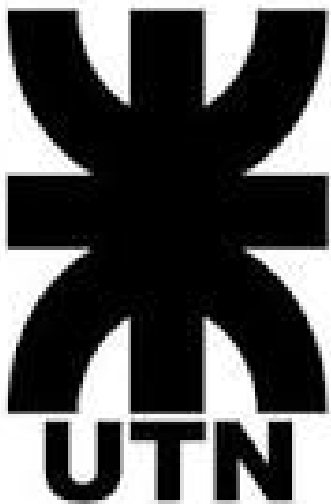


Universidad Tecnológica Nacional

Facultad Regional Córdoba



Ingeniería en Sistemas de Información

Proyecto final

Docente:

- Adjunto **Quinteros, Sergio**
- JTP **Destefanis María Laura**

Grupo: 6

Comisión: 5k3

Año: 2026

Tema:

Integrantes:

- Bonetto, Patricio; 62573
- Correa, Jeremías; 88714
- Gallardo, Valentin; 92781
- Glavas, Delfina; 89335
- Maero, Augusto; 91104

PHARO

Adaptación Accesible de Archivos y Entornos



QR CÁTEDRA



proyectopharo@gmail.com



NECESIDAD

Las personas con discapacidad visual enfrentan barreras para acceder a información digital no accesible.



OBJETIVO

Brindar acceso autónomo, eficiente y seguro a contenidos digitales a través de una plataforma multimodal con IA.



SOLUCIÓN

PHARO convierte, organiza y entrega información en formatos accesibles mediante tecnologías inteligentes y multimodales.

FUNCIONALIDADES CLAVE

OCR

OCR

Extracción de texto y fórmulas complejas



Pharolito

Asistente de IA para ayudar a entender tus documentos y acompañarte



Audio

Síntesis de voz natural



Descripción

del entorno mediante IA



Perfil Accesible

Preferencias y experiencia personalizada

1. FUENTE



Documento PDF e imagen

2. OCR



Reconocimiento óptico de caracteres

3. WORKER AI EN REDIS QUEUE



Worker AI en redis Queue

4. BACKEND GO / POSTGRESQL



API GO Base de datos PostgreSQL

5. API / EXTENSIÓN REACT NATIVE



App móvil, Extensión navegador

6. USUARIO

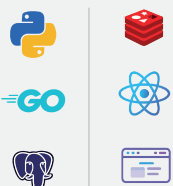


Usuario lector de pantalla / Audio

Worker AI encolado por Redis

Backend + Frontend separados

TECNOLOGÍAS



RESULTADOS

- Conversión precisa de documentos a texto.
- Entrega de información en audio clara y continua.
- Acceso seguro y controlado por el usuario.
- Experiencia accesible en web, extensión y móvil.
- Procesamiento escalable y eficiente.



BENEFICIOS

- Mayor autonomía e inclusión digital.
- Acceso equitativo a la información.
- Herramientas seguras y confiables.
- Ahorro de tiempo y esfuerzo.
- Tecnología abierta y escalable.



UNIVERSIDAD TECNOLÓGICA NACIONAL | Facultad Regional Córdoba
Ingeniería en Sistemas de Información | Proyecto Final 2026

Docentes: Ing. Sergio Quinteros | Ing. Destefanis María Laura

• Bonetto Patricio • Correa Jeremías • Gallardo Valentín • Glavas Delfina • Maero Augusto

Pharo: Plataforma Multimodal para el Acceso Inteligente a Información Digital de Personas con Discapacidad Visual.

Bonetto Patricio, Correa Jeremias, Gallardo Valentin,
Glavas Delfina, Maero Augusto.

Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract

*Con el fin de incrementar la autonomía de personas con discapacidad visual en contextos educativos y organizacionales se ha desarrollado el proyecto **Pharo**, el cual tuvo como objetivo desarrollar una plataforma digital multimodal, abordando la fragmentación de las herramientas de accesibilidad, la falta de persistencia en las configuraciones de usuario y las dificultades para acceder a documentos digitales no estructurados. Para ello, se diseñó e implementó una solución que integra procesamiento inteligente de documentos mediante reconocimiento óptico de caracteres (OCR) y modelos de lenguaje, perfiles de accesibilidad persistentes y un modelo de datos multi-organización con aislamiento seguro de la información. La plataforma incorpora una aplicación multiplataforma y una extensión de navegador que mantienen una experiencia de accesibilidad consistente en distintos dispositivos y entornos. Como resultado, el sistema permite transformar documentos PDF escaneados e imágenes en contenido accesible mediante texto estructurado y audio sintético, conservar las preferencias de accesibilidad entre sesiones y garantizar el aislamiento de los datos institucionales. Se concluye que la integración de estas capacidades en una única plataforma constituye una alternativa viable para reducir la carga cognitiva derivada del uso de múltiples herramientas y favorecer una experiencia digital más accesible, consistente e inclusiva.*

Palabras Clave

Accesibilidad Digital, Discapacidad Visual, Plataforma Multimodal, Impacto Social, Modelos de Lenguaje, Procesamiento Inteligente de Documentos.

Introducción

Las personas con discapacidad visual parcial o total enfrentan constantes barreras de accesibilidad en los entornos digitales, dificultando su interacción con información no estructurada, como documentos escaneados e interfaces sin etiquetado semántico. Si bien existen soluciones como lectores de pantalla, herramientas de reconocimiento óptico de caracteres (OCR) y asistentes basados en inteligencia artificial, estas suelen ofrecer funcionalidades aisladas, carecen de persistencia en las configuraciones de accesibilidad entre dispositivos y generan una elevada carga cognitiva al requerir el uso de múltiples aplicaciones. Esta situación limita la autonomía de los

usuarios y dificulta el acceso continuo a la información digital.

El objetivo de este trabajo es desarrollar Pharo, una plataforma multimodal y configurable que integra procesamiento inteligente de documentos, adaptación dinámica de interfaces y asistencia contextual mediante inteligencia artificial. Para ello, se propone una arquitectura modular y escalable, con separación entre backend y frontend, y con procesamiento asíncrono mediante un worker Python coordinado por Redis. Esta arquitectura incorpora reconocimiento óptico de caracteres (OCR) y modelos de lenguaje de gran escala (LLMs) para transformar contenido digital no estructurado en formatos accesibles, preservando además perfiles de accesibilidad persistentes y garantizando el aislamiento seguro de la información entre organizaciones.

Elementos del Trabajo y metodología

El desarrollo de Pharo se llevó a cabo mediante el marco de trabajo ágil Scrum [8], organizado en iteraciones de tres semanas que permitieron planificar, implementar, revisar y validar incrementos funcionales de manera continua. En cuanto a la gestión del proyecto se utilizó la guía PMBOK [9] del Project Management Institute. La gestión del código fuente se realizó mediante la estrategia GitFlow [10] sobre repositorios independientes para el backend y el frontend, complementada con revisiones automáticas de calidad, seguridad y cobertura de pruebas, además de revisiones manuales mediante *Pull Requests*, con el objetivo de asegurar la mantenibilidad y confiabilidad del software.

La solución se implementó siguiendo una arquitectura modular con separación entre backend y frontend, junto con un modelo multi-organización (multi-tenant) basado en una base de datos PostgreSQL [4] compartida. Para garantizar el aislamiento seguro de la información entre organizaciones

se aplicaron políticas de Row-Level Security (RLS), mientras que la extensión pgvector permitió el almacenamiento de representaciones semánticas y Redis se empleó como mecanismo de encolado para el procesamiento asíncrono de tareas y el almacenamiento temporal de datos.

El backend se desarrolló en Go utilizando el framework Gin [7] y el ORM GORM para la implementación de los servicios transaccionales. El procesamiento inteligente se implementó como un worker Python empaquetado en una imagen Docker y ejecutado a partir de tareas encoladas mediante Redis [5]. En dicho worker se integraron técnicas de reconocimiento óptico de caracteres (OCR) mediante Google Document AI [2] y Mathpix [11] para la extracción de contenido textual y expresiones matemáticas. Asimismo, se emplearon LangChain [3][13] para la orquestación de flujos basados en recuperación aumentada por generación (RAG), LiteLLM [13] como interfaz unificada para el acceso a modelos de lenguaje de gran escala (LLMs) y agentes de inteligencia artificial para coordinar flujos de trabajo, automatizar procesos complejos y adaptar dinámicamente la ejecución de tareas según el contexto y las necesidades del usuario.

La interfaz de usuario se implementó con React Native sobre Expo [6][14], permitiendo generar aplicaciones móviles y web a partir de un único código fuente. La gestión del estado y de las consultas remotas se realizó mediante Zustand y TanStack Query. Finalmente, el aseguramiento de la calidad se efectuó mediante pruebas unitarias, de integración y de extremo a extremo utilizando Pytest, Jest, Cypress y Playwright, incorporando además validaciones automáticas de accesibilidad con axe-core [15] para verificar el cumplimiento de las pautas WCAG 2.2 [1].

Resultados

El desarrollo culminó con un Producto Mínimo Viable (MVP) funcional que permitió la creación y administración de cuentas de usuario, la gestión de organizaciones y permisos, y la persistencia de perfiles de accesibilidad compartidos entre los distintos componentes de la plataforma. El procesamiento inteligente de documentos transformó archivos PDF escaneados e imágenes en texto estructurado mediante OCR y modelos de lenguaje de gran escala,

incorporando además la generación de audio sintético. La cobertura de pruebas de la lógica de negocio del backend superó el 80%. Asimismo, el sistema distribuyó el procesamiento asíncrono mediante Redis, utilizando workers para las tareas de OCR, generación de embeddings y consultas contextualizadas. Las evaluaciones de accesibilidad realizadas con Lighthouse CI, NVDA, VoiceOver y TalkBack verificaron la correcta navegación mediante teclado y el uso adecuado de etiquetas ARIA. Finalmente, la extensión de navegador sincronizó las preferencias de accesibilidad definidas en la plataforma con sitios web externos.

Discusión

Los resultados obtenidos indican que la integración de múltiples funcionalidades de accesibilidad en una única plataforma constituye una alternativa viable frente a las soluciones tradicionales, que suelen ofrecer herramientas aisladas y configuraciones independientes. La persistencia de los perfiles de accesibilidad favorece una experiencia de uso consistente entre dispositivos y aplicaciones, reduciendo la necesidad de reconfigurar preferencias y, en consecuencia, la carga cognitiva de los usuarios.

La incorporación de modelos de lenguaje de gran escala (LLMs) combinados con estrategias de recuperación aumentada por generación (RAG) permite mejorar la pertinencia de las respuestas al fundamentarlas en documentos previamente procesados. Asimismo, la separación entre backend, frontend y procesamiento asíncrono facilita la escalabilidad, el mantenimiento y la incorporación de nuevas funcionalidades sin afectar el funcionamiento de los demás componentes, mientras que el modelo multi-organización con políticas de Row-Level Security posibilita la utilización de la plataforma en distintos entornos institucionales preservando el aislamiento de la información.

No obstante, el sistema presenta limitaciones asociadas al procesamiento de documentos de gran tamaño o elevada complejidad gráfica, los cuales incrementan los tiempos de respuesta y el consumo de recursos computacionales. Para mitigar estos efectos se implementan restricciones sobre el tamaño de los archivos, técnicas de

optimización del procesamiento y mecanismos de selección dinámica de modelos de inteligencia artificial según la complejidad de la tarea. Estas características permiten considerar a Pharo como una base tecnológica extensible para futuras aplicaciones de accesibilidad digital apoyadas en inteligencia artificial.

Conclusión

El desarrollo de Pharo cumple el objetivo de proporcionar una plataforma multimodal orientada a mejorar la accesibilidad digital de personas con discapacidad visual mediante la integración de procesamiento inteligente de documentos, adaptación de interfaces y perfiles de accesibilidad persistentes. Los resultados obtenidos demuestran la viabilidad de unificar estas capacidades en un único ecosistema, favoreciendo una experiencia de usuario más consistente y reduciendo la dependencia de herramientas independientes.

La arquitectura implementada proporciona una base escalable y segura para su adopción en entornos educativos y organizacionales, permitiendo incorporar nuevas funcionalidades sin comprometer la integridad de la información ni la experiencia de uso. En este sentido, el trabajo constituye un aporte al desarrollo de soluciones de accesibilidad apoyadas en inteligencia artificial y orientadas a la inclusión digital.

Como líneas de trabajo futuro, se plantea incorporar capacidades de visión computacional para la interpretación del entorno físico en tiempo real y ampliar las funcionalidades colaborativas de la plataforma mediante herramientas de comunicación y apoyo entre usuarios, con el propósito de fortalecer la autonomía, la interacción y la escalabilidad del sistema.

Agradecimientos

Agradecemos al cuerpo docente de la cátedra Proyecto Final de la UTN - Facultad Regional Córdoba, en especial a nuestra tutora, Destefanis María Laura, por su acompañamiento, guía y aportes a lo largo del proyecto. Además, se extiende un agradecimiento formal a la organización Anthropic y a OpenAI que fueron nuestro apoyo en soporte técnico, auditoría y optimización tanto del ciclo de construcción del software como de la elaboración documental.

Referencias

[1] World Wide Web Consortium (W3C) – Web Content Accessibility Guidelines (WCAG) 2.2 – W3C Recommendation – 2023. Disponible en: <https://www.w3.org/TR/WCAG22/>.

[2] Google Cloud – Document AI Documentation – Google Cloud – 2026. Disponible en: <https://cloud.google.com/document-ai/docs>.

[3] Lewis, P.; Perez, E – Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks – Advances in Neural Information Processing Systems – 2020.

[4] PostgreSQL Global Development Group – PostgreSQL Documentation: Row Security Policies – PostgreSQL – 2026. Disponible en: <https://www.postgresql.org/docs/current/ddl-rowsecurity.html>.

[5] Redis Ltd. – Redis Documentation – Redis – 2026. Disponible en: <https://redis.io/docs/latest/>.

[6] React Native – React Native Documentation – Meta Open Source – 2026. Disponible en: <https://reactnative.dev/docs/getting-started>.

[7] Gin Web Framework – Gin Documentation – Gin Web Framework – 2026. Disponible en: <https://gin-gonic.com/docs/>.

[8] Schwaber, K. & Sutherland, J. – The Scrum Guide – 2020 <https://www.scrum.org/scrum-guide-2020>.

[9] Project Management Institute – Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK).

<https://www.pmi.org/pmbok-guide-standards>.

[10] Driessen, V. – A successful Git branching model (GitFlow).

<https://nvie.com/posts/a-successful-git-branching-model/>.

[11] Mathpix Inc. – Mathpix OCR API Documentation. <https://mathpix.com/docs/ocr>.

[12] Chase, H. – LangChain Documentation. <https://python.langchain.com/>.

[13] BerriAI – LiteLLM Documentation. <https://docs.litellm.ai/>.

[14] 650 Industries – Expo Documentation. <https://docs.expo.dev/>.

[15] Deque Systems – axe-core: The accessibility engine. <https://github.com/dequelabs/axe-core>.

Datos de Contacto:

Bonetto, Patricio. Universidad Tecnológica Nacional, Facultad Regional Córdoba.

patricio.bonetto@outlook.com

Correa, Jeremias Valentin. Universidad Tecnológica Nacional, Facultad Regional Córdoba. correaajere10@gmail.com.

Gallardo Scaltriti, Valentin Alberto. Universidad Tecnológica Nacional, Facultad Regional Córdoba. gallardovalen21@gmail.com

Glavas, Delfina Brenda. Universidad Tecnológica Nacional, Facultad Regional Córdoba. delfiglavas@gmail.com

Maero, Augusto. Universidad Tecnológica Nacional, Facultad Regional Córdoba. auguma02@gmail.com

AÑO	2026	CURSO Y NRO. DE GRUPO	5K3 - G6
NOMBRE DEL SISTEMA / PROYECTO			
PHARO			
CATEGORÍA (Solución De Negocio / Producto / Proyecto De Impacto Social)			
Proyecto De Impacto Social			
HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS			
ÁMBITO DE APLICACIÓN		NOMBRE Y VERSIÓN	
ENTORNO DE DESARROLLO		Visual Studio Code (v1.130.0)	
REPOSITORIOS Y VERSIONADO		GitHub, Git, GitFlow.	
PROGRAMACIÓN		Go (v1.22), Python (v3.12), React Native (v0.81.5), LiteLLM.	
BASE DE DATOS		PostgreSQL, Redis (v9.6.2), pgvector (extensión).	
COMUNICACIÓN INTERNA		Discord, WhatsApp.	
CAPACITACIÓN		Ninguna.	
PRUEBAS DE SISTEMA		Pytest, Jest (v29.7.0), Cypress, Playwright, NVDA, VoiceOver y TalkBack.	
GESTIÓN DEL PROYECTO		Scrum, Jira.	
DOCUMENTACIÓN		Google Drive y Confluence.	
MODELOS		LLMs y Visión: OpenAI (gpt-4.1-mini), Kimi / Moonshot AI (kimi-k3), Ollama (qwen2.5:7b para texto, llava:7b para visión), Anthropic Claude, LiteLLM (Gateway), LangChain (Orquestación RAG). OCR y Procesamiento: Google Document AI, Mathpix, MediaPipe. Síntesis de Voz: Google TTS / ElevenLabs.	