependen de servicios externos, VABRO propone una arquitectura distribuida que combina reconocimiento facial embebido, lectura de huellas dactilares y escaneo de iris, sin recurrir a almacenamiento en la nube.

El diseño del sistema se sustenta en tres principios esenciales. En primer lugar, la privacidad de los datos biométricos, garantizada al realizar todo el procesamiento de forma local sin transmitir información sensible por redes externas. En segundo lugar,

la autonomía energética, gracias al uso de baterías de tipo 18650 combinadas con un módulo solar fotovoltaico y una estación portátil Bluetti AC180. Finalmente, la viabilidad de transferencia, ya que la arquitectura y los materiales utilizados permiten replicar el sistema en entornos académicos, residenciales o industriales, favoreciendo su escalabilidad y adaptabilidad.

El objetivo principal del proyecto es desarrollar un prototipo funcional de cerradura inteligente multimodal, capaz de operar de forma completamente autónoma mediante identificación facial, dactilar y ocular. Para ello, se plantearon metas específicas que incluyen la implementación del reconocimiento facial embebido en una ESP32-CAM mediante visión artificial, la integración de sensores biométricos R307S e IriShield MK2120U en una arquitectura modular, el diseño de una carcasa personalizada mediante impresión 3D, el desarrollo de un sistema de alimentación híbrido y autónomo con respaldo fotovoltaico y, finalmente, la validación del sistema en escenarios reales, evaluando su rendimiento, precisión y consumo energético.

3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN

El diseño, desarrollo e implementación del sistema VABRO se abordó desde una perspectiva integral que combina ingeniería electrónica, visión artificial y arquitectura de sistemas distribuidos. En esta fase se definieron las relaciones funcionales entre los distintos módulos biométricos, el flujo de información y los mecanismos de control que garantizan la fiabilidad del proceso de autenticación. El enfoque se centró en lograr un equilibrio entre eficiencia, seguridad y autonomía operativa, de modo que cada componente pudiera desempeñar sus funciones de manera independiente pero coordinada dentro del conjunto. Asimismo, se priorizó la escalabilidad, de forma que el sistema admita futuras ampliaciones o sustituciones de sensores sin modificar su núcleo estructural. Esta etapa permitió transformar el planteamiento conceptual en un prototipo físico completamente operativo, sirviendo como base para las pruebas experimentales y la validación funcional.

La arquitectura general del sistema se estructura en tres niveles funcionales (ver Figura 1). El primer nivel corresponde a la adquisición biométrica, que incluye la captura de imágenes faciales mediante una ESP32-CAM, la lectura de huellas dactilares a través del sensor R307S y el escaneo del iris con el módulo IriShield MK2120U. El segundo nivel está dedicado al procesamiento y la toma de decisiones, donde se realiza el reconocimiento facial embebido, la validación local y la fusión de los datos biométricos mediante reglas lógicas de decisión basadas en operadores AND y OR, según la configuración deseada. Finalmente, el nivel de actuación controla la cerradura electromagnética mediante un relé de 5V y gestiona la comunicación de estado hacia una pantalla OLED o TFT, registrando además los eventos en memoria local a través de un disco USB-C.

El hardware del prototipo combina componentes de bajo coste y alto rendimiento. La ESP32-CAM actúa como nodo principal para el reconocimiento facial embebido con cámara OV2640, mientras que el sensor R307S gestiona la captura y verificación local de huellas. El sensor IriShield MK2120U se emplea para la autenticación ocular bajo un entorno Linux, ejecutándose en la ZimaBoard o la Raspberry Pi, que además funcionan como servidores intermedios de procesamiento. El sistema energético se compone de una batería 18650, un módulo de carga TP4056, un regulador de voltaje 7805 y la estación Bluetti AC180 con soporte fotovoltaico. Todo el conjunto se aloja en una carcasa fabricada mediante impresión 3D, diseñada con Autodesk Fusion 360 y elaborada en filamento PLA y PETG, como se muestra en la Figura 2.

El software del sistema se organiza en tres módulos principales. El módulo facial, basado en la ESP32-CAM, emplea un firmware derivado del conocido “ESP32-CAM WebServer”, modificado para incorporar una base de datos local y un umbral de similitud configurable. El módulo dactilar, programado en Arduino UNO o Nano, se encarga de la lectura, almacenamiento y validación de huellas, estableciendo comunicación mediante UART con el nodo central. Por su parte, el módulo ocular, ejecutado en la ZimaBoard o Raspberry Pi, utiliza el SDK del IriShield para la autenticación de iris mediante librerías en C/C++ bajo sistema Debian. Todos los módulos se coordinan mediante el programa “unidad_central.ino”, que sincroniza los resultados de los tres sensores a través de comunicación serie, activando la cerradura únicamente cuando se cumplen las condiciones de validación establecidas.

Durante la fase de integración y pruebas, se realizaron diversas etapas de validación, que incluyeron la comprobación individual de cada sensor biométrico, la calibración de los umbrales de similitud facial y ocular, y las pruebas de autonomía energética, en las cuales se demostró un funcionamiento continuo de 24 horas sin alimentación externa. Asimismo, se llevaron a cabo ensayos de precisión que registraron una tasa de aceptación falsa (FAR) inferior al 2% y una tasa de rechazo falso (FRR) inferior al 5%. El sistema fue finalmente presentado en la jornada universitaria de innovación tecnológica, donde se validó su funcionamiento en un entorno de demostración práctica.

El desarrollo de VABRO se concibió no solo como un ejercicio técnico de investigación, sino como una plataforma educativa y de transferencia de conocimiento. Todo el proceso de diseño, desarrollo y programación fue documentado con el objetivo de facilitar su uso en la docencia universitaria en áreas como electrónica, IoT y visión por computador. El código fuente completo fue registrado en la OTRI (Oficina de Transferencia de Resultados de Investigación) bajo licencia académica, con vistas a su futura evolución hacia un producto comercializable.

4. CONCLUSIONES Y REFLEXIONES FINALES SOBRE LA TRANSFERENCIA DEL CONOCIMIENTO

El proyecto VABRO demuestra la viabilidad de construir un sistema biométrico multimodal completamente local, utilizando componentes accesibles y tecnologías abiertas. Su principal aportación reside en la autonomía tanto energética como de procesamiento, lo que lo convierte en una herramienta idónea para contextos donde la conectividad es limitada o la privacidad de los datos es esencial.

Desde la perspectiva de transferencia de conocimiento, el proyecto resalta la importancia de la interdisciplinariedad, al combinar programación embebida, electrónica, inteligencia artificial y diseño industrial. Además, su viabilidad económica —con un coste total inferior a los 3.500 euros— y su carácter modular refuerzan su potencial de replicabilidad. No obstante, su transferencia directa al mercado presenta desafíos asociados a las certificaciones de seguridad biométrica (ISO/IEC 19795) y a las normativas de protección de datos personales (GDPR).

En el ámbito académico, VABRO ha servido como caso de estudio transversal entre las áreas de ingeniería, informática y diseño industrial, fortaleciendo la relación entre la universidad y la sociedad mediante la materialización de resultados de investigación aplicada.

BIBLIOGRAFÍA

- Jain, A. K., Ross, A., & Nandakumar, K. (2016). *Introduction to Biometrics*. Springer.
- Ross, A., & Jain, A. (2019). *Multimodal Biometrics: An Overview*. In *Handbook of Biometrics*. Springer.
- Daugman, J. (2015). How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1), 21–30.
- González, R. C., & Woods, R. E. (2018). *Digital Image Processing*. Pearson.
- Salvi, J., Fernandez, S., Pribanic, T., & Llado, X. (2019). State of the art on structured light patterns for 3D scanning. *Pattern Recognition*, 43(8), 2666–2680.
- European Commission (2021). *Artificial Intelligence Act: Ensuring Trustworthy AI*. Brussels.
- OpenAI (2024). *Edge AI and On-Device Inference in Security Systems*. OpenAI Research Reports.

ANEXOS

Figura 1. Arquitectura general del sistema VABRO.

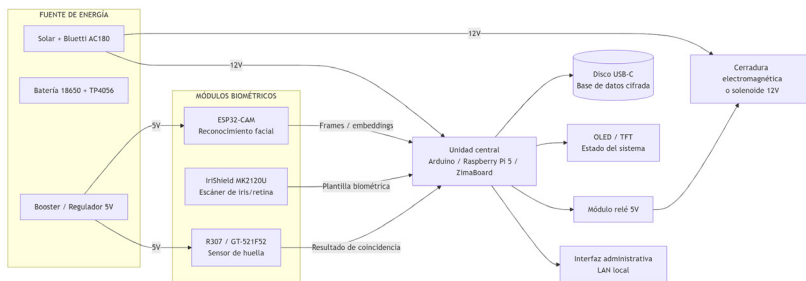


Figura 2. Prototipo físico ensamblado con sensores y carcasa impresa en 3D.



Código fuente registrado:

- *esp32-cam-webserver.ino*
- *fingerprint_sensor_r307s.ino*
- *unidad_central.ino*

PLATAFORMA INTELIGENTE DE EVALUACIÓN DINÁMICA Y ADAPTATIVA (IDEA)



RUBÉN MARTÍN GARCÍA

SERGIO ALONSO ROLLÁN

SAMUEL MARCOS PABLOS

REBECA CORDERO GUTIÉRREZ

LUCÍA MARTÍN GÓMEZ

RESUMEN

El proyecto IDEA surge como respuesta a la necesidad de contar con herramientas de evaluación accesibles, dinámicas y personalizadas en un contexto educativo cada vez más digitalizado. La propuesta combina inteligencia artificial, análisis en tiempo real, gamificación y compatibilidad con plataformas virtuales como Moodle. IDEA se configura como una solución gratuita y de código abierto que democratiza el acceso a evaluaciones de calidad, facilitando la labor de los docentes y mejorando la experiencia de aprendizaje para los estudiantes. Su diseño centrado en la accesibilidad y la adaptabilidad permite atender a diversas necesidades educativas, superando las limitaciones de otras plataformas existentes.

1. CONTEXTUALIZACIÓN DEL PROYECTO

En la última década, las tecnologías educativas han cobrado un papel central en la enseñanza y la evaluación (Redecker & Punie, 2017). La tecnología se ha convertido en una herramienta clave para solucionar diferentes problemas pedagógicos (Williamson, Eynon & Potter, 2020). Sin embargo, muchas de las soluciones actuales

presentan limitaciones significativas: requieren licencias de pago, imponen restricciones en el número de participantes o carecen de funcionalidades avanzadas de personalización. Estos factores suponen barreras importantes, especialmente para instituciones con recursos limitados o docentes que gestionan aulas con un elevado número de estudiantes.

Plataformas conocidas como Kahoot (Kahoot!, 2013) han demostrado la utilidad de las evaluaciones interactivas en tiempo real, pero sus versiones gratuitas restringen el número de usuarios, lo que las hace poco viables en aulas con un elevado número de alumnos. Además, la mayoría de estas herramientas carecen de un enfoque adaptativo que atienda a los diferentes niveles de aprendizaje de cada estudiante.

En este contexto se desarrolla IDEA, una plataforma cuyo valor diferencial radica en su gratuidad, escalabilidad, flexibilidad y capacidad de integrar en un único entorno las funcionalidades más demandadas por docentes y estudiantes. IDEA responde a la necesidad de contar con un sistema inclusivo, que se adapte tanto a la educación presencial como a la virtual, y que incorpore herramientas innovadoras basadas en inteligencia artificial para optimizar los procesos de evaluación (Luckin et al., 2016). El proyecto automatiza el proceso de evaluación digital (desde la creación de exámenes hasta la calificación de los mismos) y detecta los contenidos que plantean una mayor dificultad al alumnado, favoreciendo que el docente pueda emplear su tiempo en optimizar la calidad de la enseñanza.

2. JUSTIFICACIÓN Y OBJETIVOS

El proyecto IDEA se justifica por la necesidad de democratizar el acceso a evaluaciones digitales avanzadas. Mientras que otras plataformas suponen un coste elevado o una experiencia limitada en sus versiones gratuitas, IDEA garantiza una solución abierta y accesible para cualquier institución educativa.

El **objetivo principal** es desarrollar una plataforma de evaluación dinámica y adaptativa que permita a los docentes crear, personalizar y administrar pruebas en vivo sin limitaciones de participantes, ofreciendo a los estudiantes una experiencia accesible, interactiva y ajustada a diferentes contextos de enseñanza.

De este objetivo general se derivan los siguientes **objetivos específicos**:

- Implementar un sistema de auto-generación de exámenes mediante inteligencia artificial.
- Desarrollar herramientas de análisis estadístico y gamificación que permitan retroalimentación tanto al docente como al discente en tiempo real.
- Crear un banco de preguntas configurable que facilite la exportación e importación de contenidos.
- Garantizar la compatibilidad de la herramienta con plataformas como Moodle.

- Diseñar con principios de accesibilidad universal para incluir a estudiantes con diferentes necesidades educativas.

3. DISEÑO, DESARROLLO E IMPLEMENTACIÓN

3.1. Actividades técnicas

El desarrollo técnico de IDEA comenzó con la definición de una arquitectura de sistema robusta. Se implementó un servidor físico de alto rendimiento, equipado con procesador Intel Xeon, amplia memoria RAM y tarjeta gráfica de última generación compatible con CUDA. Esta infraestructura permite ejecutar procesos intensivos de inteligencia artificial y gestionar grandes volúmenes de datos de manera eficiente.

Posteriormente, se abordó la programación del *backend* y *frontend* de la aplicación. El *backend* se diseñó para generar preguntas a partir del contenido de la asignatura, gestionar usuarios, almacenar datos de evaluación y garantizar la seguridad de la información. El *frontend*, por su parte, se enfocó en ofrecer una experiencia accesible intuitiva para el usuario, con menús claros y navegación sencilla.

La integración de modelos de inteligencia artificial fue uno de los hitos clave. Se aplicaron algoritmos de procesamiento de lenguaje natural (PLN) capaces de generar bancos de preguntas a partir de contenidos curriculares, clasificar por niveles de dificultad y proponer variaciones automáticas. De esta forma, se facilita la creación de múltiples modelos de exámenes, reduciendo el tiempo invertido por el profesorado. Además, la plataforma IDEA reduce el problema actual de plagio entre el alumnado, lo que supone un mal creciente desde que la evaluación se realiza con medios tecnológicos.

Por último, se desplegó un sistema de análisis en tiempo real que permite a los docentes monitorizar el rendimiento del alumnado durante la realización de los exámenes. Las respuestas de los alumnos se analizan para descubrir sus fortalezas y debilidades y para conocer si hay alguna parte del contenido de la asignatura que requiera un repaso. Por tanto, esta funcionalidad no solo mejora el seguimiento académico, sino que también ofrece datos valiosos para la toma de decisiones pedagógicas.

3.2. Actividades de diseño

El diseño de la interfaz de usuario se centró en la accesibilidad y la sencillez. Desde un inicio se trabajó en prototipos visuales que fueron testeados con docentes y estudiantes. Estos prototipos sirvieron para identificar mejoras en la navegación y garantizar que la curva de aprendizaje para el uso de la plataforma fuese mínima.

Se implementaron criterios de accesibilidad universal, como compatibilidad con lectores de pantalla, navegación por teclado y uso de colores contrastados. Estas me-

didadas buscan asegurar que IDEA pueda ser utilizada por estudiantes con diferentes capacidades y necesidades.

3.3. Actividades funcionales y pedagógicas

La plataforma IDEA incorpora funcionalidades que responden a las demandas pedagógicas actuales. El banco de preguntas permite organizar ítems por temas, niveles de dificultad y tipo de evaluación, ofreciendo flexibilidad en la preparación de pruebas tanto para el profesorado como para el alumnado.

La gamificación fue otra de las innovaciones incluidas en la herramienta. Se introdujeron mecánicas que transforman la evaluación en una experiencia más motivadora y participativa, fomentando la implicación del alumnado.

Asimismo, se configuró un sistema de retroalimentación adaptativa, capaz de ofrecer comentarios inmediatos y personalizados según el rendimiento de cada estudiante. Esta funcionalidad refuerza el aprendizaje y permite al docente identificar áreas de mejora de forma ágil. Del mismo modo, IDEA se convierte en una herramienta clave para el estudio y/o el repaso de la materia por parte de los discentes.

Finalmente, la compatibilidad con Moodle asegura que la plataforma pueda integrarse fácilmente en entornos educativos ya consolidados como un módulo más, facilitando su adopción sin necesidad de grandes cambios en la infraestructura existente.

3.4. Resultados obtenidos

El desarrollo del proyecto IDEA ha dado lugar a una plataforma funcional y accesible, disponible para docentes y estudiantes sin limitaciones en el número de participantes. La herramienta permite la creación y administración de exámenes en tiempo real, con un rendimiento estable y escalable incluso en contextos con grandes volúmenes de usuarios.

La implementación de inteligencia artificial ha demostrado su eficacia en la generación automática de bancos de preguntas, adaptados a las necesidades de cada asignatura y nivel de dificultad. Esta funcionalidad ha reducido significativamente el tiempo invertido en la preparación de pruebas y ha incrementado la capacidad de personalización.

Se han incorporado herramientas de análisis en vivo y gamificación, que ofrecen a los docentes retroalimentación inmediata y detallada sobre el desempeño de los estudiantes, al mismo tiempo que aumentan la motivación y el compromiso del alumnado durante las pruebas.

La plataforma dispone de un banco de preguntas configurable y exportable, organizado por temáticas y niveles, que facilita la elaboración de múltiples modelos de

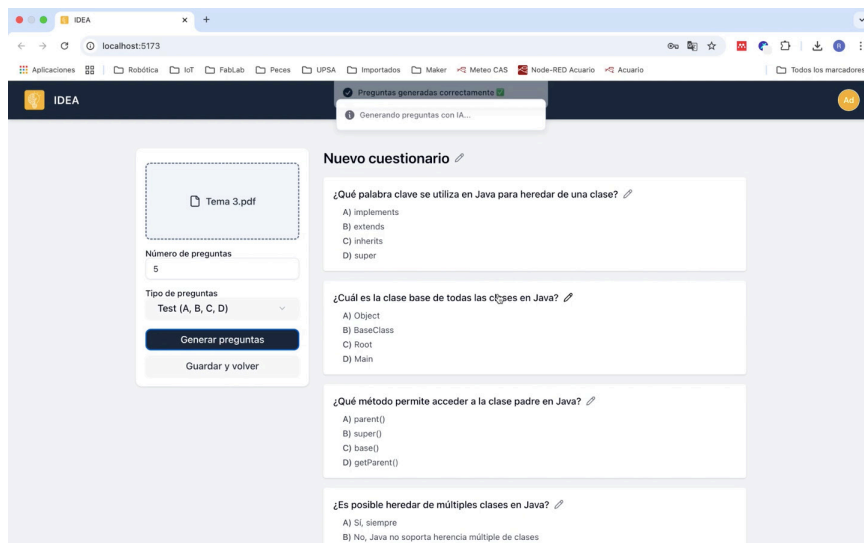
exámenes. Este recurso permite a los docentes diversificar las pruebas, evitar repeticiones y mejorar la fiabilidad de las evaluaciones.

IDEA ha alcanzado una integración efectiva con Moodle, lo que facilita su adopción en entornos virtuales de aprendizaje ya existentes, garantizando compatibilidad y escalabilidad en instituciones educativas de distinto tamaño y recursos.

En términos de diseño, la interfaz es clara, intuitiva y adaptada a criterios de accesibilidad universal, lo que asegura que tanto docentes como estudiantes puedan utilizar la herramienta sin necesidad de formación especializada y con independencia de sus condiciones personales.

En conjunto, estos resultados confirman que IDEA no solo cumple con los objetivos planteados inicialmente, sino que también aporta un valor añadido al ecosistema de tecnologías educativas, consolidándose como una alternativa gratuita, flexible e inclusiva frente a las soluciones comerciales actuales.

Figura 1. Ejemplo de generación de un test básico a partir de un tema en formato PDF



4. CONCLUSIONES Y REFLEXIONES FINALES

El proyecto IDEA ha alcanzado los objetivos planteados, consolidándose como una plataforma innovadora que combina personalización, accesibilidad y gratuidad. La integración de inteligencia artificial y gamificación, junto con su compatibilidad con Moodle, convierten a IDEA en una alternativa real a las soluciones de pago actuales.

Entre las reflexiones finales destacan los retos de transferencia del conocimiento. La adaptación de la plataforma a diferentes sistemas educativos internacionales requerirá ajustes normativos y técnicos. Asimismo, la sostenibilidad de un modelo gratuito y abierto plantea la necesidad de establecer colaboraciones con instituciones que garanticen su mantenimiento y evolución futura.

Otro desafío será la mejora continua de los algoritmos de IA, que deben evolucionar para atender nuevas metodologías pedagógicas y para ofrecer una personalización cada vez más precisa. No obstante, los avances logrados muestran el potencial de IDEA como referente en el ámbito de la evaluación educativa.

BIBLIOGRAFÍA

- Kahoot!. Kahoot! learning platform. Oslo, Kahoot! AS, 2013; 1:1-1. Disponible en: <https://kahoot.com>
- Luckin R, Holmes W, Griffiths M, Forcier LB. Intelligence unleashed: An argument for AI in education. London, Pearson Education, 2016; 1:1-120.
- Redecker C, Punie Y. European framework for the digital competence of educators: DigCompEdu. Luxembourg, Publications Office of the European Union, 2017; 1:1-100.
- Williamson B, Eynon R, Potter J. Pandemic politics, pedagogies and practices: Digital technologies and distance education during the coronavirus emergency. *Learn Media Technol* 2020; 45:107-114.



**MANTÉNGASE INFORMADO
DE LAS NUEVAS PUBLICACIONES**

**Suscríbese gratis
al boletín informativo
www.dykinson.com**

Y benefíciense de nuestras ofertas semanales