



UNIVERSIDAD
CATÓLICA
DE CUENCA

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA DE INFORMÁTICA,
CIENCIAS DE LA COMPUTACIÓN E
INNOVACIÓN TECNOLÓGICA**

CARRERA DE SOFTWARE

**Desarrollo de Frontend para el Sistema de Gestión de
Pacientes en UDIPSAI**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO DE SOFTWARE**

AUTOR: CARLOS ALBERTO VALLADAREZ BRIONES

DIRECTOR: ING. SEGUNDO ISRAEL SAÑAY SAÑAY, MGST

CUENCA - ECUADOR

2025

DIOS, PATRIA, CULTURA Y DESARROLLO



UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA DE INFORMÁTICA,
CIENCIAS DE LA COMPUTACIÓN E
INNOVACIÓN TECNOLÓGICA**

CARRERA DE SOFTWARE

**Desarrollo de Frontend para el Sistema de Gestión de
Pacientes en UDIPSAI**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO DE SOFTWARE**

AUTOR: CARLOS ALBERTO VALLADAREZ BRIONES

DIRECTOR: ING. SEGUNDO ISAAEL SAÑAY SAÑAY, MGST

CUENCA - ECUADOR

2025

DIOS, PATRIA, CULTURA Y DESARROLLO

Declaratoria de Autoría y Responsabilidad

Carlos Alberto Valladarez Briones portador de la cédula de ciudadanía N° **0105599385**. Declaro ser el autor de la obra: “**Desarrollo de Frontend para el Sistema de Gestión de Pacientes en UDIPSAI**”, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Cuenca, **20 de Febrero de 2025**

F:

Carlos Alberto Valladarez Briones

C.I. 0105599385

CERTIFICADO

Certifico que el presente Trabajo de investigación fue desarrollado por **Carlos Alberto Valladarez Briones** con el tema “**Desarrollo de Frontend para el Sistema de Gestión de Pacientes en UDIPSAI**”, bajo mi supervisión.

Atentamente,



Ing. Segundo Isael Sañay Sañay

Dedicatoria

“Vida antes que muerte, fuerza antes que debilidad, viaje antes que destino”

-Brandon Sanderson

A mi padre con toda mi gratitud, esto es para ti.

A todos mis amigos y familiares quienes me han brindado fuerza y apoyo a lo largo de los años.

Y de manera especial a las tres personas más importantes de mi vida. Mi madre, Martha, quien me dio la vida; mi abuelita Julia, quien me cuidó y apoyo siempre desde niño enseñándome a vivirla; y al amor de mi vida, Michelle, quien me ha dado motivos para vivirla, quien me ha dado fuerza cuando he sido débil, y con quien he recorrido este viaje hasta este destino.

Agradecimientos

Se extiende el más sincero agradecimiento al personal de UDIPSAI por su valiosa colaboración en la toma de requisitos para este proyecto.

De igual forma quiero expresar mi profundo agradecimiento al Ing. Isael Sañay, Mgst. por su arduo trabajo y dedicación como mentor y director del presente trabajo de titulación.

También quiero reconocer a mi compañero David Mora por su compromiso y dedicación al asumir la responsabilidad del desarrollo del backend del sistema.

Finalmente, extiendo mi profundo agradecimiento a la Lic. Michelle Tinizaray por su apoyo incondicional, así como por su meticulosa revisión y corrección del borrador de este artículo.

Resumen

La Unidad de Diagnóstico, Investigación Psicopedagógica y Apoyo a la Inclusión (UDIPSAI) se enfrenta a problemas en la gestión de pacientes debido al manejo analógico de información, lo que dificulta el acceso a los datos incrementando su vulnerabilidad. Para solventar esta situación, se ha desarrollado un producto de software de gestión de pacientes que permite el registro, almacenamiento y consultas de información de manera eficiente; mejorando la calidad del servicio y evitando la revictimización. El proyecto incluyó la recopilación de requisitos funcionales y no funcionales con el estándar de calidad FURPS+, así como el desarrollo del frontend para la interacción de los usuarios y la capacitación de ellos en base a material didáctico elaborado. El desarrollo se orientó con el marco de trabajo ágil SCRUM permitiendo un proceso iterativo e incremental que abarcó la recopilación de requisitos, planificación y el desarrollo en sprints, hasta el despliegue, la capacitación a los stakeholders y la refactorización del producto. Con relación a la mejora del proceso analógico, se optimizó el acceso a la información de los pacientes, reforzó la seguridad de los datos de la UDIPSAI, consiguiendo una percepción de mejora de la eficiencia de la gestión de pacientes y reducción de duplicidad de los datos. En concreto, el 100 % de los usuarios calificaron la aplicación entre 4 y 5 en la escala de Likert, y un 66,6 % afirmó que había mejorado el tiempo necesario para cumplimentar fichas y registros.

Palabras clave: *Unidad de Diagnóstico, Investigación Psicopedagógica, Apoyo a la Inclusión, Revictimización, Frontend.*

Abstract

The Diagnostic, Psychopedagogical Research, and Inclusion Support Unit (UDIPSAI) faces challenges in patient management due to the analog handling of information, which complicates data access and increases vulnerability. To address this issue, patient management software has been developed to enable efficient registration, storage, and retrieval of information, improving service quality and preventing revictimization. The development process entailed the collection of functional and non-functional requirements adhering to the FURPS+ quality standard, the creation of a user interface for interaction, and the delivery of training based on didactic materials. The development followed the agile SCRUM framework, enabling an iterative and incremental process that involved requirements gathering, planning and development in sprints, deployment, stakeholder training, and product refactoring. The enhancement of the analog process entailed optimizing access to patient information, thereby enhancing the security of UDIPSAI's data and leading to an increase in the efficiency of patient management while reducing data duplication. The evaluation of the application revealed that 100% of users rated it between 4 and 5 on the Likert scale, and 66.6% reported a reduction in the time required to complete forms and records.

Keywords: *Diagnostic Unit, Psychopedagogical Research, Inclusion Support, Revictimization, Frontend.*

**Desarrollo de Frontend para el Sistema de Gestión de
Pacientes en UDIPSAI**

**Frontend Development for UDIPSAI's Patient Management
System**

Introducción

En un mundo donde los sistemas informáticos son indispensables para la gestión eficiente de información, las instituciones enfrentan el desafío de adaptarse a la transformación digital para optimizar procesos, garantizar la seguridad de los datos y mejorar la calidad del servicio. La Unidad de Diagnóstico, Investigación Psicopedagógica y Apoyo a la Inclusión (UDIPSAI), según Sanmartín (2023).

Brinda apoyo pedagógico y psicopedagógico a estudiantes desde el nivel inicial hasta la educación superior, especialmente aquellos con necesidades educativas especiales como dislexia, dislalia o discapacidad intelectual. Su objetivo principal es promover la inclusión y el desarrollo integral de sus usuarios en el sistema educativo.

En la actualidad, UDIPSAI se enfrenta a serios problemas debido a la gestión manual de la información de más de 4250 pacientes, lo que incluye el almacenamiento de datos en carpetas físicas gestionadas individualmente por los especialistas. Este método presenta limitaciones significativas: dificulta el acceso rápido a la información, incrementa el riesgo de extravío o uso no autorizado de datos sensibles y en situaciones delicadas como el maltrato, contribuye a la revictimización de los pacientes al obligarlos a repetir información innecesariamente. Estos problemas se agravan aun al considerar que la mayoría de los usuarios son menores de edad, por lo que es aún más urgente de contar con un sistema seguro y eficiente.

A nivel nacional, la implementación de sistemas similares ha mostrado resultados positivos al adoptar patrones arquitecturales como MVC y frameworks tecnológicos como Laravel. Por ejemplo, Marín et al. (2022) desarrollaron sistemas para la gestión de citas médicas, mientras que Cajas et al. (2023) diseñaron aplicaciones específicas para niños con dificultades de aprendizaje. Inspirados por estas iniciativas, el presente proyecto se orientó en el desarrollo de un sistema de gestión digital para UDIPSAI que optimice la administración de datos, reduzca los errores asociados al manejo manual y facilite el acceso rápido y seguro a la información.

El sistema fue desarrollado mediante tecnologías como: Java para la construcción del backend y JavaScript para el frontend, con el marco de trabajo ágil SCRUM. Este enfoque permitió realizar iteraciones rápidas, ofrecer feedback continuo de los stakeholders y desarrollo de un sistema ajustado a las necesidades de los usuarios. Asimismo, se proporcionó capacitación al personal con materiales didácticos y videotutoriales garantizando de esta forma un uso y manejo efectivo del sistema.

Metodología

El proyecto se estructuró en torno a una metodología de investigación exploratoria que permitió seleccionar herramientas tecnológicas idóneas para garantizar la calidad y la eficiencia del sistema. Este enfoque integral, respaldado por el marco de trabajo SCRUM y el estándar de calidad FURPS, garantizó que el sistema cumpliera con los estándares técnicos y las expectativas de los

especialistas, y contribuyó de manera significativa a la modernización de los procesos de UDIPSAI.

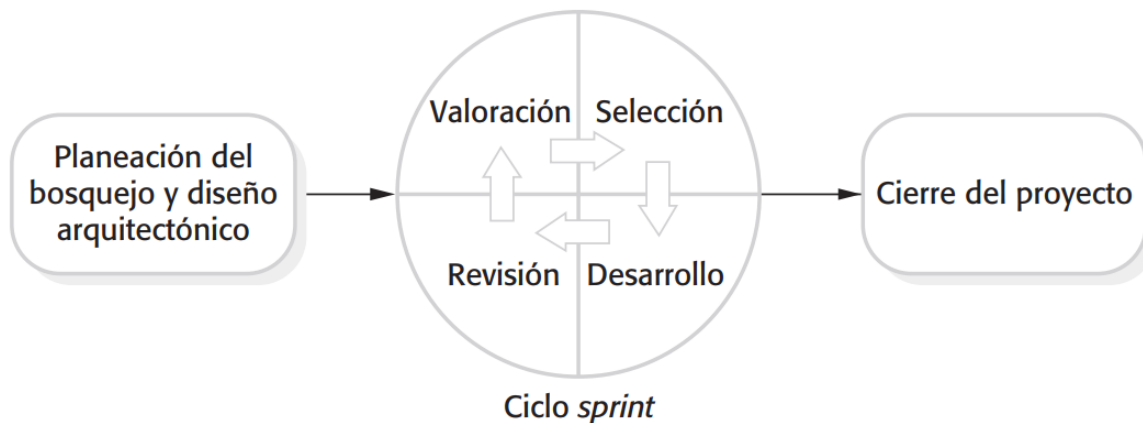
Enfoque de Desarrollo

En el proyecto se utilizó un enfoque ágil basado en el marco de trabajo SCRUM, reconocido por su eficiencia en la gestión de proyectos dinámicos. Velasco (2021), comenta que SCRUM prioriza las interacciones humanas sobre los procesos, convirtiéndose en una metodología ideal para el desarrollo de software. Según Sommerville (2011), SCRUM se estructura en tres fases generales, como se ilustra en la Figura 1:

1. Planeación del bosquejo: Define los objetivos generales del proyecto y establece la arquitectura del software.
2. Ciclos Sprint: Permiten la entrega incremental del sistema mediante iteraciones organizadas en cuatro pasos. Valoración, selección, desarrollo y revisión.
3. Cierre del proyecto: Incluye la documentación, capacitación de los especialistas de UDIPSAI mediante tutoriales y el despliegue del sistema.

Figura 1

Fases generales del marco de trabajo SCRUM, incluyendo la planeación inicial, un ciclo iterativo de sprint compuesto por valoración, selección, desarrollo y revisión, y el cierre del proyecto, destacando su naturaleza cíclica y adaptable.



Fuente: Sommerville (2011)

Con el objetivo de asegurar que el sistema se adapte a las necesidades específicas de UDIPSAI, se implementaron una serie de entrevistas con stakeholders de la institución, para identificar los requisitos funcionales y no funcionales. Los requisitos iniciales recopilados fueron refinados y clasificados mediante la metodología FURPS+, que organiza los atributos de calidad del software en diferentes categorías.

- Funcionalidad: Capacidades específicas del sistema.
- Usabilidad: Facilidad de interacción y navegación.

- **Fiabilidad:** Estabilidad y robustez del sistema.
- **Rendimiento:** Eficiencia en el procesamiento de datos.
- **Soporte:** Mantenimiento y adaptabilidad futura.

La clasificación, como menciona Sañay et al. (2023), avaló un desarrollo alineado con los objetivos técnicos y operativos del proyecto.

Diseño Centrado en el Usuario

De acuerdo con Espin-Loachamin et al. (2022), el diseño centrado en el usuario es una filosofía orientada a satisfacer necesidades humanas a través de interfaces claras y adaptativas. Para este proyecto, se implementó la librería **Ant Design (antd)**, que cumple con las **Web Content Accessibility Guidelines (WCAG)**, mejorando la accesibilidad de la aplicación. Además, se integraron principios de **UX**, como la **Usabilidad Estética** y la **Ley de Conectividad Uniforme** (Yablonski, 2022), optimizando la interacción del usuario mediante interfaces consistentes y de bajo esfuerzo cognitivo.

Arquitectura

El sistema se diseñó bajo el modelo de **Cliente-Servidor**, una arquitectura versátil y modular que permite centralizar la información en una base de datos. Según Singh et al. (2023), esta estructura separa el **frontend**, encargado de la interacción del usuario, y el **backend**, que gestiona la lógica de negocio. Yadav (2009) destaca que esta arquitectura es ideal para aplicaciones modernas debido a su escalabilidad y eficiencia en la distribución de servicios. El servidor, basado en **REST**, soporta funciones como generación de archivos PDF y gestión de usuarios.

Herramientas de Desarrollo

Para la selección de herramientas, se consideraron índices como el **PYPL** (Carbonnelle, 2023) y la encuesta global **State of JS 2023** (Greif, 2024). Las principales herramientas empleadas en el proceso de desarrollo son:

- **JavaScript:** Un Lenguaje dinámico fundamental para el desarrollo de aplicaciones web interactivas y eficientes (MDN Web Docs, 2024).
- **React.js:** Biblioteca que utiliza un **DOM virtual** para optimizar las actualizaciones, mejorando la modularidad y escalabilidad del desarrollo (Levlin, 2020).
- **Next.js:** Framework que simplifica el desarrollo fullstack, ofreciendo renderizado híbrido y optimización automática (Next.js, 2024).
- **Zustard:** Librería para el manejo eficiente de estados, integrada con el almacenamiento local del navegador.

- **next-translate-plugin:** Plugin para la gestión multilingüe mediante archivos JSON, proporcionando escalabilidad para nuevos idiomas.
- **Ant Design (antd):** Un sistema modular de diseño que incluye componentes preconfigurados para tablas, layouts y barras laterales, facilitando la creación de interfaces funcionales y atractivas (Ferreira et al., 2023).
- **Axios:** Librería para realizar llamadas HTTP eficientes hacia APIs REST (Pokharel, 2024).
- **Playwright:** Framework de pruebas automatizadas que asegura la calidad de flujos complejos (Knott, 2024).
- **Kanban:** Herramienta visual que organiza el flujo de trabajo y facilita la colaboración dentro del marco **SCRUM** (Behnam, 2022).
- **TAMAPP:** Modelo de aceptación tecnológica que evalúa la adopción del sistema mediante encuestas estructuradas (Campoverde-Calle et al., 2024).

El uso de dichas herramientas y metodologías permitió desarrollar un sistema accesible, escalable y centrado en las necesidades del usuario, cumpliendo con los objetivos del proyecto, permitiendo optimizar los procesos internos de UDIPSAI.

Desarrollo de FrontEnd con SCRUM.

La ejecución del proyecto se realizó en diez sprints, cada uno con una duración de dos semanas. En el sprint uno, se llevaron a cabo entrevistas con los stakeholders con el propósito de recopilar los requisitos principales. Como resultado, se delimitaron las funcionalidades del sistema, entre las que destacan el registro de pacientes, la gestión de especialistas y usuarios, así como la creación de fichas médicas por especialidad. Seguido, se implementó el Product Backlog mediante la herramienta Trello, facilitando la aplicación del marco de trabajo Scrum. En paralelo, se desarrolló el diseño general del sistema, la estructura de la base de datos y el prototipado inicial de la aplicación.

Análisis de Requisitos

En el contexto de las entrevistas realizadas en el Sprint uno, el objetivo principal fue identificar la problemática de la UDIPSAI y determinar la solución del producto de software. Como resultado se obtuvo requisitos funcionales y no funcionales [aplicación del FURPS+] que se lista a continuación:

Funcionalidad (F):

F1 - Registro y gestión de pacientes mediante una ficha digitalizada.

F2 - Implementar búsquedas por cédula de identidad, número de ficha o nombre.

F3 - Permitir acceso a información específica de pacientes para especialistas.

F4 - Crear formularios especializados (historia clínica, terapia de lenguaje, fonoaudiología, psicología clínica y educativa).

F5 - Generar y registrar seguimientos de citas con detalles específicos.

F6 - Registrar el historial de cambios en las fichas, con valores anteriores y nuevos.

F7 - Generar archivos PDF a partir de fichas especializadas.

F8 - Digitalizar y almacenar resultados de pruebas en formato PDF.

F9 - Gestionar la información de especialistas (crear, listar, actualizar, eliminar).

F10 - Asignar o eliminar pasantes para especialistas.

F11 - Permitir a los usuarios cambiar sus contraseñas.

F12 - Garantizar la seguridad en el almacenamiento y uso de datos.

Usabilidad (U):

U1 - Diseñar una interfaz intuitiva y accesible.

U2 - Proporcionar documentación clara para facilitar el uso del sistema.

U3 - Garantizar el acceso remoto para especialistas.

Confiabilidad (R):

R1 - Mantener un historial de cambios en las fichas de pacientes.

R2 - Asegurar la disponibilidad del sistema para acceso remoto.

Desempeño (P):

P1 - Optimizar el tiempo de respuesta en consultas y operaciones CRUD.

P2 - Garantizar la eficiencia en almacenamiento y recuperación de datos.

Soporte (S):

S1 - Proporcionar soporte técnico para instalación y configuración.

S2 - Capacitar a los especialistas en el uso del sistema.

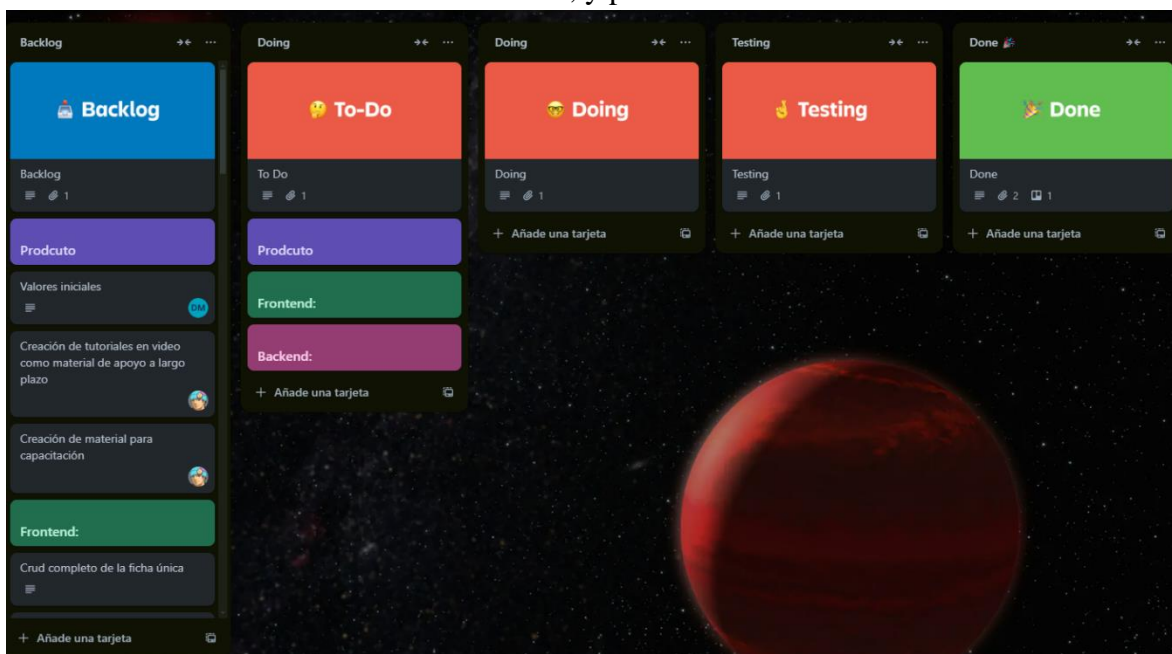
S3 - Ofrecer tutoriales en video para resolver dudas.

S4 - Brindar soporte post - implantación durante tres meses.

Con los requisitos identificados, se estableció el Product Backlog. Las historias de usuario [escenarios o requerimientos funcionales] fueron desarrolladas considerando el estándar de calidad FURPS+, incorporando estimaciones de esfuerzo y priorización. Además, durante el Sprint 1, se creó un tablero Kanban en Trello (Figura 2), con columnas denominadas Backlog, To-Do, Doing, Testing y Done. Estas columnas se organizaron por tipo de tarea: desarrollo frontend, backend y producto.

Figura 2

Tablero Kanban, utilizado a lo largo del proyecto con las columnas Backlog, To-Do, Doing, Testing, y Done, junto con separadores para los tipos de tarea, ya sean de desarrollo backend, frontend, y producto.



Fuente: Creación Propia

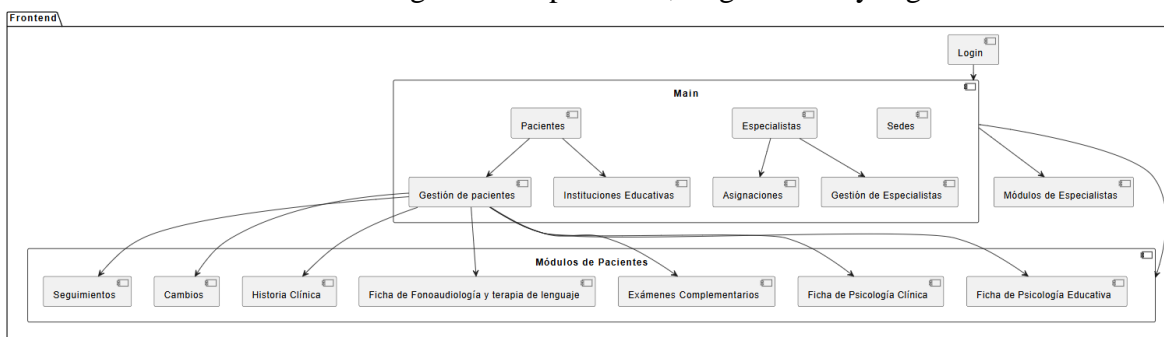
En el ámbito de la gestión de datos, se implementó el sistema de bases de datos relacional MySQL como motor, esto debido a su capacidad de establecer consultas eficientes. En el desarrollo backend, se utilizó el framework Spring, el cual emplea Hibernate como ORM (Object Relational Mapping) para gestionar las interacciones con la base de datos mediante la definición de clases. El diseño general de la base datos incluye tablas principales como "Paciente", "Especialista" y "Ficha Médica", conectadas a otras tablas secundarias como "Jornada", "Seguimiento", "Test" y "Especialidad". Cada entidad contiene atributos claves, como identificadores únicos (id) y relaciones entre tablas para garantizar integridad referencial. Por ejemplo, la tabla "Paciente" está vinculada a "Instituciones Educativas", "Jornada" y "Sede", mientras que "Especialista" se relaciona con "Especialidad" y "Permisos". Este diseño soporta funcionalidades como asignación de pacientes a pasantes, gestión de historiales médicos y generación de seguimientos.

Para finalizar, en el diseño de la solución en el Sprint 1, se elaboró un diagrama de componentes (Figura 3) que ilustra los módulos del sistema. Este incluye un módulo de inicio de sesión (Login),

encargado de la autenticación, que redirige a un módulo principal denominado Main. Desde Main se accede a los módulos de Pacientes, Especialistas y Sedes, los cuales cuentan con submódulos específicos. Por ejemplo, el módulo de Pacientes incluye funcionalidades para gestionar instituciones educativas y la información de cada paciente. El módulo de Especialistas, a su vez, permite la gestión de información y asignaciones de pacientes a pasantes.

Figura 3

Diagrama de Componentes, el diagrama muestra el flujo del sistema de gestión de usuarios para UDIPSAI, comenzando con el módulo de "Login" que dirige al menú principal ("Main"). Desde allí se accede a los módulos de "Pacientes", "Especialistas" y "Sedes", que incluyen funcionalidades como gestión de pacientes, asignaciones y seguimientos.



Fuente: Creación Propia

Diseño de Prototipos

En el Sprint 1, se realizó el prototipado de los módulos del sistema utilizando Figma para la maquetación de las pantallas. Se siguió el sistema de diseño Antd; además se emplearon datos de prueba en la creación de interfaces limpias con elementos reconocibles (Figura 4).

Figura 4

Interfaz maquetada, perteneciente a Creación de pacientes desarrollado en Figma utilizando el sistema de diseño de antd.

Universidad Católica de Cuenca

ADMIN ADMIN [Salir](#)

Inicio / Pacientes / Nuevo ▾

Ficha Única: Nueva Ficha

Información Personal

Fecha de apertura de la ficha
mm/dd/yyyy

¿Pertenece a proyecto?

Subir Imagen
[Subir Imagen](#)

Sede
CUENCA ▾

Fuente: Creación Propia

Implementación

Del Sprint 1 al 4, se desarrollaron operaciones CRUD para pacientes (F1, F2, F3) y submódulos como seguimientos (F5), exámenes complementarios (F8) y registro de cambios (F6, R1). También se implementaron funcionalidades de carga masiva de datos desde Excel, inicio de sesión y asignación de pacientes a especialistas (F10). En respuesta a nuevos requerimientos, se incluyó la gestión de sedes e instituciones educativas, optimizando la lógica del sistema.

En los Sprints 5 al 9, se diseñaron fichas de especialidad (F4) basadas en documentos de especialistas, se agregó un historial detallado de cambios y se generaron documentos PDF (F7) con Jasper Reports.

En el Sprint 10, se desplegó el sistema (S1) siguiendo los pasos: instalación de nginx, npm y pm2; clonación del repositorio en el servidor; y configuración del puerto 3000 con nginx. El build de la aplicación combinó páginas estáticas y dinámicas, mejorando el rendimiento. Además, se ejecutó la capacitación (S2) incluyó presentaciones y entrega de videotutoriales.

El desarrollo se completó en un período de cinco meses (10 sprints), lo que demuestra una capacidad de ejecución eficiente, cumpliendo con los requisitos establecidos y adaptándose a nuevos requerimientos según el feedback recibido.

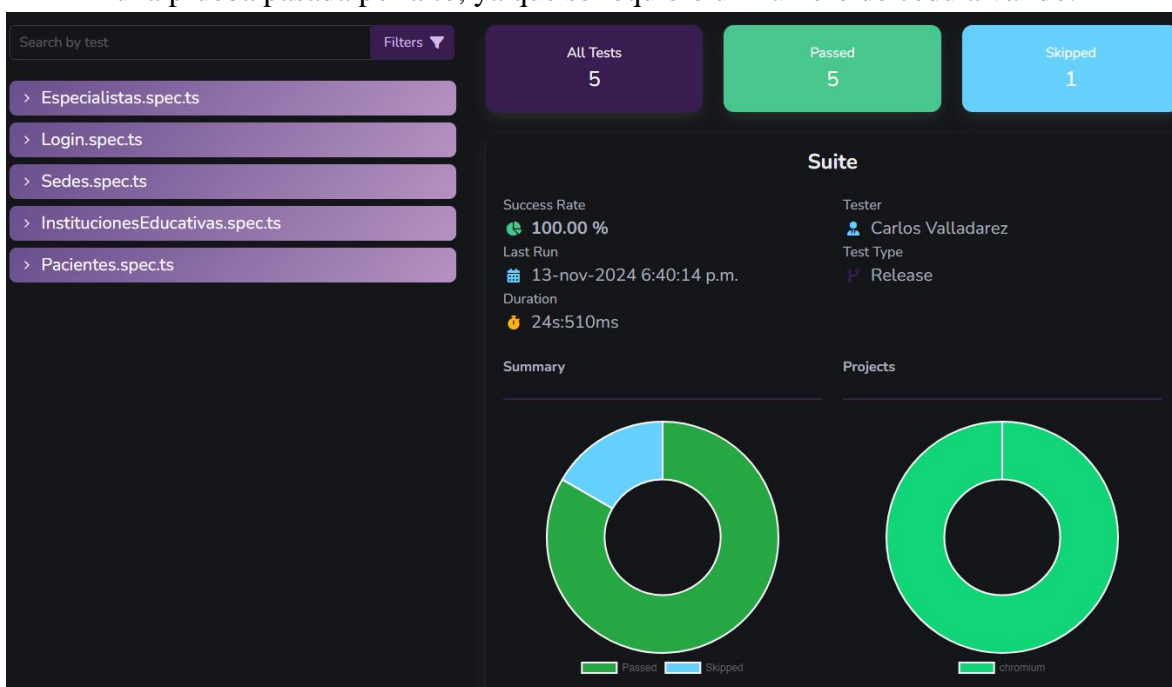
Resultados

Pruebas

Durante el ciclo de desarrollo del proyecto, se aplicaron pruebas automatizadas end-to-end utilizando el framework de testing Playwright, como se muestra en la Figura 5. Para facilitar la construcción de estas pruebas, se empleó la extensión para Visual Studio Code de Playwright, la cual ofrece herramientas que permiten al desarrollador “grabar” las acciones realizadas en la interfaz. Esta funcionalidad permite replicar las acciones de usuario y adicionalmente permite la comparación del contenido de las páginas, la verificación de la aparición de mensajes de error y la confirmación que los datos visualizados sean correctos.

Figura 5

Resultados de pruebas automatizadas, elaboradas con Playwright en las que se contemplan los módulos de especialistas, el inicio de sesión, el módulo de sedes, instituciones educativas y gran parte del módulo de gestión de pacientes, con una tasa de éxito del 100%, se puede identificar una prueba pasada por alto, ya que se requiere un número de cédula válido.



Fuente: Creación Propia

Si bien es cierto que esto facilita el desarrollo, no significa que este proceso sea fácil ni rápido, motivo por que muchos desarrollos optan por no realizar pruebas automatizadas. En este desarrollo en cuestión, se realizaron pruebas generales de los módulos de Sedes, Instituciones Educativas, Especialistas, y Pacientes. Un inconveniente de este framework es que no cuenta con una función permita medir la cobertura de las pruebas del código. Este proceso permitió identificar fallos que fueron corregidos, tales como la falta de validación de datos o errores en el flujo de la aplicación.

Despliegue

El sistema se desplegó según lo planificado y fue utilizado durante un mes. Para evaluar la satisfacción de los usuarios, se realizaron encuestas basadas en la matriz de evaluación del TAMAPP.

Esta encuesta fue realizada a los asistentes a la capacitación [son los usuarios de la aplicación], quienes pertenecen a las áreas de Coordinación, Secretaría, Psicología Educativa, Fonoaudiología y Terapia de Lenguaje, Recuperación Pedagógica y Estimulación temprana. Cabe resaltar que actualmente no hay ningún especialista del área de Psicología Clínica que utilice el sistema.

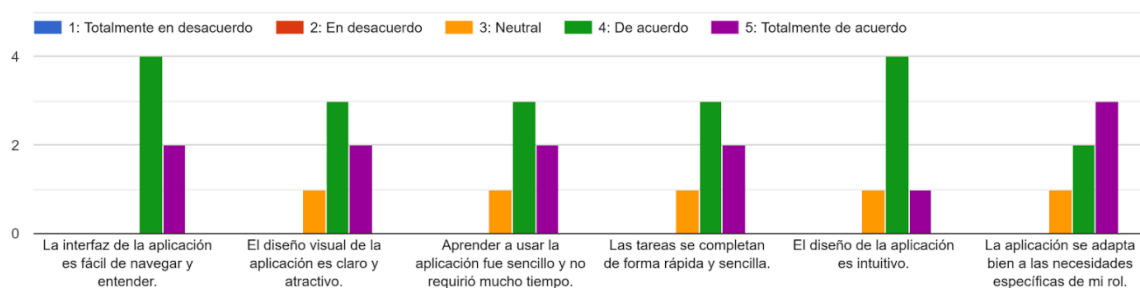
Evaluación del sistema

Usabilidad y Navegación

La percepción general del sistema es positiva, siendo valorado como fácil de usar, intuitivo y claro en su diseño, adaptándose adecuadamente a las necesidades de los roles de los encuestados. Al menos el 83.3% de los participantes calificaron con 4 o 5 en este apartado (Figura 6).

Figura 6

Resultados de Usabilidad y Navegación, con valoraciones del 83.3% mayormente positivas de entre 4 y 5.



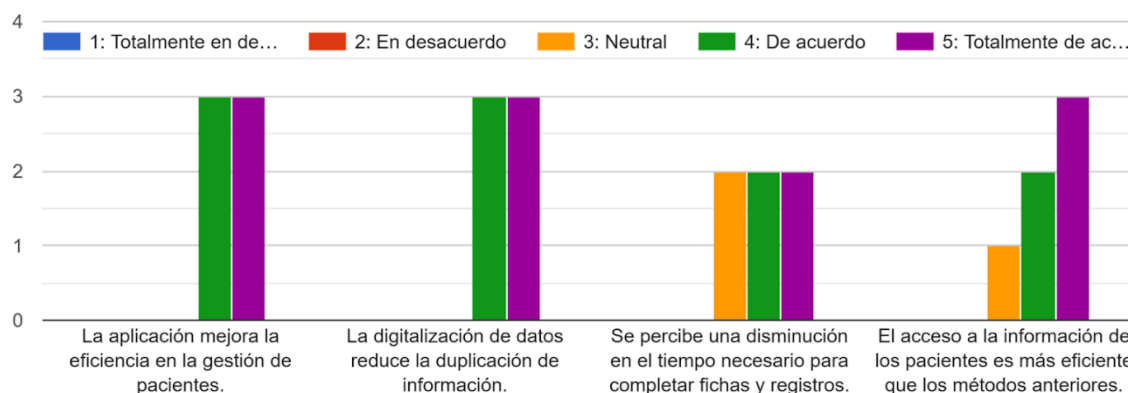
Fuente: Creación Propia

Eficiencia

El sistema mostró una mejora significativa en la gestión de pacientes, reduciendo la duplicidad de información y optimizando el tiempo para completar las fichas. Los resultados de la encuesta indican puntuaciones equilibradas entre 4 (50%) y 5 (50%) en la eficiencia del sistema (figura 7). Además, se registran mejoras en el acceso a la información con una valoración alta.

Figura 7

Resultados de Impacto en la Eficiencia, donde se puede rescatar una percepción en la mejora de la Eficiencia en la mayoría de los apartados, con valoraciones de entre 4 y 5.



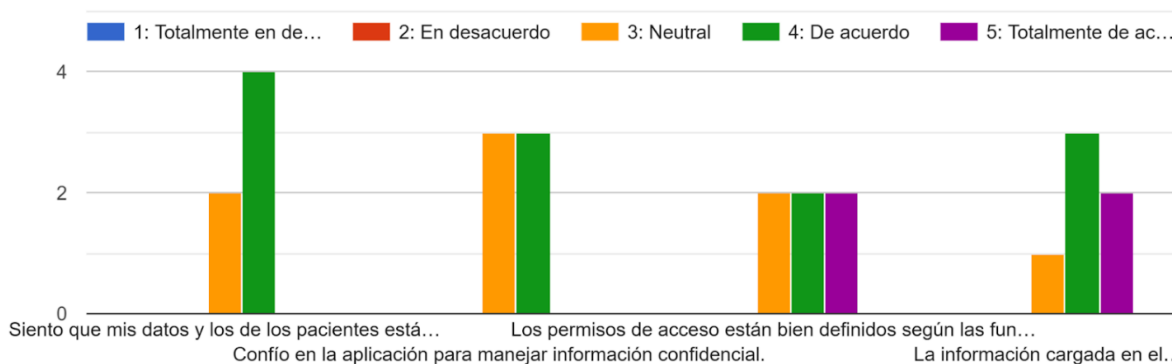
Fuente: Creación Propia

Seguridad y Confidencialidad

Si bien la percepción en esta dimensión fue más neutral comparada con las otras secciones, los resultados en general son positivos. Esto podría corresponder a que la seguridad y confidencialidad de la información no son tan tangibles para los usuarios (Figura 8).

Figura 8

Resultados de Seguridad y Confidencialidad, donde muchos de los resultados son neutrales, aunque sobresalen las valoraciones de 4, dando a entender que los encuestados sienten que la información se encuentra más segura.



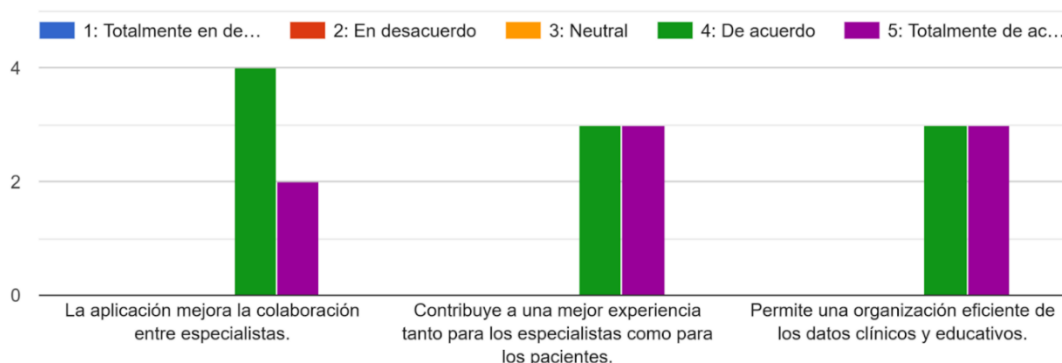
Fuente: Creación Propia

Colaboración y Experiencia del Usuario

La evaluación revela que el sistema mejoró la colaboración entre especialistas y la experiencia de usuario, tanto para ellos como para los pacientes. También permitió una organización eficiente de los datos. Los resultados de este apartado son positivos (Figura 9).

Figura 9

Resultados de Colaboración y Experiencia del Usuario, con valoraciones de 4 y 5 en un 100% describiendo una mejora considerable en la experiencia de usuario y colaboración entre especialistas.



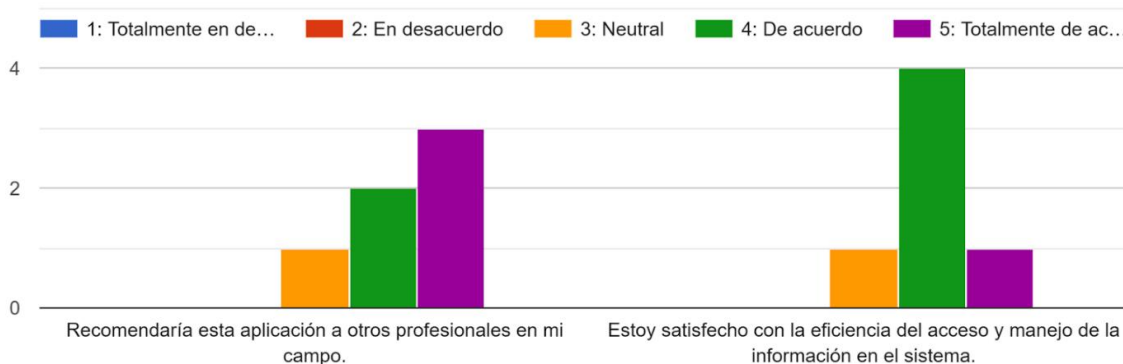
Fuente: Creación Propia

Retroalimentación General

En términos de rendimiento, el sistema fue evaluado satisfactoriamente especialmente en acceso y manejo de información. (Figura 10), lo que derivó en que el 83.3% de los encuestados consideran recomendar el sistema a otros profesionales.

Figura 10

Resultados de Retroalimentación General, sobresale la satisfacción del usuario al punto de estar de acuerdo en recomendar el uso del sistema a otros especialistas.



Fuente: Creación Propia

En resumen, la percepción de los usuarios de la aplicación ha sido satisfactoria observándose mejora en la eficiencia de los procesos relacionados con la gestión de pacientes, así como en la seguridad de la información. La usabilidad del sistema se ha interpretado como buena, con una interfaz fácil de entender, atractiva e intuitiva. Se alcanzaron los objetivos planteados del proyecto de gestión de pacientes de UDIPSAI que permita registrar, almacenar y acceder a la información de manera eficiente y segura, mejorando la calidad del servicio y reduciendo la revictimización de los pacientes, gracias a la mejora de la accesibilidad a la información.

Discusión

En el <<post – desarrollo>> de la aplicación es indispensable reflexionar sobre los conocimientos adquiridos, en primer lugar, la comunicación del equipo. El marco de trabajo, SCRUM se erige como una herramienta fundamental, propiciando una comunicación constante y facilitando espacios para la reflexión sobre los avances realizados, así como feedback constante por parte del usuario para mejorar la calidad del producto.

Además, el uso del estándar de calidad FURPS y Trello [herramienta de gestión] permitió una correcta gestión del proyecto, y la generación de entregables al final de cada uno de los Sprint [ciclos de SCRUM]. También, se evidenció cuán necesario resulta crear un sistema fácil de usar e intuitivo, sin embargo, esto puede sobrecargar la interfaz volviéndola tediosa o redundante. Para evitarlo, es necesario la retroalimentación [Feedback de los potenciales usuarios], ya que, permite probar y validar el software.

La ejecución de los Sprint permitió liberar versiones hasta alcanzar la funcionalidad total del producto [software: fácil de utilizar e intuitivo], con el uso de la librería antd y las guías de UX para mejorar la experiencia del usuario. El prototipado de la aplicación en la herramienta Figma permitió mejorar la calidad [usabilidad] del producto final, adaptándose al Feedback de los usuarios. La aplicación de design system [antd] en la interfaz gráfica de los usuarios, resultó en una aplicación visualmente atractiva, consistente, fácil de implementar y adaptable a un enfoque multiplataforma [aplicación responsiva].

El uso de Next.js si bien, facilita muchas funcionalidades en el desarrollo como el enrutamiento, o el data fetching del lado del servidor, también generó roces en el desarrollo. Dada su naturaleza de generación híbrida [Server Side Rendering, y Client Side Rendering], no es compatible con gestores de estado como Zustand, ya que, estos solo funcionan del lado del cliente, al menos sin utilizar un middleware, lo que da lugar a problemas en la comunicación entre el servidor y el cliente. Esto propició la necesidad de verificar la información disponible en el cliente de la aplicación desde el backend de Next.js, lo cual causaba errores. Este problema se resolvió identificando si el módulo en el que se llamaba a Zustand se encontraba en el lado del cliente.

El realizar pruebas automatizadas con Playwright facilitó la identificación de errores a lo largo del proyecto, permitiendo validar el producto y asegurar la calidad final.

La implementación de la función multilingüe supuso un cambio total de paradigma. Inicialmente se intentó implementar manualmente, utilizando un archivo JSON y llamando a variables, pero resultó complicado mantener el estado a lo largo de la aplicación debido a la naturaleza propia de Next.js. Debido a esto se optó por utilizar una librería similar [**next-translate-plugin**] que se encarga de identificar el idioma mediante la URL. Es decir, si se accede a /home en otro idioma [por ejemplo: para el idioma inglés, acceder /en/home]. Se menciona un cambio de paradigma al momento de programar, ya que todos los textos se transforman en variables que serán llaman en

un archivo JSON, aunque solo es cuestión de acostumbrarse. Lamentablemente no se encontró un plugin que ayudará a mejorar la calidad de vida al programar con este nuevo paradigma.

En cuanto al despliegue de la aplicación el principal problema fueron los cortes de energía eléctrica que ocurrieron en Ecuador en el año 2024, esto dificultó el despliegue, y la disponibilidad del sistema por el mismo motivo.

Los usuarios de UDIPSAI, valoraron positivamente el sistema de gestión de pacientes. Las calificaciones obtenidas mediante la encuesta aplicada para la evaluación del sistema oscilaron entre 3 y 5 valoraciones en la escala Likert, indicando la eficiencia, y efectividad en la gestión de la información de los pacientes.

Conclusiones

Conclusiones

1. El desarrollo frontend del sistema de gestión de pacientes para la Unidad de Diagnóstico, Investigación Psicopedagógica y Apoyo a la Inclusión ha resultado en una implementación exitosa de un sistema. Este desarrollo ha cumplido con todos los objetivos establecidos. La encuesta realizada a los usuarios de UDIPSAI indica que la satisfacción respecto al sistema es muy elevada y que existe un aumento en la eficiencia de los procesos que realizan, como la gestión de los pacientes, o el tiempo de registro de la información.
2. El sistema permite a sus usuarios registrar, almacenar y acceder de manera eficiente y segura a la información de sus pacientes de manera eficiente y segura, reduciendo la necesidad de solicitar constantemente información a los pacientes y por ende reduciendo la revictimización de los pacientes con casos graves.
3. Se consiguió recopilar de forma efectiva los requerimientos funcionales y no funcionales aplicando el estándar FURPS, necesarios para el desarrollo del sistema, y se recibió una feedback constante, la cual, permitió mantener el norte en el desarrollo de la aplicación.
4. El empleo de la metodología ágil SCRUM para el desarrollo del frontend permitió mejorar la calidad de servicios que UDIPSAI brinda a sus pacientes, mediante la implementación de interfaces intuitivas y adaptadas a las necesidades de los usuarios. El uso de SCRUM facilitó una rápida adaptación a nuevos requerimientos mediante el feedback constante recibido.
5. Se elaboró material audiovisual para la capacitación y consulta de los especialistas actuales y futuros de la unidad de diagnóstico, ayudando a que estos puedan dar un uso eficaz del sistema.

Recomendaciones

1. A largo plazo, el sistema tiene el potencial de expandirse; en el sentido de que la base de datos puede ser utilizada en diferentes proyectos, lo que permite el acceso tanto a la información de pacientes como a la de especialistas.

2. Se puede seguir trabajando en nuevos módulos que ayuden a los especialistas en tareas específicas, se pueden desarrollar aplicativos, así como la creación de aplicaciones destinadas a los pacientes, así como aplicativos lúdicos, o exámenes digitalizados.

3. Se recomienda seguir trabajando en el sistema y seguir las sugerencias obtenidas en la encuesta, por ejemplo: una opción para agregar nuevos campos a los formularios de forma, la incorporación de recomendaciones basadas en pruebas, presumiblemente mediante el uso de inteligencia artificial y el registro del uso de la aplicación por parte de los especialistas, ingresando la fecha y hora del inicio de sesión.

Referencias bibliográficas

- Behnam, H. (2022). Detecting and comparing Kanban boards using Computer Vision.
- Campoverde-Calle, M. S., Sañay-Sañay, S. I., & Cabrera-Duffaut, A. E. (2024). Desarrollo de Metodología de Validación de APPS basada en el Modelo de Aceptación Tecnológica: Caso de estudio “BodyUC.” *MQRInvestigar*, 8(1), 3742–3770. <https://doi.org/10.56048/mqr20225.8.1.2024.3742-3770>
- Carbonnelle, P. (2023). PYPL PopularitY of Programming Language index. Github.Io. <https://pypl.github.io/PYPL.html>
- Design values - Ant Design. (2022). Ant.deSign. <https://ant.design/docs/spec/values>
- Espin-Loachamin, A. I., Iza-Carrera, D. H., & Paredes-Amaguaya, A. I. (2022). Diseño centrado en el usuario para la creación de un catálogo de productos de consumo masivo. *Polo del Conocimiento*, 7(4), 650-661.
- Ferreira, Fabio, and Marco Tulio Valente. (2023) "Detecting code smells in React-based Web apps." *Information and Software Technology* 155 (2023): 107111.
- Greif, S., y Burel, E. (2024). State of JavaScript 2023. Stateofjs.com. Recuperado el 22 de octubre de 2024, de <https://2023.stateofjs.com/>
- Knott, S. (2024). Npm: Playwright. Npm. <https://www.npmjs.com/package/playwright>
- Levlin, M. (2020). DOM benchmark comparison of the front-end JavaScript frameworks React, Angular, Vue, and Svelte.
- Marín, M. Á. F., & Tolmo, D. G. (2022). Sistema informático Web para la gestión de citas e historial médico de pacientes. *Metropolitana de Ciencias Aplicadas*, 5(1), 8.
- Matute-Álvarez, J. I., Tenén-Banegas, A. D., Sañay-Sañay, S. I., & Gaona-Pineda, J. S. (2023). Sistema de realidad virtual para la gestión de la visualización e interacción con los ecosistemas del Bioparque AMARU Cuenca, Ecuador. *MQRInvestigar*, 7(3), 3313–3338. <https://doi.org/10.56048/mqr20225.7.3.2023.3313-3338>
- Mozilla. (2024.). JavaScript documentation. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- Next.js. (2024.). Next.js Documentation. <https://nextjs.org/docs>
- Pokharel, B. (2024). A Centralized Dashboard for DevOps. Theseus.Fi. https://www.theseus.fi/bitstream/handle/10024/870125/Pokharel_Bikarna.pdf?sequence=3

- Sanmartín, F. S. (2023, 5 junio). UDIPSAI: un servicio especializado de evaluación psicopedagógica para la inclusión y adaptación al sistema educativo. Universidad Católica de Cuenca. <https://www.ucacue.edu.ec/udipsai-un-servicio-especializado-de-evaluacion-psicopedagogica-para-la-inclusion-y-adaptacion-al-sistema-educativo/>
- Scrum.org. (2020). What is Scrum? Scrum.org. <https://www.scrum.org/learning-series/what-is-scrum/>
- Singh, G., Javed, M., & Dhaliwal, B. K. (2022). Full Stack Web Development: Vision Challenges and Future Scope. *International Research Journal of Engineering and Technology (IRJET)*, 9.
- Sommerville, I. (2011). *Ingeniería del software* (M. I. A. Galipienso, A. Botia Martinez, F. Mora Lizan, & J. P. Trigueros Jover, Trans.; 7th ed.). Pearson Education.
- Velasco, M. V. E., Villacis, J. A. N., Chávez, P. R. S., & Cuchipec, W. C. C. (2021). Revisión sistemática de la metodología SCRUM para el desarrollo de Software. *Dominio de las Ciencias*, 7(4), 54.
- Yablonski, J. (n.d.). Home. Laws of UX. Retrieved December 14, 2024, from <https://lawsofux.com/es/>
- Yadav, S. C. (2009). *An introduction to client/server computing*. New Age International.