



Formando líderes para la construcción  
de un nuevo país en paz

**Aplicación web para la creación de portafolios con experiencias inmersivas utilizando la metodología SCRUM para mejorar la presentación de los proyectos hechos por los estudiantes de ingeniería de sistemas**

Juan Pablo Marquez Sanchez

1004922828

Ingeniería del Software III

Jose Gerardo Chacon Rangel

Universidad de Pamplona

Facultad de Ingeniería y Arquitectura

Ingeniería de Sistemas

Villa del Rosario – Norte de Santander

2025



## Tabla de contenido

1.	Resumen .....	8
2.	Abstract .....	9
3.	Introducción .....	10
4.	Justificación .....	12
5.	Planteamiento del problema .....	13
5.1	Pregunta de investigación de su proyecto .....	15
6.	Objetivos.....	15
6.1	Objetivo general .....	15
6.2	Objetivos específicos .....	15
7.	Estado del arte.....	16
7.1	Marco referencial.....	16
7.1.1	Canva: .....	16
7.1.2	Wix.....	16
7.1.3	Plan Estratégico de Tecnologías de la Información (PETI) - Norte de Santander	17
7.1.4	BeBetter Developers (Región Andina) .....	17
7.1.5	Creasotol (Cúcuta) .....	18
7.1.6	Solweb (Cúcuta) .....	18
7.2	Marco conceptual.....	19
7.2.1	Portafolio digital .....	19
7.2.2	Next.js.....	19
7.2.3	Node.js.....	19



7.2.4	TypeScript.....	20
7.3	Marco Contextual .....	20
7.4	Marco Legal .....	20
7.4.1	La Ley 1581 de 2012 (Ley de Tratamiento de Datos):.....	20
7.4.2	Ley 527 de 1999 (Ley de Comercio Electrónico):.....	21
8.	Metodología .....	22
8.1	Metodología de desarrollo .....	22
8.2	Fases del proyecto:.....	24
8.2.1	Planificación del producto (Product Backlog & Sprint 0) .....	24
8.2.2	Sprint Planning & Definición de Requisitos .....	24
8.2.3	Desarrollo iterativo e incremental (Sprints de implementación) .....	24
8.2.4	Revisión del Sprint y Retrospectiva.....	25
8.3	Patrón de diseño de software .....	25
9.	Singleton.....	26
9.1	¿Porque usar Singleton en este proyecto?.....	26
9.2	Problema:.....	26
9.2.1	Garantizar que una clase tenga una única instancia .....	26
9.2.2	Proporcionar un punto de acceso global a dicha instancia. ....	27
9.3	Solución: .....	28
9.4	Principales beneficios:.....	28
10.	Desarrollo .....	29



10.1	Para el desarrollo del objetivo específico 1: “Investigar las tecnologías y herramientas adecuadas para el desarrollo de una aplicación web que permita la creación de portafolios con experiencias inmersivas.” .....	29
10.1.1	Tecnologías Frontend .....	29
10.1.2	Tecnologías Backend.....	31
10.2	Para el desarrollo del objetivo específico 2: “Definir los requisitos de la aplicación web, considerando las necesidades de los estudiantes de ingeniería de sistemas para la presentación de sus proyectos definiendo una arquitectura por capas para su implementación.” .....	33
10.2.1	Requisitos Funcionales .....	34
10.2.2	Requisitos No Funcionales.....	34
10.2.3	Definición de tecnología frontend .....	34
10.2.4	Definición de tecnología backend.....	35
10.2.5	Arquitectura por capas .....	35
10.3	Para el desarrollo del objetivo específico 3: “Crear una aplicación web con arquitectura por capas que integre experiencias inmersivas para la generación de portafolios interactivos, facilitando la visualización y comprensión de los proyectos desarrollados” .....	36
10.3.1	Capa de Presentación:.....	37
10.3.2	Presentación de interfaces de usuario: .....	41
10.3.3	Presentación del portafolio generado .....	50
10.3.4	Backend:.....	54



10.3.5	Capa de Navegación.....	56
10.3.6	Capa de Base de Datos .....	60
10.4	Para el desarrollo del objetivo específico 4: “Validar la efectividad de la aplicación web mediante pruebas con estudiantes, evaluando su impacto en la presentación de los proyectos académicos.” .....	63
10.4.1	Pruebas de interfaz: .....	63
10.4.2	Retroalimentación: .....	66
10.4.3	Pruebas de software: .....	67
11.	Conclusiones .....	70
12.	Bibliografía.....	73

### Tabla de tablas

<b>Tabla 1,</b>	<i>Comparación entre los diferentes tipos de patrones de diseño.....</i>	<i>25</i>
<b>Tabla 2,</b>	<i>Tabla de comparación entre tecnologías Frontend.....</i>	<i>30</i>
<b>Tabla 3,</b>	<i>Tabla comparativa entre tecnologías Backend.....</i>	<i>32</i>

### Tabla de ilustraciones

<b>Ilustración 1,</b>	<i>Metodología scrum .....</i>	<i>22</i>
<b>Ilustración 2,</b>	<i>Ejemplo del problema que resuelve Singleton. ....</i>	<i>27</i>
<b>Ilustración 3,</b>	<i>Tecnología frontend más demandadas.....</i>	<i>30</i>
<b>Ilustración 4,</b>	<i>Funcionamiento Frontend y Backend.....</i>	<i>32</i>
<b>Ilustración 5,</b>	<i>Arquitectura por capas.....</i>	<i>36</i>
<b>Ilustración 6,</b>	<i>Estructura de carpetas en NextJS .....</i>	<i>37</i>



<b>Ilustración 7, Carpeta APP en NextJS .....</b>	<b>37</b>
<b>Ilustración 8, Carpeta COMPONENTS en NextJS .....</b>	<b>38</b>
<b>Ilustración 9, Carpeta LIB en NextJS .....</b>	<b>39</b>
<b>Ilustración 10, Esquemas del proyecto en NextJS .....</b>	<b>39</b>
<b>Ilustración 11, Carpeta UI en NextJS.....</b>	<b>40</b>
<b>Ilustración 12, Pantalla principal de la aplicación web.....</b>	<b>42</b>
<b>Ilustración 13, Pantalla de detalles de la aplicación web.....</b>	<b>42</b>
<b>Ilustración 14, Pantalla de precios .....</b>	<b>43</b>
<b>Ilustración 15, Pantalla de login .....</b>	<b>44</b>
<b>Ilustración 16, Pantalla de registro de usuario .....</b>	<b>45</b>
<b>Ilustración 17, Pantalla del dashboard .....</b>	<b>46</b>
<b>Ilustración 18, Sección de presentación.....</b>	<b>46</b>
<b>Ilustración 19, Sección de proyectos .....</b>	<b>47</b>
<b>Ilustración 20, Sección de redes sociales .....</b>	<b>48</b>
<b>Ilustración 21, Sección de colores .....</b>	<b>49</b>
<b>Ilustración 22, Sección de usuario .....</b>	<b>49</b>
<b>Ilustración 23, Sección de presentación generada.....</b>	<b>50</b>
<b>Ilustración 24, Sección de proyectos generada.....</b>	<b>51</b>
<b>Ilustración 25, Sección de proyectos en un proyecto abierto .....</b>	<b>52</b>
<b>Ilustración 26, Sección de redes sociales .....</b>	<b>53</b>
<b>Ilustración 27, Estructura MVC en el backend .....</b>	<b>54</b>
<b>Ilustración 28, Configuración de rutas y CORS del backend.....</b>	<b>56</b>
<b>Ilustración 29, Definición de las rutas del usuario .....</b>	<b>57</b>
<b>Ilustración 30, Definición de las rutas de perfiles .....</b>	<b>57</b>

*“Formando líderes para la construcción de un nuevo país en paz”*



<b>Ilustración 31, Definición de las rutas de proyectos .....</b>	<b>58</b>
<b>Ilustración 32, Definición de las rutas de redes sociales .....</b>	<b>59</b>
<b>Ilustración 33, Definición de las rutas de tecnologías.....</b>	<b>59</b>
<b>Ilustración 34, Definición de las rutas de los colores.....</b>	<b>60</b>
<b>Ilustración 35, Estructura de la base de dato en la API.....</b>	<b>60</b>
<b>Ilustración 36, Diagrama de la composición de la base de datos .....</b>	<b>62</b>
<b>Ilustración 37, Prueba de la aplicación con un estudiante.....</b>	<b>63</b>
<b>Ilustración 38, Prueba del registro de usuario con un estudiante .....</b>	<b>64</b>
<b>Ilustración 39, Prueba de generación de portafolio con un estudiante .....</b>	<b>65</b>
<b>Ilustración 40, Pruebas de la API en cada entidad de la base de datos.....</b>	<b>67</b>
<b>Ilustración 41, Pruebas de todas las consultas de la entidad Usuario.....</b>	<b>68</b>
<b>Ilustración 42, Pruebas de todas las consultas de la entidad Proyectos .....</b>	<b>69</b>





## 1. Resumen

El mercado laboral para ingenieros de sistemas sin experiencia requiere una presentación efectiva de sus proyectos para demostrar sus habilidades. Esta plataforma mejorará la visibilidad del talento estudiantil, fomentará la empleabilidad y fortalecerá la conexión con el sector productivo. El objetivo general es desarrollar una aplicación web para la creación de portafolios inmersivos que faciliten la presentación de proyectos estudiantiles, promoviendo la innovación, la empleabilidad y el acceso a oportunidades profesionales. Se aplicará Scrum, dividido en: 1) Planificación del producto (backlog inicial, objetivos), 2) Sprint planning (definición de tareas por iteración), 3) Desarrollo iterativo (implementación de funcionalidades en cada sprint) y 4) Revisión y retrospectiva (evaluación y mejoras continuas). Además, se usará el patrón Singleton para gestionar la configuración global y el acceso a datos, asegurando eficiencia y escalabilidad en la aplicación. Los datos se obtendrán de proyectos estudiantiles, prácticas académicas y trabajos de investigación de estudiantes de Ingeniería de Sistemas, integrando documentación, multimedia y modelos interactivos que reflejen el proceso de aprendizaje y desarrollo tecnológico. Como resultado, se desarrollará una aplicación web que permitirá a los estudiantes crear portafolios digitales con experiencias inmersivas, facilitando su presentación ante docentes, empresas y comunidades académicas. Esta plataforma no solo optimizará la forma en que los estudiantes exhiben sus proyectos, sino que también servirá como un repositorio de conocimiento accesible para futuras generaciones, promoviendo la colaboración y la mejora continua en la educación y la industria tecnológica.

**Palabras clave:** Portafolios digitales, educación inmersiva, empleabilidad, innovación.





## 2. Abstract

The job market for systems engineers without experience requires an effective presentation of their projects to demonstrate their skills. This platform will improve the visibility of student talent, foster employability and strengthen the connection with the productive sector. The overall objective is to develop a web application for the creation of immersive portfolios that facilitate the presentation of student projects, promoting innovation, employability and access to professional opportunities. Scrum will be applied, divided into: 1) Product planning (initial backlog, objectives), 2) Sprint planning (definition of tasks by iteration), 3) Iterative development (implementation of functionalities in each sprint) and 4) Review and retrospective (evaluation and continuous improvement). In addition, will use the Singleton pattern to manage the global configuration and access to data, ensuring efficiency and scalability in the application. The data will be obtained from student projects, academic internships and research work of Systems Engineering students, integrating documentation, multimedia and interactive models that reflect the process of learning and technological development. As a result, a web application will be developed that will allow students to create digital portfolios with immersive experiences, facilitating their presentation to teachers, companies and academic communities. This platform will not only optimise the way in which students showcase their projects, but will also serve as a repository of accessible knowledge for future generations, promoting collaboration and continuous improvement in education and the technology industry.

**Keywords:** digital portfolios, immersive education, employability, innovation.



### 3. Introducción

En el contexto actual del mercado laboral, los ingenieros de sistemas recién egresados enfrentan un gran reto: demostrar sus capacidades técnicas y creativas en un entorno altamente competitivo sin contar aún con experiencia profesional. Esta situación ha generado una creciente necesidad de contar con herramientas efectivas que permitan a los estudiantes y egresados exponer de forma estructurada, visual y profesional sus proyectos académicos, investigaciones y prácticas preprofesionales. En respuesta a esta necesidad, el presente proyecto propone el desarrollo de una aplicación web orientada a la creación de portafolios digitales inmersivos que integren documentación, recursos multimedia y elementos interactivos. Esta plataforma está pensada para mejorar la visibilidad del talento estudiantil, promover la innovación académica, facilitar el acceso a oportunidades laborales y fortalecer la conexión entre la academia y el sector productivo.

Diversos estudios recientes han resaltado la importancia de los portafolios digitales como herramienta para la empleabilidad. Los portafolios digitales permiten a los estudiantes documentar su progreso, reflejar habilidades transversales y demostrar competencias adquiridas durante su formación, (López-Meneses, 2022). Además, los portafolios fomentan la autonomía del estudiante y permiten una evaluación auténtica basada en evidencias reales del aprendizaje. En el ámbito de las carreras tecnológicas, donde los proyectos prácticos constituyen una parte esencial del proceso formativo, contar con un medio de presentación profesional se vuelve indispensable para resaltar el nivel de innovación y aplicabilidad de los desarrollos realizados por los estudiantes, (Hernández-Carrera, 2023).

El desarrollo de esta plataforma se llevará a cabo utilizando la metodología ágil Scrum, ampliamente reconocida por su enfoque iterativo e incremental para la gestión de proyectos de



software. Scrum permite adaptar el desarrollo a las necesidades cambiantes de los usuarios y facilita la entrega de valor continuo a través de sprints definidos. Scrum mejora la colaboración entre los miembros del equipo y aumenta la calidad del producto al centrarse en la retroalimentación continua y la mejora progresiva, (Ahmad, 2021). En este proyecto, se estructurará en cuatro etapas clave: planificación del producto (backlog inicial y objetivos generales), planificación de sprint (definición de tareas por iteración), desarrollo iterativo (implementación de funcionalidades por sprint) y revisión y retrospectiva (evaluación y ajustes). Este enfoque metodológico no solo garantiza eficiencia y control, sino también una mayor participación de los usuarios durante todo el ciclo de vida del proyecto.

Como resultado, se desarrollará una aplicación web funcional y accesible que permitirá a los estudiantes construir portafolios digitales inmersivos. Esta plataforma no solo mejorará la manera en que se exhiben los proyectos estudiantiles, sino que también se consolidará como un repositorio abierto de conocimiento útil para docentes, investigadores y nuevos estudiantes. En última instancia, se busca que esta iniciativa contribuya al fortalecimiento de la formación profesional en Ingeniería de Sistemas, fomente la innovación educativa, y estreche los lazos entre la academia y el sector productivo. La estructura del presente documento se organiza en cuatro capítulos. El capítulo 1 presenta el resumen, la introducción, la justificación y los objetivos del estudio. En el capítulo 2 se desarrollan los marcos referenciales: teórico, conceptual, contextual y legal. El capítulo 3 describe la metodología empleada para alcanzar los objetivos propuestos. Finalmente, el capítulo 4 expone los resultados obtenidos, así como las conclusiones del trabajo y la bibliografía consultada.



#### 4. Justificación

En la actualidad, la digitalización de la educación y la industria exige que los estudiantes de Ingeniería de Sistemas presenten sus proyectos de manera innovadora y accesible. Sin embargo, muchos alumnos enfrentan dificultades para destacar sus habilidades y conocimientos en un entorno altamente competitivo. Una aplicación web para la creación de portafolios con experiencias inmersivas permitirá a los estudiantes proyectar sus capacidades de manera más efectiva, facilitando la conexión con empresas y el sector productivo. De este modo, se fomentará la empleabilidad y el reconocimiento del talento emergente, contribuyendo al desarrollo tecnológico y social de la comunidad académica y profesional.

El aprendizaje basado en proyectos es una metodología ampliamente utilizada en la formación de ingenieros, ya que permite la aplicación práctica de conocimientos teóricos. No obstante, la falta de herramientas adecuadas para la presentación de estos proyectos limita su impacto y alcance. La implementación de una plataforma que integre experiencias inmersivas mejorará la documentación y exhibición del trabajo estudiantil, incentivando la creatividad y la innovación. Además, servirá como un repositorio de conocimientos y soluciones tecnológicas que podrán ser utilizados por futuras generaciones de estudiantes e investigadores.

El uso de la metodología Scrum en el desarrollo de esta aplicación garantiza un proceso ágil, iterativo y adaptable a las necesidades de los usuarios. Scrum promueve la colaboración, la mejora continua y la entrega de valor en cortos períodos de tiempo, asegurando un producto final de alta calidad. Implementar esta metodología no solo optimiza la creación de la plataforma, sino que también introduce a los estudiantes en un marco de trabajo ampliamente utilizado en la industria del software, preparándolos para enfrentar desafíos reales en el ámbito laboral.



Este proyecto contribuye a los ODS, en particular al Objetivo 4: "Educación de calidad", al proporcionar a los estudiantes una herramienta que potencia su aprendizaje y visibilidad en el mercado laboral. También impacta el Objetivo 8: "Trabajo decente y crecimiento económico", ya que facilita la conexión entre los talentos emergentes y las oportunidades profesionales. Asimismo, promueve el Objetivo 9: "Industria, innovación e infraestructura", al impulsar el uso de tecnologías avanzadas en la educación y la presentación de proyectos. (UNESCO, 2015)

Desde un punto de vista práctico, esta aplicación web solucionará la problemática de la limitada exposición de proyectos estudiantiles en formatos tradicionales como documentos o diapositivas. Mediante el uso de tecnologías interactivas e inmersivas, los portafolios digitales mejorarán la forma en que los estudiantes presentan sus trabajos, haciéndolos más atractivos y comprensibles para docentes, reclutadores y empresas. Además, al centralizar estos portafolios en una plataforma digital, se facilitará el acceso, la actualización y el intercambio de información, potenciando la difusión de conocimientos y la colaboración académica y profesional.

## 5. Planteamiento del problema

La adecuada presentación y difusión de proyectos académicos resulta clave para el reconocimiento del talento y la innovación dentro de las comunidades universitarias. En carreras como Ingeniería de Sistemas, donde se desarrollan soluciones tecnológicas, es esencial que estos trabajos trasciendan las aulas y lleguen a públicos más amplios, incluidos futuros empleadores y otros desarrolladores. La visibilidad de los proyectos estudiantiles no solo fortalece el portafolio profesional de los estudiantes, sino que también promueve la cultura de la innovación. (Ramírez, 2022). Además, muchas iniciativas valiosas no se aprovechan plenamente por falta de plataformas efectivas que conecten estos desarrollos con escenarios



reales de aplicación, lo cual representa una pérdida de potencial tanto para los estudiantes como para el entorno académico y empresarial. (Torres, 2023).

En muchas universidades, incluyendo los programas de Ingeniería de Sistemas, los proyectos académicos desarrollados por los estudiantes se presentan únicamente en eventos internos o son entregados como productos finales que rara vez son compartidos más allá del aula. Esto limita considerablemente su alcance, ya que no existen plataformas digitales efectivas ni metodologías que fomenten una difusión sistemática y atractiva de estos trabajos. A pesar de los avances tecnológicos y del acceso a herramientas modernas de desarrollo web, muchos proyectos permanecen guardados en repositorios o carpetas personales, sin lograr una exposición real o retroalimentación externa.

Adicionalmente, la mayoría de los estudiantes no cuentan con un espacio propio en línea donde puedan construir su identidad profesional a través de un portafolio bien estructurado y visualmente atractivo. Aunque existen plataformas como GitHub, estas no siempre son intuitivas o personalizables para mostrar una narrativa clara de sus logros. Esta carencia también impide a docentes y empresas conocer con mayor profundidad las capacidades técnicas y creativas de los futuros profesionales. Se espera contar con una herramienta digital que permita a los estudiantes crear portafolios interactivos e inmersivos, donde puedan integrar no solo los aspectos técnicos de sus proyectos, sino también experiencias visuales y narrativas que reflejen su proceso creativo. Esta plataforma debe permitir una personalización amigable, visualmente atractiva y capaz de vincularse con tecnologías actuales como inteligencia artificial, realidad aumentada o modelos de presentación 3D, para captar la atención de un público más amplio.





## 5.1 Pregunta de investigación de su proyecto

De este modo, se busca que los estudiantes tengan una mayor visibilidad dentro y fuera de la academia, facilitando tanto la retroalimentación de sus pares como la conexión con posibles oportunidades laborales. Asimismo, esta solución promoverá la sistematización y organización de sus avances académicos y profesionales en un solo lugar, impulsando así la cultura de autoevaluación, mejora continua y proyección profesional. Teniendo esto en cuenta se planteó la siguiente pregunta: ¿Cómo se puede mejorar la presentación y difusión de los proyectos desarrollados por los estudiantes de ingeniería de sistemas?

## 6. Objetivos

### 6.1 Objetivo general

Desarrollar aplicación web para la creación de portafolios con experiencias inmersivas utilizando la metodología scrum para mejorar la presentación de los proyectos hechos por los estudiantes de ingeniería de sistemas

### 6.2 Objetivos específicos

- Investigar las tecnologías y herramientas adecuadas para el desarrollo de una aplicación web que permita la creación de portafolios con experiencias inmersivas.
- Definir los requisitos de la aplicación web, considerando las necesidades de los estudiantes de ingeniería de sistemas para la presentación de sus proyectos definiendo una arquitectura por capas para su implementación.
- Crear una aplicación web con arquitectura por capas que integre experiencias inmersivas para la generación de portafolios interactivos, facilitando la visualización y comprensión de los proyectos desarrollados.





- Validar la efectividad de la aplicación web mediante pruebas con estudiantes, evaluando su impacto en la presentación de los proyectos académicos.

## 7. Estado del arte

### 7.1 Marco referencial

#### 7.1.1 Canva:

Canva es una plataforma de diseño gráfico en línea que ha transformado la manera en que los usuarios crean contenido visual, incluyendo portafolios digitales. Aunque su enfoque inicial fue en presentaciones y gráficos para redes sociales, ha evolucionado hacia herramientas de creación de portafolios visuales altamente personalizables. Los usuarios pueden escoger entre plantillas prediseñadas, modificar colores, tipografías y añadir imágenes o enlaces a proyectos propios. Este nivel de personalización, sin requerir conocimientos de programación, hace que Canva sea un referente internacional relevante. Es similar a tu aplicación porque también permite que los usuarios construyan una presencia en línea a través de una interfaz sencilla. Además, su capacidad para compartir portafolios mediante enlaces públicos o privados aporta funcionalidades clave que podrías considerar implementar. (Canva, 2025)

#### 7.1.2 Wix

Wix es un constructor de sitios web que se ha posicionado como uno de los más populares a nivel global por su facilidad de uso y sus múltiples herramientas para personalizar sitios, especialmente portafolios profesionales. Con Wix, los usuarios pueden crear su página desde cero o modificar plantillas específicas para diseñadores, desarrolladores, fotógrafos, entre otros. Su sistema de edición "drag and drop" permite crear secciones, subir imágenes, vincular proyectos y gestionar formularios de contacto. Wix se relaciona con tu proyecto porque



ofrece un entorno accesible para que personas sin experiencia técnica puedan gestionar su presencia digital. Además, integra funcionalidades como galerías, blogs y acceso a código personalizado para usuarios avanzados, lo cual amplía las posibilidades de personalización, similar a lo que podrías incluir en tu aplicación. (Wiz, 2025).

### **7.1.3 Plan Estratégico de Tecnologías de la Información (PETI) - Norte de Santander**

El PETI 2024-2027 de la Gobernación de Norte de Santander establece como uno de sus objetivos el fomento del uso de herramientas digitales y tecnologías de la información en todos los sectores del departamento. Aunque no es un proyecto de desarrollo de portafolios digitales como tal, promueve la creación de soluciones tecnológicas locales que respondan a necesidades específicas del territorio. Este contexto brinda un marco de apoyo institucional y una posible base de usuarios interesados en soluciones que fortalezcan su presencia digital. Para tu proyecto, el PETI puede ofrecer oportunidades de colaboración o financiamiento, y justifica la pertinencia regional de una herramienta como la tuya. (Gobernación de Norte de Santander, 2025).

### **7.1.4 BeBetter Developers (Región Andina)**

BeBetter Developers es una empresa ubicada en la región nororiental de Colombia que ofrece desarrollo de páginas web y soluciones tecnológicas para emprendimientos y profesionales. Entre sus proyectos, han construido sitios personalizados que sirven como portafolios para artistas, diseñadores y freelancers. Esto los convierte en un referente útil, pues muestran cómo se puede abordar la creación de portafolios desde una perspectiva técnica y comercial. BeBetter ha demostrado cómo adaptarse a las tendencias de diseño web modernas manteniendo la accesibilidad y la facilidad de mantenimiento para sus clientes, aspectos que



deberías considerar al construir tu plataforma. Además, sus desarrollos locales validan la viabilidad de proyectos como el tuyo en el contexto regional. (BeBetter Developers, 2025).

### 7.1.5 Creasotol (Cúcuta)

Creasotol es una empresa de desarrollo web localizada en Cúcuta, Norte de Santander, que se enfoca en construir soluciones digitales a medida para pequeñas y medianas empresas. Aunque no está centrada exclusivamente en portafolios digitales, muchos de sus servicios están dirigidos a profesionales y emprendedores que necesitan mostrar su trabajo en línea. Diseñan sitios web personalizados que pueden incluir secciones como “sobre mí”, galería de proyectos, y enlaces a redes sociales, muy similares a lo que plantea tu aplicación. El enfoque de Creasotol en la personalización y el acompañamiento al cliente refleja la importancia de construir herramientas intuitivas y adaptables a las necesidades locales, lo cual podría inspirarte a incluir funcionalidades específicas para el mercado colombiano o cucuteño. (Creasotol, 2025).

### 7.1.6 Solweb (Cúcuta)

Solweb es otra empresa tecnológica establecida en Cúcuta que ofrece diseño y desarrollo de páginas web y aplicaciones móviles. Su oferta incluye la creación de sitios auto gestionables que permiten a los usuarios actualizar su contenido sin depender de un desarrollador, lo que es esencial para un portafolio dinámico. Aunque sus servicios están más orientados a negocios, también atienden proyectos personales como páginas de presentación para profesionales independientes. La similitud con tu proyecto radica en su propuesta de brindar autonomía al usuario para gestionar su sitio. Además, la experiencia local que tienen les permite entender mejor las necesidades del entorno digital en Cúcuta, lo que puede servirte



como base para identificar funciones valiosas que tu plataforma debería incluir para usuarios de la región. (solweb, 2025).

## 7.2 Marco conceptual

### 7.2.1 Portafolio digital

Un portafolio digital es una herramienta que permite a los estudiantes y profesionales mostrar sus competencias, logros y proyectos a través de una plataforma en línea. Estos portafolios no solo sirven como repositorios visuales, sino también como espacios dinámicos que reflejan el crecimiento y evolución de su autor. Los portafolios digitales promueven la reflexión, la evaluación continua y la presentación estructurada de evidencias del aprendizaje. (Barrett, 2007).

### 7.2.2 Next.js

Next.js es un framework basado en React que permite el desarrollo de aplicaciones web optimizadas, escalables y modernas. Ofrece características como el renderizado del lado del servidor (SSR), generación de sitios estáticos (SSG) y enrutamiento automático, lo que permite construir sitios con mejor rendimiento y mayor visibilidad en buscadores. Next.js combina lo mejor del desarrollo frontend moderno en un solo entorno, lo que lo convierte en una opción ideal para interfaces rápidas y accesibles. (Vercel, 2023).

### 7.2.3 Node.js

Node.js es un entorno de ejecución para JavaScript en el servidor que permite desarrollar aplicaciones escalables, rápidas y eficientes gracias a su modelo asíncronico y orientado a eventos. Es especialmente útil para construir APIs RESTful y manejar múltiples solicitudes simultáneamente. Node.js ha revolucionado el backend al permitir el desarrollo con



JavaScript tanto en el cliente como en el servidor, promoviendo una arquitectura unificada. (Tilkov, 2022).

#### 7.2.4 TypeScript

TypeScript es un superset de JavaScript que introduce tipado estático, facilitando la detección temprana de errores, el mantenimiento del código y la escalabilidad en proyectos grandes. Su integración con Node.js y Next.js permite construir aplicaciones robustas y seguras. TypeScript mejora significativamente la experiencia de desarrollo al proporcionar herramientas de validación y autocompletado que reducen errores humanos. (Bierman, 2023).

### 7.3 Marco Contextual

Para este proyecto se plantea el desarrollo de una aplicación web destinada a la creación de portafolios digitales inmersivos, donde los estudiantes puedan presentar sus proyectos académicos, trabajos de investigación y experiencias en prácticas profesionales. Esta plataforma no solo facilitará la presentación estructurada e interactiva de dichos contenidos, sino que también fortalecerá la empleabilidad, fomentará la innovación y contribuirá al acercamiento entre la academia y la industria. Como resultado, se espera entregar una solución tecnológica que no solo beneficie a los estudiantes en su transición hacia el mundo laboral, sino que también funcione como un repositorio de conocimiento institucional, promoviendo la colaboración académica y la transferencia de saberes en el campo de la Ingeniería de Sistemas.

### 7.4 Marco Legal

#### 7.4.1 La Ley 1581 de 2012 (Ley de Tratamiento de Datos):

Esta ley tiene como finalidad regular el tratamiento de datos personales para garantizar y proteger el derecho fundamental de habeas data, que consiste en el derecho de las personas



a conocer, actualizar y rectificar la información que se haya recogido sobre ellas en bases de datos. Es necesario obtener el consentimiento previo, expreso e informado del titular de los datos para el tratamiento de sus datos personales. Además, Las organizaciones que recolecten y traten datos personales deben garantizar la seguridad de la información y prevenir su alteración, pérdida, consulta, uso o acceso no autorizado. (EL CONGRESO DE COLOMBIA, 2012).

#### **7.4.2 Ley 527 de 1999 (Ley de Comercio Electrónico):**

Esta ley establece el marco legal para el uso de mensajes de datos, el comercio electrónico y las firmas digitales, reconociendo su validez jurídica y probatoria. Esta ley fue promulgada con el propósito de adaptar el ordenamiento jurídico a los avances tecnológicos, brindando seguridad y confianza a las transacciones electrónicas. Define conceptos clave como mensaje de datos, firma digital y entidades de certificación, y regula su uso tanto en el ámbito privado como público. Asimismo, promueve el desarrollo del comercio electrónico, facilitando la modernización de procesos administrativos y comerciales. La ley otorga equivalencia funcional entre documentos electrónicos y físicos, siempre que se garantice su autenticidad, integridad y disponibilidad. Esta normativa ha sido fundamental para impulsar la transformación digital en Colombia y mejorar la eficiencia en los servicios digitales. Además, permite a las organizaciones optimizar recursos y reducir el uso del papel, fomentando prácticas sostenibles. (EL CONGRESO DE COLOMBIA, 1999).





## 8. Metodología

### 8.1 Metodología de desarrollo

La metodología que se usó para este proyecto es la metodología ágil SCRUM es un marco ágil para la gestión y desarrollo de proyectos, especialmente en software. Se basa en ciclos iterativos e incrementales llamados sprints y sigue un conjunto de eventos y roles definidos. Su implementación se puede dividir en tres grandes fases:

**Ilustración 1, Metodología scrum**



*Nota,* La ilustración representa las fases de la metodología scrum. Fuente tomada de (AVSVM cloud, 2024)

**Planificación del Proyecto (Pre-Sprint):** Antes de iniciar el desarrollo, se establecen las bases del proyecto.

- Definición del Product Backlog: Se crea una lista de características, requisitos y tareas necesarias para el producto, priorizadas por el Product Owner.
- Formación del Equipo Scrum: Se asignan los roles clave:
  - Scrum Master: Facilita el proceso y elimina impedimentos.





- Product Owner: Define y prioriza los requisitos.
- Equipo de Desarrollo: Desarrolladores encargados de implementar las funcionalidades.
- Estimación y Priorización: Se evalúan las tareas y se definen prioridades para el desarrollo.
- Sprint Planning: Se seleccionan los elementos del backlog que se desarrollarán en el primer sprint.

**Ejecución del Sprint (Desarrollo Iterativo):** Cada sprint es un ciclo de trabajo que dura entre 1 y 4 semanas e incluye varias actividades clave:

- Daily Scrum: Reunión diaria de 15 minutos donde el equipo responde tres preguntas clave: ¿Qué hice ayer?, ¿Qué haré hoy?, ¿Tengo algún impedimento?
- Desarrollo y Testing: El equipo trabaja en las tareas seleccionadas y realiza pruebas continuas.
- Sprint Review: Al final del sprint, se presenta el incremento del producto al Product Owner y a los interesados.
- Sprint Retrospective: El equipo analiza lo que funcionó bien y qué se puede mejorar en el próximo sprint.

**Cierre y Entrega (Post-Sprint):** Cuando se han completado suficientes sprints y el producto cumple con los criterios de aceptación:

- Entrega del Producto: Se lanza una versión funcional al usuario o cliente.
- Evaluación y Mejora: Se recopilan retroalimentaciones para futuras iteraciones o mejoras.



- Mantenimiento y Soporte: Si es necesario, se realizan correcciones o actualizaciones en el producto.

## 8.2 Fases del proyecto:

### 8.2.1 Planificación del producto (*Product Backlog & Sprint 0*)

Investigar las tecnologías y herramientas adecuadas para el desarrollo de la aplicación. Se realizará un análisis de las tecnologías disponibles (frameworks, librerías, bases de datos, herramientas de experiencias inmersivas) y se documentarán en un backlog inicial. También se definirán los criterios de aceptación para cada funcionalidad clave.

### 8.2.2 Sprint Planning & Definición de Requisitos

Definir los requisitos de la aplicación web considerando las necesidades de los estudiantes, además de establecer una arquitectura por capas para su implementación. Se organizarán reuniones con los stakeholders (estudiantes, docentes y posibles empleadores) para determinar los requerimientos funcionales y no funcionales. Luego, se diseñará la arquitectura por capas (capa de presentación, lógica de negocio y acceso a datos) asegurando modularidad y escalabilidad.

### 8.2.3 Desarrollo iterativo e incremental (*Sprints de implementación*)

Crear una aplicación web con arquitectura por capas que integre experiencias inmersivas para la generación de portafolios interactivos. Se dividirá el desarrollo en sprints donde se implementarán de forma iterativa las funcionalidades clave, tales como la creación y gestión de portafolios, integración de multimedia y elementos interactivos. Se aplicará el patrón Singleton para manejar configuraciones globales y optimizar el acceso a datos.



### 8.2.4 Revisión del Sprint y Retrospectiva

Validar la efectividad de la aplicación web mediante pruebas con estudiantes, evaluando su impacto en la presentación de los proyectos académicos. Se realizarán pruebas piloto con estudiantes para medir la usabilidad, accesibilidad e impacto de la plataforma en la presentación de sus proyectos. A partir del feedback recibido, se realizarán mejoras en los sprints siguientes para optimizar la experiencia del usuario.

## 8.3 Patrón de diseño de software

Los patrones de diseño de software son soluciones reutilizables para problemas comunes en el desarrollo de software. Estos patrones ayudan a mejorar la organización, reutilización y mantenimiento del código, permitiendo a los desarrolladores diseñar software más flexible y escalable.

**Tabla 1,** Comparación entre los diferentes tipos de patrones de diseño

Tipo de Patrón	Descripción	Ejemplo de Patrones
<b>Creacionales</b>	Se centran en la creación eficiente de objetos, evitando la dependencia directa de clases concretas.	Factory Method, Abstract Factory, Builder, Prototype, Singleton
<b>Estructurales</b>	Definen la composición y organización de las clases y objetos para formar estructuras más grandes.	Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy
<b>Comportamiento</b>	Se encargan de la comunicación y la asignación de responsabilidades entre objetos.	Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor



*Nota*, Esta tabla muestra una comparación de los propósitos de cada tipo de patrón de diseño, más adelante se explicarán cada patrón. Fuente de la tabla (OpenAI, 2025).

## 9. Singleton

Singleton es un patrón de diseño creacional que nos permite asegurarnos de que una clase tenga una única instancia, a la vez que proporciona un punto de acceso global a dicha instancia.

### 9.1 ¿Porque usar Singleton en este proyecto?

El patrón de diseño que se decidió usar es el SINGLETON, este garantiza que una clase tenga una única instancia en toda la aplicación y proporciona un punto de acceso global a esa instancia. Su uso es ideal en escenarios donde se necesita una única fuente de verdad, como en la gestión de configuración, acceso a bases de datos o control de recursos compartidos.

### 9.2 Problema:

El patrón Singleton resuelve dos problemas al mismo tiempo, vulnerando el Principio de responsabilidad única:

#### 9.2.1 *Garantizar que una clase tenga una única instancia*

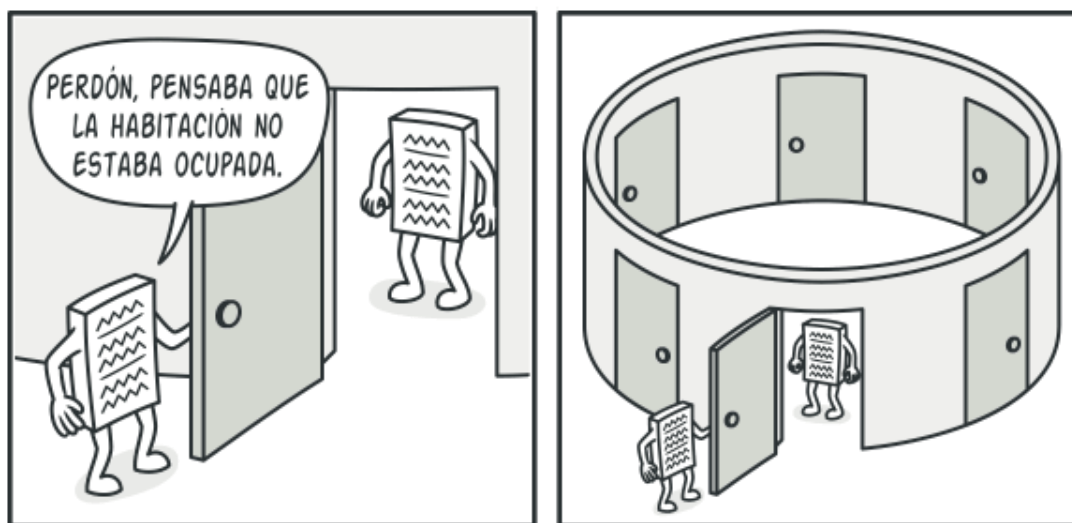
¿Por qué querría alguien controlar cuántas instancias tiene una clase? El motivo más habitual es controlar el acceso a algún recurso compartido, por ejemplo, una base de datos o un archivo.

Por ejemplo: imagina que has creado un objeto y al cabo de un tiempo decides crear otro nuevo. En lugar de recibir un objeto nuevo, obtendrás el que ya habías creado. Ten en cuenta que este comportamiento es imposible de implementar con un constructor normal, ya



que una llamada al constructor siempre debe devolver un nuevo objeto por diseño. (Refactoring Guru, 2021).

**Ilustración 2, Ejemplo del problema que resuelve Singleton.**



*Nota,* Puede ser que los clientes ni siquiera se den cuenta de que trabajan con el mismo objeto todo el tiempo. (Refactoring Guru, 2021)

### 9.2.2 Proporcionar un punto de acceso global a dicha instancia.

¿Recuerdas esas variables globales que utilizaste (bueno, sí, fui yo) para almacenar objetos esenciales? Aunque son muy útiles, también son poco seguras, ya que cualquier código podría sobrescribir el contenido de esas variables y descomponer la aplicación.

Al igual que una variable global, el patrón Singleton nos permite acceder a un objeto desde cualquier parte del programa. No obstante, también evita que otro código sobrescriba esa instancia. Este problema tiene otra cara: no queremos que el código que resuelve el primer problema se encuentre disperso por todo el programa. Es mucho más conveniente tenerlo dentro de una clase, sobre todo si el resto del código ya depende de ella.

*"Formando líderes para la construcción de un nuevo país en paz"*



Hoy en día el patrón Singleton se ha popularizado tanto que la gente suele llamar singleton a cualquier patrón, incluso si solo resuelve uno de los problemas antes mencionados. (Refactoring Guru, 2021)

### 9.3 Solución:

Todas las implementaciones del patrón Singleton tienen estos dos pasos en común:

- Hacer privado el constructor por defecto para evitar que otros objetos utilicen el operador new con la clase Singleton.
- Crear un método de creación estático que actúe como constructor. Tras bambalinas, este método invoca al constructor privado para crear un objeto y lo guarda en un campo estático.

Las siguientes llamadas a este método devuelven el objeto almacenado en caché.

Si tu código tiene acceso a la clase Singleton, podrá invocar su método estático. De esta manera, cada vez que se invoque este método, siempre se devolverá el mismo objeto.

### 9.4 Principales beneficios:

Evita la creación de múltiples instancias de una clase, lo que garantiza un comportamiento consistente en toda la aplicación. Al limitar la cantidad de instancias, se reducen los costos de memoria y procesamiento. Esto es útil en sistemas con recursos limitados. Cualquier parte del código puede acceder a la misma instancia sin necesidad de pasar referencias.

Se pueden implementar bloqueos para garantizar que la instancia se cree de manera segura en aplicaciones concurrentes. Se usa en configuraciones de aplicaciones, gestores de caché, registros de logs, conexiones a bases de datos y más. Facilita la gestión del código al evitar la necesidad de múltiples inicializaciones y control manual de instancias. Al tener una



única instancia, se evitan problemas relacionados con la duplicación de datos o la sincronización de estados en diferentes instancias.

## 10. Desarrollo

### 10.1 Para el desarrollo del objetivo específico 1: “Investigar las tecnologías y herramientas adecuadas para el desarrollo de una aplicación web que permita la creación de portafolios con experiencias inmersivas.”

En la actualidad, el desarrollo de aplicaciones web ha evolucionado significativamente gracias a la aparición de frameworks y tecnologías modernas que permiten construir interfaces ricas, dinámicas y altamente interactivas. Para lograr una experiencia inmersiva en la creación de portafolios digitales, es necesario realizar una investigación exhaustiva sobre las herramientas y tecnologías disponibles que puedan ofrecer soporte a componentes multimedia, interactividad avanzada y rendimiento optimizado tanto en dispositivos móviles como en entornos de escritorio. La elección de las tecnologías adecuadas debe estar basada en criterios de usabilidad, escalabilidad, facilidad de integración y soporte comunitario, para asegurar la sostenibilidad del sistema a largo plazo.

#### 10.1.1 Tecnologías Frontend

Entre las tecnologías frontend más utilizadas para este tipo de aplicaciones se encuentran React, Vue y Angular, las cuales permiten la construcción de interfaces de usuario reactivas y modulares. React, por ejemplo, destaca por su flexibilidad y gran ecosistema de componentes que facilitan la integración de librerías para animaciones, visualizaciones 3D y experiencias interactivas. Estas herramientas se complementan con tecnologías CSS modernas como TailwindCSS o Sass, que permiten personalizar el diseño y la presentación visual del portafolio de forma rápida y eficiente.



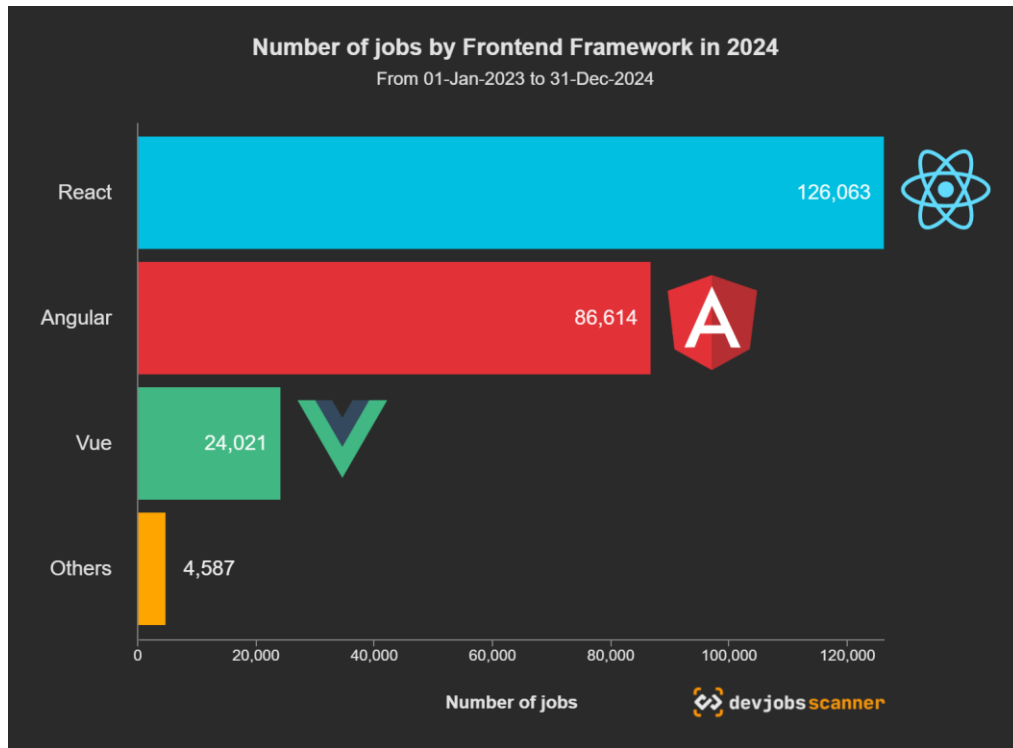


**Tabla 2, Tabla de comparación entre tecnologías Frontend**

Tecnología	Facilidad de Uso	Curva de Aprendizaje	Rendimiento	Soporte para Experiencias Inmersivas
React	Alta	Moderada	Alto	Excelente (soporte con Three.js, WebXR)
Vue.js	Muy alta	Baja	Alto	Buena (compatible con librerías de 3D y animación)
Angular	Media	Alta	Muy alto	Buena (requiere configuración más compleja)
Next.js	Alta	Moderada	Muy alto	Excelente (ideal para renderizado SSR y experiencias avanzadas)

*Nota,* Esta tabla muestra una comparación entre las diferentes frameworks frontend más usados y recomendados del mundo. Fuente tomada de (Emadamerho-Atori, 2024).

**Ilustración 3, Tecnología frontend más demandadas.**





*Nota: La anterior ilustración en un gráfico de barras presenta las tecnologías con mayor número de ofertas de trabajo. (Logan, 2025)*

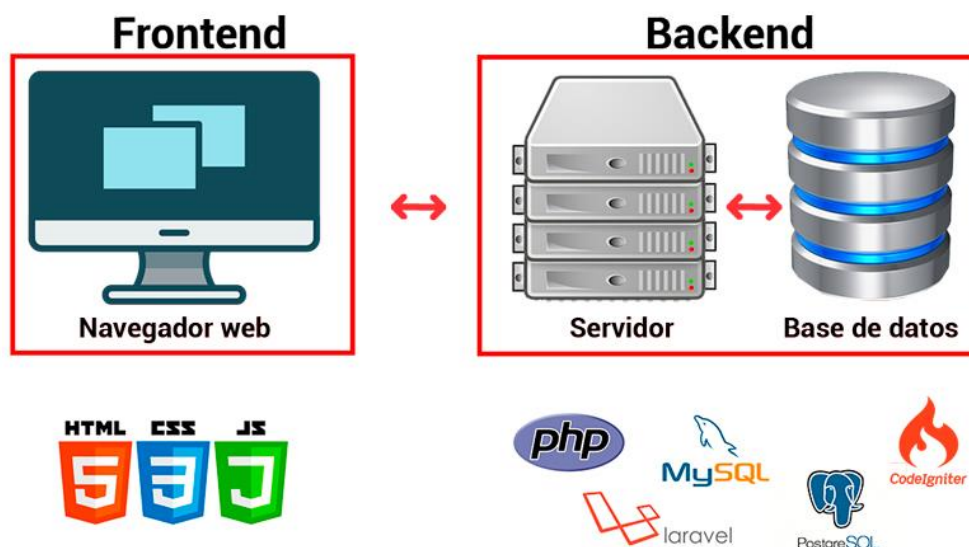
Después de analizar durante 12 meses alrededor de ~250k ofertas de trabajo frontend, encontramos que React es el framework frontend más demandado con un total de 126k ofertas de trabajo contabilizadas. Angular se sitúa en segundo lugar, con un total de 87k ofertas. En tercera posición encontramos a Vue, con 24k ofertas. «Otros» representa todos los demás frameworks frontend, como Svelte, Solid.js, Alpine, Backbone, Lit... A pesar de que algunos de ellos son bastante populares ahora mismo, la demanda real de ofertas de trabajo es realmente baja. (Logan, 2025).

### 10.1.2 Tecnologías Backend

El Backend es en pocas palabras el área lógica de toda web, app, o programa. El Backend se refiere a la arquitectura interna del sitio y asegura que todos los elementos funcionen de forma idónea. Y lo más importante, NO ES VISIBLE a los ojos del usuario final, que muchas veces ni siquiera es consciente del reto que implica. El Backend no tiene ningún elemento visual, sino que se refiere al área lógica de la web. Es la rama de arquitectura interna del sitio, la función de ingeniería que los programadores desarrollamos, es la funcionalidad propia de la web. No se ve, pero está ahí y es vital para que la web funcione, mucho más que la tipografía y el copyright. (Suratica, 2024).



**Ilustración 4, Funcionamiento Frontend y Backend**



*Nota:* La anterior ilustración modela el funcionamiento del backend y el frontend además de algunas tecnologías que predominan en cada uno. Fuente tomada de (Suratica, 2024)

En cuanto al backend, Node.js, Django y Spring Boot son opciones destacadas que proporcionan robustez en la gestión de datos y escalabilidad en entornos de producción. Node.js, por su naturaleza asincrónica y su compatibilidad con JavaScript, facilita la construcción de API REST eficientes para el manejo de proyectos, archivos multimedia y perfiles de usuario. Asimismo, bases de datos como MongoDB y PostgreSQL ofrecen opciones adecuadas para almacenar información estructurada y no estructurada de los portafolios, desde textos y documentos hasta imágenes y videos.

**Tabla 3, Tabla comparativa entre tecnologías Backend**

Tecnología	Facilidad de Uso	Curva de Aprendizaje	Escalabilidad	Integración con Frontend
------------	------------------	----------------------	---------------	--------------------------



<b>Node.js</b>	Alta	Moderada	Alta	Excelente (con React, Vue, etc.)
<b>Django</b>	Alta	Moderada	Alta	Buena (REST Framework, GraphQL)
<b>Spring Boot</b>	Media	Alta	Muy alta	Excelente (con Angular, React)

*Nota,* Esta tabla muestra una comparación entre las diferentes frameworks backend más usados y recomendados del mundo. Fuente tomada de (Singh, 2023).

## 10.2 Para el desarrollo del objetivo específico 2: “Definir los requisitos de la aplicación web, considerando las necesidades de los estudiantes de ingeniería de sistemas para la presentación de sus proyectos definiendo una arquitectura por capas para su implementación.”

Los requisitos funcionales y no funcionales son fundamentales para garantizar el éxito de una aplicación. Los funcionales definen qué debe hacer el sistema, permitiendo a los usuarios crear portafolios digitales interactivos, gestionar su visibilidad y recibir retroalimentación, lo cual es esencial para su propósito educativo y profesional. Por otro lado, los no funcionales aseguran que la experiencia de uso sea eficiente, segura y accesible, lo cual mejora la satisfacción del usuario final. La identificación y documentación clara de estos requisitos permite al equipo de desarrollo establecer prioridades, diseñar una arquitectura robusta y reducir riesgos durante la implementación. Además, estos requisitos sirven como base para validar el producto final y garantizar que cumpla con los objetivos del proyecto. Su correcta gestión asegura que la aplicación sea mantenible, escalable y confiable en el tiempo, alineándose con las necesidades tanto de los estudiantes como del sector educativo y productivo.



### 10.2.1 Requisitos Funcionales

- A. Los usuarios deben poder registrarse e iniciar sesión en la plataforma.
- B. Los estudiantes podrán crear y personalizar sus portafolios digitales.
- C. Se podrá incluir imágenes, videos, documentos y enlaces interactivos.
- D. El usuario podrá configurar su portafolio como público o privado.
- E. Docentes y empresas podrán dejar opiniones o sugerencias.

### 10.2.2 Requisitos No Funcionales

- A. La interfaz debe ser intuitiva y fácil de usar.
- B. El sistema debe proteger los datos mediante cifrado y protocolos seguros.
- C. La aplicación debe ser rápida y eficiente, con tiempos de carga bajos.
- D. Debe funcionar correctamente en distintos navegadores y dispositivos.
- E. La plataforma debe permitir crecer en funcionalidades y usuarios sin perder rendimiento.

### 10.2.3 Definición de tecnología frontend

Next.js fue elegido como tecnología frontend para la aplicación *ExponTE* debido a su capacidad para ofrecer renderizado del lado del servidor (SSR), lo cual mejora el rendimiento y la optimización para motores de búsqueda (SEO), algo crucial para la visibilidad de los portafolios estudiantiles. Además, su sistema de enrutamiento simplifica la navegación entre secciones de la aplicación, y su compatibilidad con componentes reutilizables permite desarrollar de forma ágil. Next.js también cuenta con soporte integrado para desarrollo full-stack, facilitando la conexión eficiente con el backend y mejorando la experiencia del desarrollador. Next.js se posiciona como una solución ideal para interfaces atractivas y funcionales. (Stack Overflow, 2023).



#### 10.2.4 Definición de tecnología backend

Node.js con TypeScript fue seleccionado para el backend de *ExponTE* por su eficiencia, escalabilidad y robustez en el desarrollo de aplicaciones web modernas. Node.js permite manejar múltiples peticiones concurrentes gracias a su modelo asincrónico basado en eventos, lo cual es ideal para una plataforma que gestiona usuarios, portafolios y contenido multimedia en tiempo real. Al incorporar TypeScript, se obtiene un sistema de tipado fuerte que mejora la mantenibilidad del código, reduce errores en tiempo de desarrollo y facilita la colaboración entre desarrolladores. Esta combinación potencia la calidad del software y permite construir APIs limpias, organizadas y seguras. Para una aplicación centrada en la educación y la innovación tecnológica, esta elección garantiza una base sólida que puede crecer y adaptarse fácilmente a nuevas funcionalidades y demandas del entorno académico y empresarial. (JetBrains, 2023).

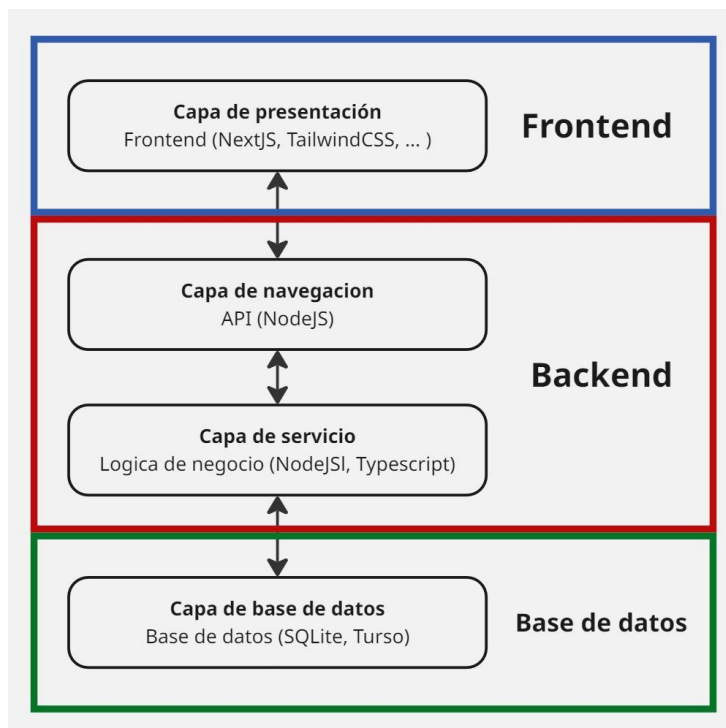
#### 10.2.5 Arquitectura por capas

La arquitectura por capas es un enfoque estructurado que divide una aplicación en módulos bien definidos, cada uno con responsabilidades específicas. Esta metodología mejora la mantenibilidad, escalabilidad y reutilización del código al separar claramente las funciones del sistema. En la ilustración siguiente se representa esta arquitectura aplicada, la cual se organiza en tres capas principales: Frontend, Backend y Base de datos. La capa de presentación o Frontend incluye tecnologías como Next.js y Tailwind CSS, enfocadas en la interfaz visual y la interacción con el usuario. El Backend se divide en dos niveles: una capa de navegación, donde se implementa una API con Node.js, y una capa de servicio, que contiene la lógica de negocio escrita en Node.js y TypeScript. Finalmente, en la capa de base de datos, se encuentran sistemas como SQLite o Turso, encargados del almacenamiento de la información.





**Ilustración 5, Arquitectura por capas**



Nota: La anterior ilustración se presenta la arquitectura por capas desarrollada para este informe, Fuente tomada del Autor.

### 10.3 Para el desarrollo del objetivo específico 3: “Crear una aplicación web con arquitectura por capas que integre experiencias inmersivas para la generación de portafolios interactivos, facilitando la visualización y comprensión de los proyectos desarrollados”.

Basándonos en la arquitectura por capas hecha en la **ilustración 5**, vamos a explicar el desarrollo de la aplicación empezando por la capa de presentación hasta la capa de base de datos:

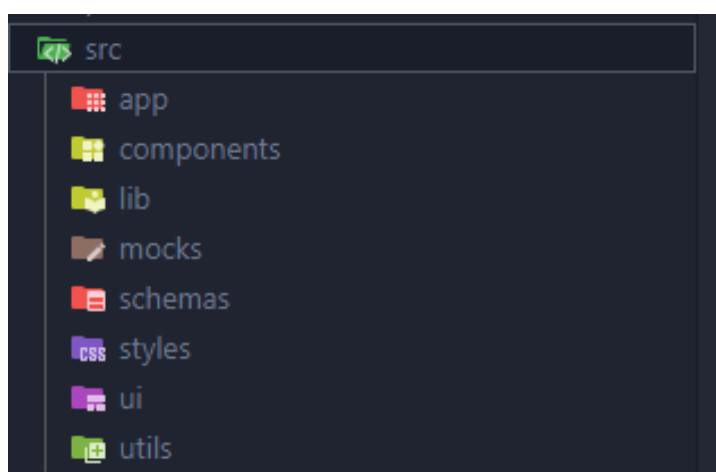




### 10.3.1 *Capa de Presentación:*

Para el desarrollo de la capa de presentación se usó el framework frontend NextJS en la cual se definió con una estructura de carpetas con las siguientes definiciones que se explicaran más adelante:

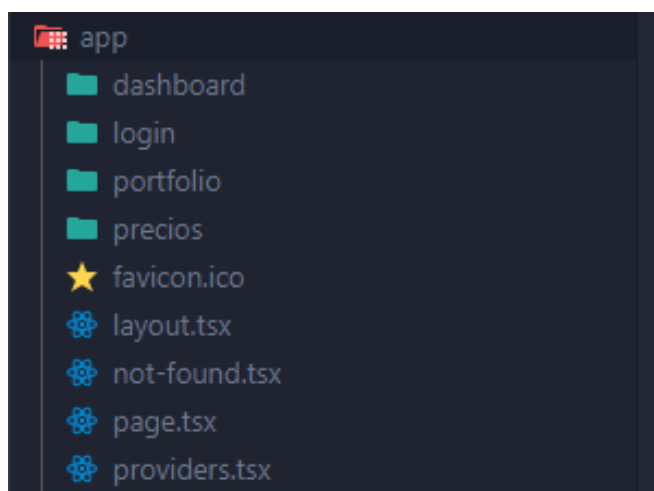
**Ilustración 6, Estructura de carpetas en NextJS**



Nota: La anterior ilustración se presenta la estructura de carpetas en NextJS que representa la capa de presentación del proyecto, Fuente tomada del Autor.

**APP:** en la carpeta “app” se define todas las rutas de la aplicación, como sus direcciones y además las rutas a donde debe ir sino se encuentra la ruta:

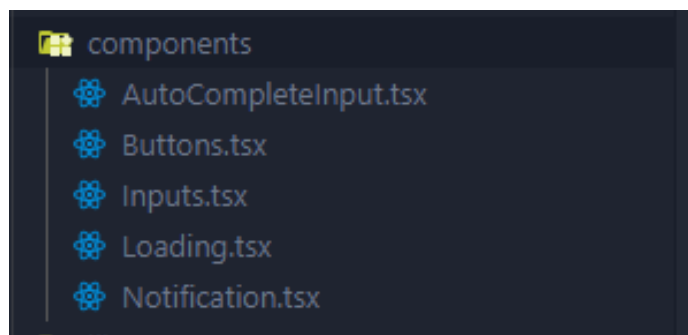
**Ilustración 7, Carpeta APP en NextJS**



Nota: La anterior ilustración se presenta la carpeta APP donde se definen las rutas del proyecto, Fuente tomada del Autor.

**COMPONENTS:** la carpeta “components” define los archivos de código que son reutilizables, esto es beneficioso para que un proyecto sea escalable. Como por ejemplo los botones, inputs, notificaciones, Etc.

**Ilustración 8,** *Carpeta COMPONENTS en NextJS*

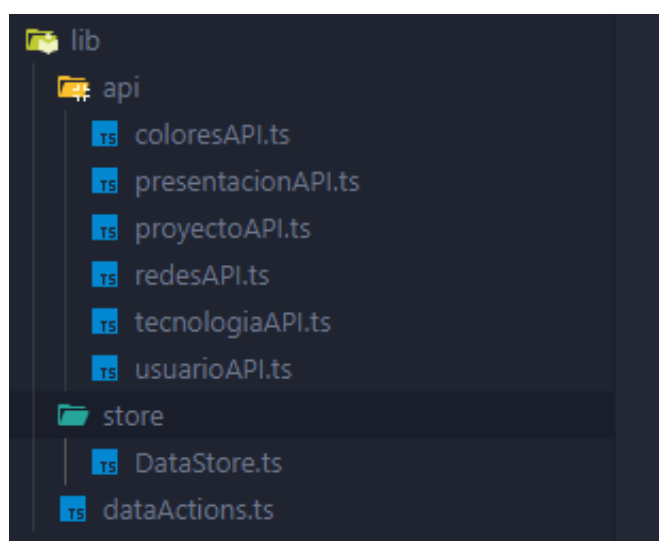


Nota: La anterior ilustración se presenta la carpeta COMPONENTS donde se encuentran los componentes reutilizables del proyecto, Fuente tomada del Autor.



**LIB:** la carpeta “lib” es un conjunto de funciones y recursos que se utilizan en el proyecto para realizar tareas comunes o complejas sin tener que reescribir el código desde cero, como en este caso: funciones de consultas a la base de datos, manejo de estados globales y manipulación de datos.

**Ilustración 9,** *Carpeta LIB en NextJS*



Nota: La anterior ilustración se presenta la carpeta LIB donde se encuentran las funciones y recursos que necesita el proyecto, Fuente tomada del Autor.

**SCHEMAS:** la carpeta de “schemas” se definen las interfaces del proyecto sobre todos los datos, estados globales y variables del proyecto en general:

**Ilustración 10,** *Esquemas del proyecto en NextJS*



```
1 export interface DataPresentacion {
2   id: number;
3   foto: string;
4   nombre: string;
5   titulos: string;
6 }
7 export interface Tecnologia {
8   id: number;
9   nombre_tecnologia: string;
10 }
11
12 export interface DataProyect {
13   id: number;
14   imagen: string;
15   titulo: string;
16   descripcion: string;
17   tecnologias: Tecnologia[];
18   linkGithub: string;
19   linkDemo: string;
20 }
21
22 export interface Colores {
23   id: number;
24   textoColor: string;
25   textoTituloColor: string;
26   navegacionColor: string;
27   presentacionColor: string;
28   proyectosColor: string;
29   redesColor: string;
30 }
31
32
33 export type RedSocial = "linkedin" | "github" | "gmail" | "instagram" | "facebook" | "youtube";
34 export type RedesState = { id: number; social: RedSocial; activo: boolean; usuario: string }[];
```

Nota: La anterior ilustración se presenta el archivo de “schemas” que definen los tipos de datos de cada objeto en el proyecto, Fuente tomada del Autor.

**UI:** esta carpeta se encuentran los componentes de “UI” = “User Interface” (Interfaz de Usuario) del proyecto. Como por ejemplo componente de visualizar los proyectos, de navegación y más.

### Ilustración 11, Carpeta UI en NextJS

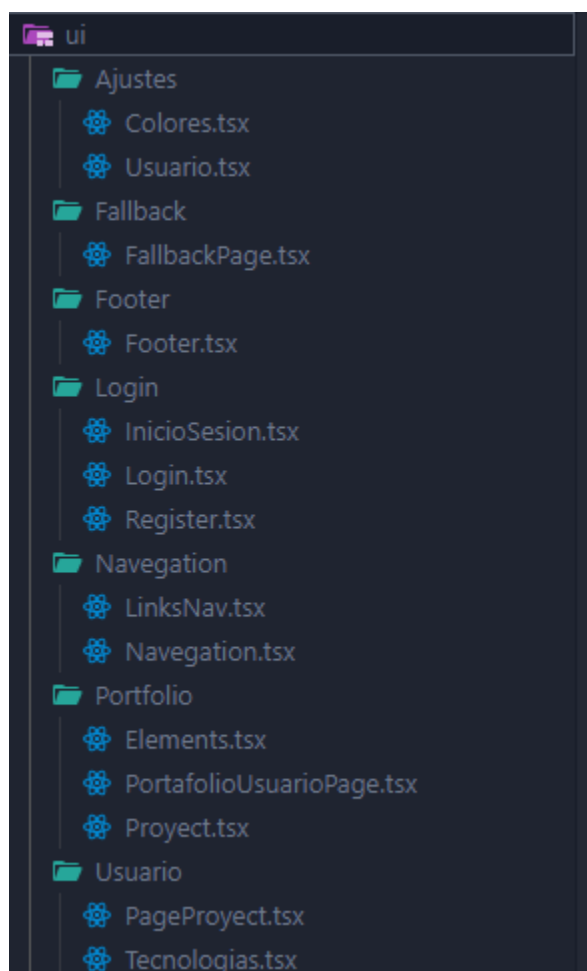
*“Formando líderes para la construcción de un nuevo país en paz”*



SC-CER96940



Universidad de Pamplona  
Pamplona - Norte de Santander - Colombia  
Tel: (+57) 3152429405 - 3160244475



Nota: La anterior ilustración se presenta la carpeta de “UI” donde se encuentra los componentes de interfaz de usuario del proyecto, Fuente tomada del Autor.

### 10.3.2 Presentación de interfaces de usuario:

Las siguientes ilustraciones hacen referencia a las pestañas de interfaz de usuario del proyecto:

La ilustración 12 muestra la pantalla principal del proyecto, desde la cual es posible acceder al sistema y visualizar opciones como los precios disponibles. Esta interfaz ha sido diseñada para ofrecer una experiencia intuitiva al usuario, permitiendo una navegación fluida



entre las diferentes secciones del sistema. Desde este punto de inicio, los usuarios pueden ingresar a las funcionalidades principales y explorar la información relevante de manera rápida y eficiente

**Ilustración 12, Pantalla principal de la aplicación web**



Nota: La anterior ilustración se presenta la pantalla principal del proyecto donde se puede acceder al sistema como los precios, Fuente tomada del Autor.

La ilustración 13 muestra la pantalla de detalles del proyecto, donde se presenta una descripción clara de sus principales características. Esta vista permite al usuario comprender el propósito y los elementos más relevantes del sistema desarrollado.

**Ilustración 13, Pantalla de detalles de la aplicación web**



Nota: La anterior ilustración se presenta la pantalla de detalles sobre el proyecto, Fuente tomada del Autor.

La ilustración 14 muestra la pantalla de precios, la cual aún se encuentra en desarrollo. Actualmente solo está disponible el plan básico, mientras que el plan premium se habilitará tras la evaluación del impacto del proyecto

**Ilustración 14, Pantalla de precios**

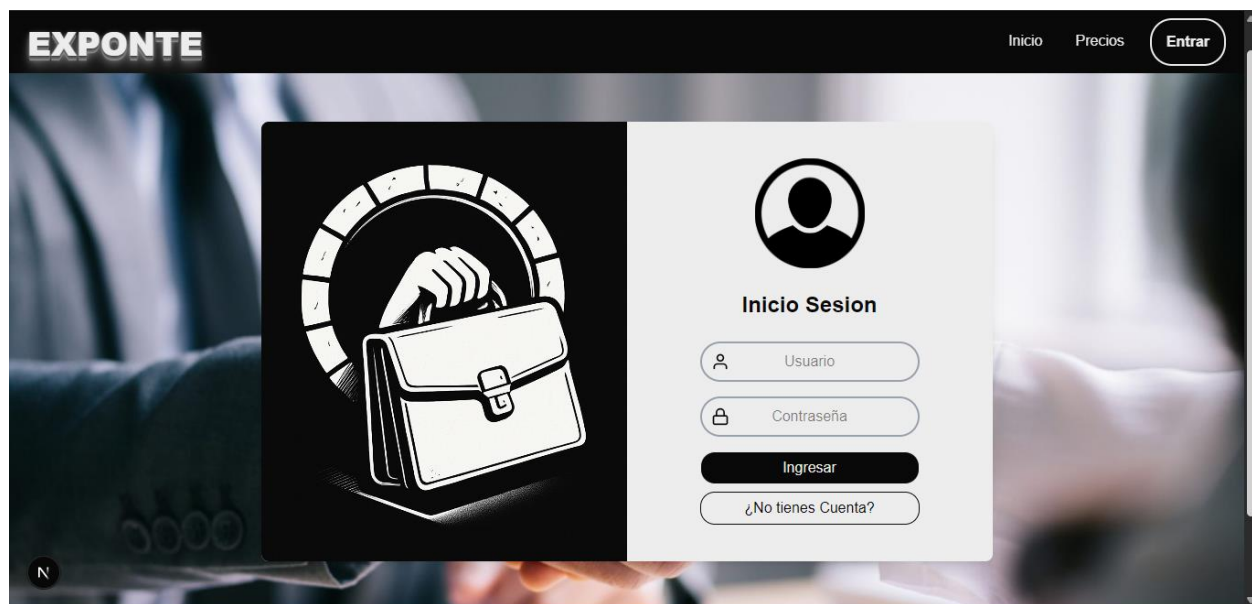




Nota: La anterior ilustración se presenta la pantalla de precios que aún se encuentra en desarrollo, Fuente tomada del Autor.

La ilustración 15 muestra la pantalla de inicio de sesión del sistema. Desde aquí, el usuario puede acceder de forma segura a sus datos personales y funcionalidades asociadas.

**Ilustración 15, Pantalla de login**



Nota: La anterior ilustración se presenta la pantalla de login, Fuente tomada del Autor.

La ilustración 16 muestra la pantalla de registro del sistema. En esta sección, el usuario puede crear una cuenta para acceder a las funcionalidades disponibles

**Ilustración 16, Pantalla de registro de usuario**





Nota: La anterior ilustración se presenta la pantalla de registro, Fuente tomada del Autor.

La ilustración 17 muestra la pantalla del panel principal (**dashboard**) del sistema. Desde esta vista, el usuario puede modificar la información de su portafolio y acceder a otras configuraciones disponibles.

**Ilustración 17, Pantalla del dashboard**

Nota: La anterior ilustración se presenta la pantalla de dashboard, Fuente tomada del Autor.

La ilustración 18 muestra la pantalla del panel principal (dashboard), específicamente en la sección de presentación. En esta área, el usuario puede subir su información inicial para completar su portafolio personal.

**Ilustración 18, Sección de presentación**



Seccion de Presentacion

https://avatars.githubusercontent.com/u/143953366?v=4

Hola, Soy Juan Pablo Marquez Sanchez

Un desarrollador de software

Guardar Cambios

Nota: La anterior ilustración se presenta la pantalla de dashboard en la sección de presentación, Fuente tomada del Autor.

La ilustración 19 muestra la pantalla del panel principal (**dashboard**), en la sección de proyectos. En esta sección, el usuario puede subir sus proyectos, registrar las tecnologías utilizadas y añadir información adicional relevante.

**Ilustración 19, Sección de proyectos**

Seccion de Proyectos

Proyecto 1

https://juanpamarquez.github.io/portfolio/images-projects/libro-click.webp

Libro-Click 2

Libro-Click es una plataforma desarrollada con Next.js en el frontend y Node.js en el backend, con el obje

https://github.com/JuanPaMarquez/libro-click-frontend/

https://libro-click-frontend.vercel.app/

nextjs

typescript

tailwindcss

nodejs

sqlite

+

Actualizar Proyecto

Añadir Proyecto



Nota: La anterior ilustración se presenta la pantalla de dashboard en la sección de proyectos, Fuente tomada del Autor.

La ilustración 20 muestra la pantalla del panel principal (**dashboard**), en la sección de redes sociales. Aquí, el usuario puede agregar enlaces a sus perfiles sociales y activar o desactivar su visibilidad según prefiera.

**Ilustración 20, Sección de redes sociales**

Seccion de Redes

LINKEDIN

☒

<https://www.linkedin.com/in/juanp-marquez>

GITHUB

☒

<https://github.com/JuanPaMarquez>

GMAIL

☒

[juan013sanchez@gmail.com](mailto:juan013sanchez@gmail.com)

INSTAGRAM

☐

Usuario de instagram

FACEBOOK

☐

<http://facebook.com/usuario>

YOUTUBE

☐

Usuario de youtube

Guardar Cambios

Nota: La anterior ilustración se presenta la pantalla de dashboard en la sección de redes sociales, Fuente tomada del Autor.



La ilustración 21 muestra la pantalla del panel principal (**dashboard**), en la sección de colores. Desde aquí, el usuario puede personalizar la paleta de colores de su portafolio según sus preferencias.

**Ilustración 21, Sección de colores**

**Colores:**

Color del texto

Color del título

Fondo de navegación

Fondo de presentación

Fondo de proyectos

Fondo de redes

**Guardar Cambios**

Nota: La anterior ilustración se presenta la pantalla de dashboard en la sección de colores, Fuente tomada del Autor.

La ilustración 22 muestra la pantalla del panel principal (**dashboard**), en la sección del usuario. Aquí, el usuario puede cambiar su nombre de usuario para acceder a su portafolio mediante el enlace proporcionado, así como actualizar su contraseña.

**Ilustración 22, Sección de usuario**



## Usuario:

Nombre de usuario:

Actualizar Nombre de Usuario

Contraseña:

Repetir Contraseña:

Mostrar contraseña

Cambiar Contraseña

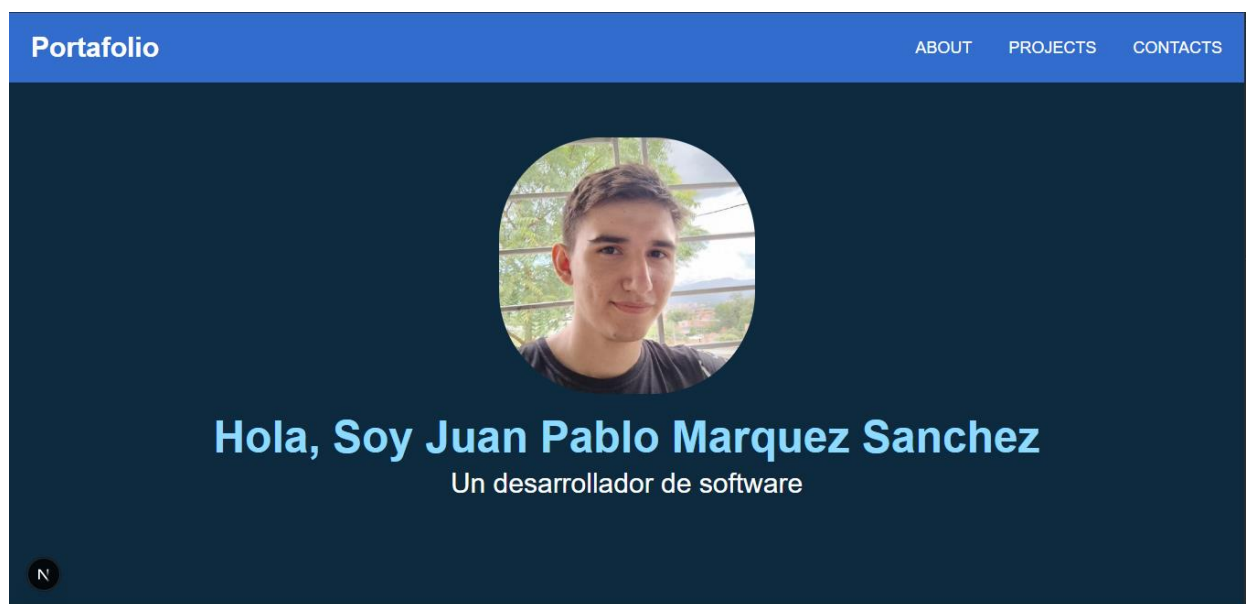
Nota: La anterior ilustración se presenta la pantalla de dashboard en la sección del usuario,  
Fuente tomada del Autor.

### 10.3.3 Presentación del portafolio generado

En las siguientes ilustraciones muestran las pantallas del portafolio generado por la aplicación e información proporcionada por el usuario.

**Ilustración 23, Sección de presentación generada**

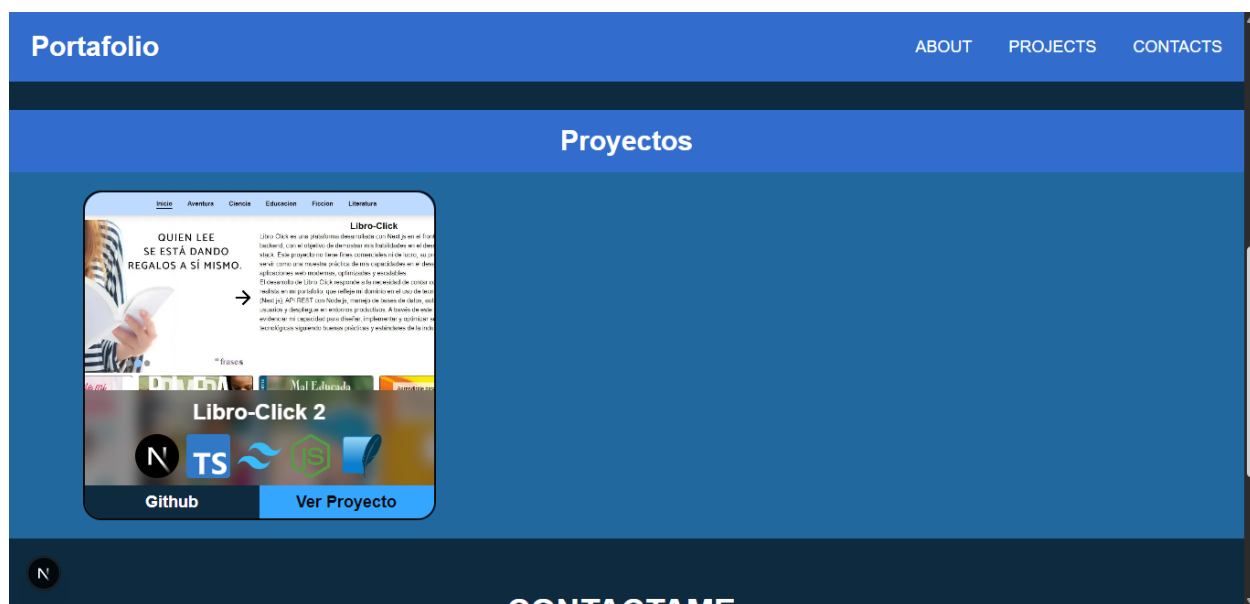




Nota: La anterior ilustración se presenta la pantalla del portafolio generado por la aplicación en la sección de presentación, Fuente tomada del Autor.

La ilustración 24 muestra la pantalla del portafolio generado por la aplicación, en la sección de proyectos proporcionados por el usuario. Desde aquí, se puede acceder al código en GitHub y, si está disponible, también a una demo del proyecto.

**Ilustración 24,** *Sección de proyectos generada*



Nota: La anterior ilustración se presenta la pantalla del portafolio generado por la aplicación en la sección de proyectos, Fuente tomada del Autor.

La ilustración 25 muestra la pantalla del portafolio generado por la aplicación, en la sección de proyectos. En esta vista, el usuario puede visualizar la descripción, imagen y tecnologías asociadas a un proyecto abierto.

**Ilustración 25,** *Sección de proyectos en un proyecto abierto*



Nota: La anterior ilustración se presenta la pantalla del portafolio generado por la aplicación en la sección de proyectos donde el usuario abrió un proyecto, Fuente tomada del Autor.

La ilustración 26 muestra la pantalla del portafolio generado por la aplicación, en la sección de redes sociales. En esta área se visualizan las redes sociales que el usuario ha habilitado.

**Ilustración 26,** *Sección de redes sociales*



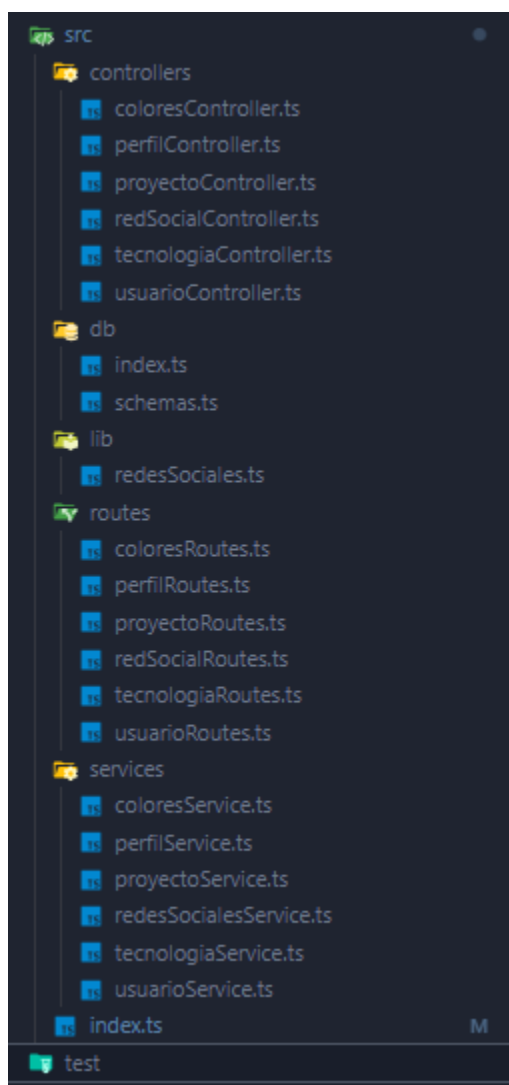
Nota: La anterior ilustración se presenta la pantalla del portafolio generado por la aplicación en la sección de redes sociales, Fuente tomada del Autor.

#### 10.3.4 Backend:

El backend del proyecto tiene una estructura de carpetas siguiendo una arquitectura MVC (modelo vista controlado):

La ilustración 27 muestra la estructura de carpetas del backend de la aplicación, dividida en diferentes módulos: rutas (**routes**), que permiten la comunicación directa con el frontend; esquemas (**db/schemas**), que definen los modelos de cada entidad; controladores (**controllers**), que contienen la lógica de procesamiento y envío de datos; y servicios (**services**), que manejan la interacción con la base de datos. Además, incluye las carpetas de testing y otros archivos complementarios necesarios para el funcionamiento del sistema.

**Ilustración 27, Estructura MVC en el backend**



Nota: La anterior ilustración se presenta la estructura de carpetas del backend de la aplicación,  
Fuente tomada del Autor.



### 10.3.5 Capa de Navegación

Esta capa se desarrolla en el backend usando NodeJS y Typescript para mayor control de los datos en las siguientes ilustraciones se presentarán las rutas de navegación que hace el backend para que el frontend pueda comunicarse correctamente.

**Ilustración 28, Configuración de rutas y CORS del backend**

```

1  import express from 'express';
2  import usuarioRouter from './routes/usuarioRoutes';
3  import perfilRouter from './routes/perfilRoutes';
4  import proyectoRouter from './routes/proyectoRoutes';
5  import tecnologiaRouter from './routes/tecnologiaRoutes';
6  import redSocialRoutes from './routes/redSocialRoutes';
7  import coloresRouter from './routes/coloresRoutes';
8
9  const app = express();
10 const cors = require('cors');
11 const PORT = 3001;
12
13 const corsOptions = {
14   // origin: 'https://exponte.vercel.app',
15   origin: 'http://localhost:3000',
16   methods: 'GET,POST,PUT,DELETE',
17   allowedHeaders: 'Content-Type'
18 };
19
20 app.use(cors(corsOptions));
21
22 app.use(express.json());
23
24 app.use('/api', usuarioRouter);
25 app.use('/api', perfilRouter);
26 app.use('/api', proyectoRouter);
27 app.use('/api', tecnologiaRouter);
28 app.use('/api', redSocialRoutes);
29 app.use('/api', coloresRouter);
30
31 app.listen(PORT, () => {
32   console.log(`Servidor corriendo en http://localhost:${PORT}`);
33 });

```



Nota: La anterior ilustración se presenta la configuración del API en la definición de las rutas como de los CORS, Fuente tomada del Autor.

La ilustración 29 muestra la definición de la navegación relacionada con la entidad usuario. Esta estructura define cómo se gestionan y organizan las rutas y acciones asociadas a los usuarios en la aplicación.

**Ilustración 29,** *Definición de las rutas del usuario*

```

1  import { Router } from "express";
2  import {
3    crearUsuario,
4    obtenerUsuarioPorId,
5    verificarSesionUsuario,
6    modificarNombreUsuario,
7    modificarPasswordUsuario,
8    obtenerUsuarioPorNombre
9  } from "../controllers/usuarioController";
10
11  const usuarioRouter = Router();
12
13  usuarioRouter.post("/usuario", crearUsuario);
14  usuarioRouter.get("/usuario/:id", obtenerUsuarioPorId);
15  usuarioRouter.post("/usuario/verificar", verificarSesionUsuario);
16  usuarioRouter.put("/usuario/nombre/:id", modificarNombreUsuario);
17  usuarioRouter.put("/usuario/password/:id", modificarPasswordUsuario);
18  usuarioRouter.get("/usuario/nombre/:nombre_usuario", obtenerUsuarioPorNombre);
19
20  export default usuarioRouter;

```

Nota: La anterior ilustración se define la navegación en la entidad usuario, Fuente tomada del Autor.

**Ilustración 30,** *Definición de las rutas de perfiles*





```
You, la semana pasada | 1 author (You)
1  import { Router } from "express";
2  import {
3      crearPerfil,
4      obtenerPerfilPorUsuarioId,
5      modificarPerfil
6  } from "../controllers/perfilController";
7
8  const perfilRouter = Router();
9
10 perfilRouter.post("/perfil", crearPerfil);
11 perfilRouter.get("/perfil/:usuario_id", obtenerPerfilPorUsuarioId);
12 perfilRouter.put("/perfil/:id", modificarPerfil);
13
14 export default perfilRouter;
```

Nota: La anterior ilustración se presenta la definición de navegación con la sección perfil,  
Fuente tomada del Autor.

### Ilustración 31, Definición de las rutas de proyectos

```
You, la semana pasada | 1 author (You)
1  import { Router } from "express";
2  import {
3      crearProyectos,
4      obtenerProyectosPorUsuario,
5      modificarProyectos,
6      eliminarProyectos
7  } from "../controllers/proyectoController";
8
9  const proyectoRouter = Router();
10
11 proyectoRouter.post("/proyecto", crearProyectos);
12 proyectoRouter.get("/proyecto/:usuario_id", obtenerProyectosPorUsuario);
13 proyectoRouter.put("/proyecto/:id", modificarProyectos);
14 proyectoRouter.delete("/proyecto/:id", eliminarProyectos);
15
16 export default proyectoRouter;
```

Nota: La anterior ilustración se presenta la definición de navegación con la sección proyectos,  
Fuente tomada del Autor.



**Ilustración 32,** Definición de las rutas de redes sociales

```

You, hace 3 días | 1 author (You)
1  import { Router } from 'express';
2  import {
3      crearRedesSociales,
4      obtenerRedesSocialesPorUsuario,
5      modificarRedSocial
6  }
7  } from '../controllers/redSocialController';
8
9  const redSocialRoutes = Router();
10
11  redSocialRoutes.post('/redes-sociales', crearRedesSociales);
12  redSocialRoutes.get('/redes-sociales/:usuarioId', obtenerRedesSocialesPorUsuario);
13  redSocialRoutes.put('/redes-sociales', modificarRedSocial);
14
15  export default redSocialRoutes;
You, la semana pasada * Agregar funcionalidad

```

Nota: La anterior ilustración se presenta la definición de navegación con la sección redes sociales, Fuente tomada del Autor.

**Ilustración 33,** Definición de las rutas de tecnologías

```

You, hace 4 días | 1 author (You)
1  import { Router } from 'express';
2  import {
3      eliminarTecnologia,
4      agregarTecnologia
5  } from '../controllers/tecnologiaController';
6
7  const tecnologiaRouter = Router();
8
9  tecnologiaRouter.post('/tecnologias', agregarTecnologia);
10  tecnologiaRouter.delete('/tecnologias/:id', eliminarTecnologia);
11
12  export default tecnologiaRouter;

```

Nota: La anterior ilustración se presenta la definición de navegación con el manejo de tecnologías, Fuente tomada del Autor.



**Ilustración 34,** Definición de las rutas de los colores

```

You, anteayer | 1 author (You)
1  import { Router } from 'express';
2  import {
3      crearColores,
4      obtenerColoresPorUsuario,
5      modificarColores
6  } from '../controllers/coloresController';
7
8  const coloresRouter = Router();
9
10 // coloresRouter.post('/colores', crearColores);
11 coloresRouter.get('/colores/:usuario_id', obtenerColoresPorUsuario);
12 coloresRouter.put('/colores', modificarColores);
13
14 export default coloresRouter;
15

```

Nota: La anterior ilustración se presenta la definición de navegación con el manejo de los colores, Fuente tomada del Autor.

### 10.3.6 Capa de Base de Datos

En esta sección vamos a presentar la construcción de la capa de base de datos manejada en este proyecto usan SQLite y Turso.tech para la administración de datos en la nube. La API se encarga de la comunicación del frontend con la base de datos mediante Drizzel-orm para se debe definir la estructura de la base de datos en el backend para que la administre de forma correcta como se ve en la siguiente ilustración:

**Ilustración 35,** Estructura de la base de dato en la API



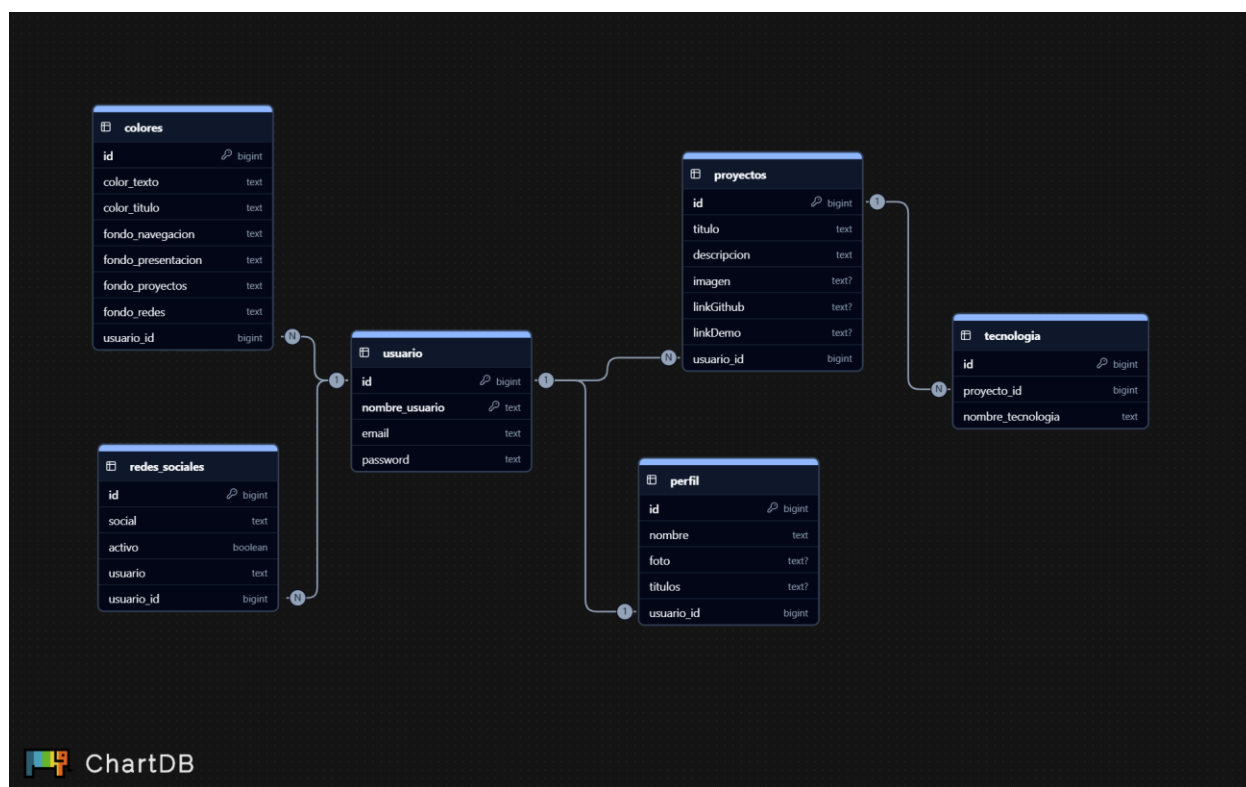
```
1 import { integer, text, sqliteTable, numeric } from 'drizzle-orm/sqlite-core';
2
3 // Tabla usuario
4 export const usuario = sqliteTable('usuario', {
5   id: integer('id').primaryKey({ autoIncrement: true }),
6   nombre_usuario: text('nombre_usuario').notNull().unique(),
7   email: text('email').notNull(),
8   password: text('password').notNull(),
9 });
10
11 // Tabla perfil
12 export const perfil = sqliteTable('perfil', {
13   id: integer('id').primaryKey({ autoIncrement: true }),
14   nombre: text('nombre').notNull(),
15   foto: text('foto'),
16   titulos: text('titulos'),
17   usuario_id: integer('usuario_id').notNull().references(() => usuario.id),
18 });
19
20 // Tabla proyectos
21 export const proyectos = sqliteTable('proyectos', {
22   id: integer('id').primaryKey({ autoIncrement: true }),
23   titulo: text('titulo').notNull(),
24   descripcion: text('descripcion').notNull(),
25   imagen: text('imagen'),
26   linkGithub: text('linkGithub'),
27   linkDemo: text('linkDemo'),
28   usuario_id: integer('usuario_id').notNull().references(() => usuario.id),
29 });
30
31 // Tabla redes sociales
32 export const redes_sociales = sqliteTable('redes_sociales', {
33   id: integer('id').primaryKey({ autoIncrement: true }),
34   social: text('social').notNull(),
35   activo: numeric('activo').notNull(), // 1 para activo, 0 para inactivo
36   usuario: text('usuario'), // este parece opcional, si quieres lo puedes quitar
37   usuario_id: integer('usuario_id').notNull().references(() => usuario.id),
38 });
39
40 // Tabla tecnologia
41 export const tecnologia = sqliteTable('tecnologia', {
42   id: integer('id').primaryKey({ autoIncrement: true }),
43   proyecto_id: integer('proyecto_id').notNull().references(() => proyectos.id, { onDelete: 'cascade' }),
44   nombre_tecnologia: text('nombre_tecnologia').notNull(),
45 });
46
47 export const colores = sqliteTable('colores', {
48   id: integer('id').primaryKey({ autoIncrement: true }),
49   color_texto: text('color_texto').notNull(),
50   color_titulo: text('color_titulo').notNull(),
51   fondo_navegacion: text('fondo_navegacion').notNull(),
52   fondo_presentacion: text('fondo_presentacion').notNull(),
53   fondo_proyectos: text('fondo_proyectos').notNull(),
54   fondo_redes: text('fondo_redes').notNull(),
55   usuario_id: integer('usuario_id').notNull().references(() => usuario.id),
56 });
```



Nota: La anterior ilustración se presenta la definición de la base de datos en NodeJS, Fuente tomada del Autor.

La ilustración 36 muestra el diseño de la base de datos, realizado con la herramienta ChartDB. Este diagrama refleja la estructura y relaciones entre las distintas tablas que conforman el sistema.

**Ilustración 36,** *Diagrama de la composición de la base de datos*



Nota: La anterior ilustración muestra la base de datos usando la herramienta ChartDB, Fuente tomada del Autor en (ChartDB, 2025).





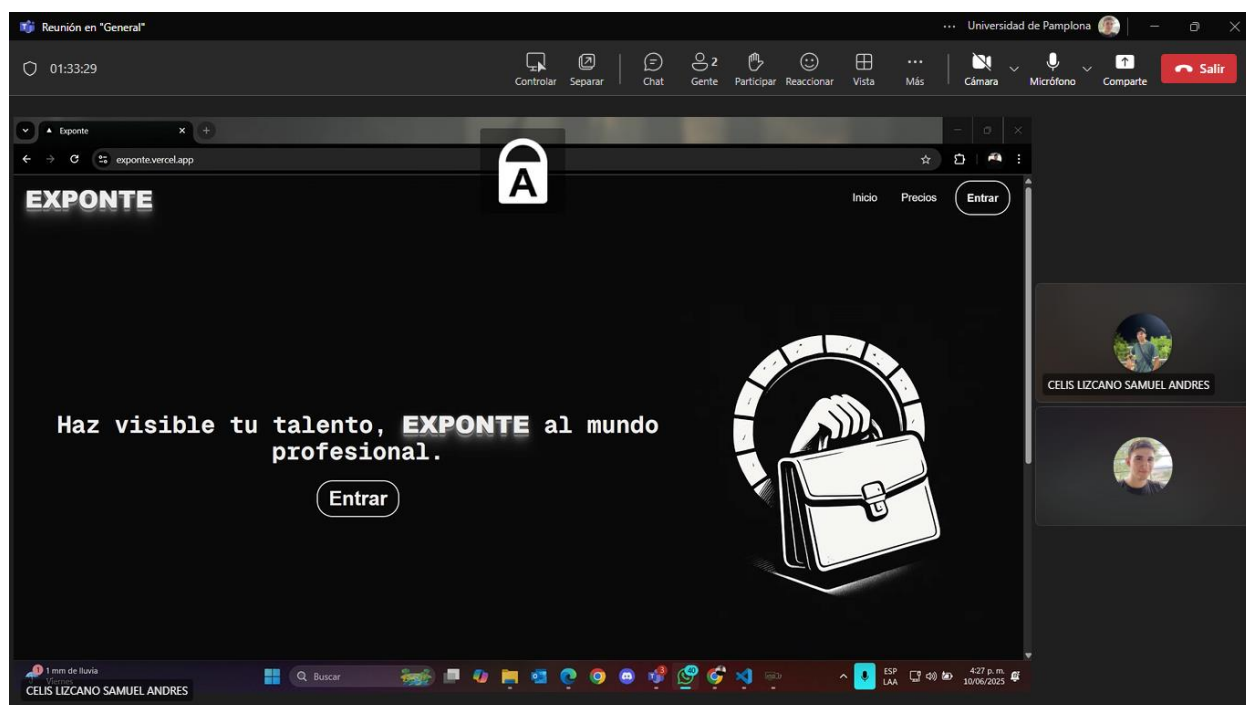
#### **10.4 Para el desarrollo del objetivo específico 4: “Validar la efectividad de la aplicación web mediante pruebas con estudiantes, evaluando su impacto en la presentación de los proyectos académicos.”**

Para validar la efectividad de la aplicación web desarrollada, se diseñó una estrategia de evaluación centrada en pruebas con usuarios reales, en este caso, estudiantes que estuvieran desarrollando o presentando proyectos académicos. El propósito fue analizar cómo la herramienta influye en la manera en que los estudiantes estructuran, organizan y presentan sus trabajos. Se establecieron criterios clave para guiar la evaluación, como la facilidad de uso, la claridad en la presentación, el tiempo invertido y la percepción de mejora en comparación con métodos tradicionales. Esta etapa es crucial, ya que permite obtener evidencia empírica sobre el impacto real del sistema en un entorno académico.

##### **10.4.1 Pruebas de interfaz:**

La ilustración 37 evidencia el desarrollo de una de las pruebas de usuario realizadas con un estudiante del programa de Ingeniería de Sistemas. Durante esta sesión, el estudiante interactuó directamente con la aplicación web, explorando sus funcionalidades mientras realizaba la presentación de un proyecto académico. Esta actividad permitió observar en tiempo real aspectos como la facilidad de navegación, la comprensión de la interfaz y la capacidad del sistema para adaptarse a las necesidades del usuario. La información recolectada en esta prueba contribuyó significativamente al análisis de la usabilidad y efectividad de la herramienta.

#### **Ilustración 37, Prueba de la aplicación con un estudiante**

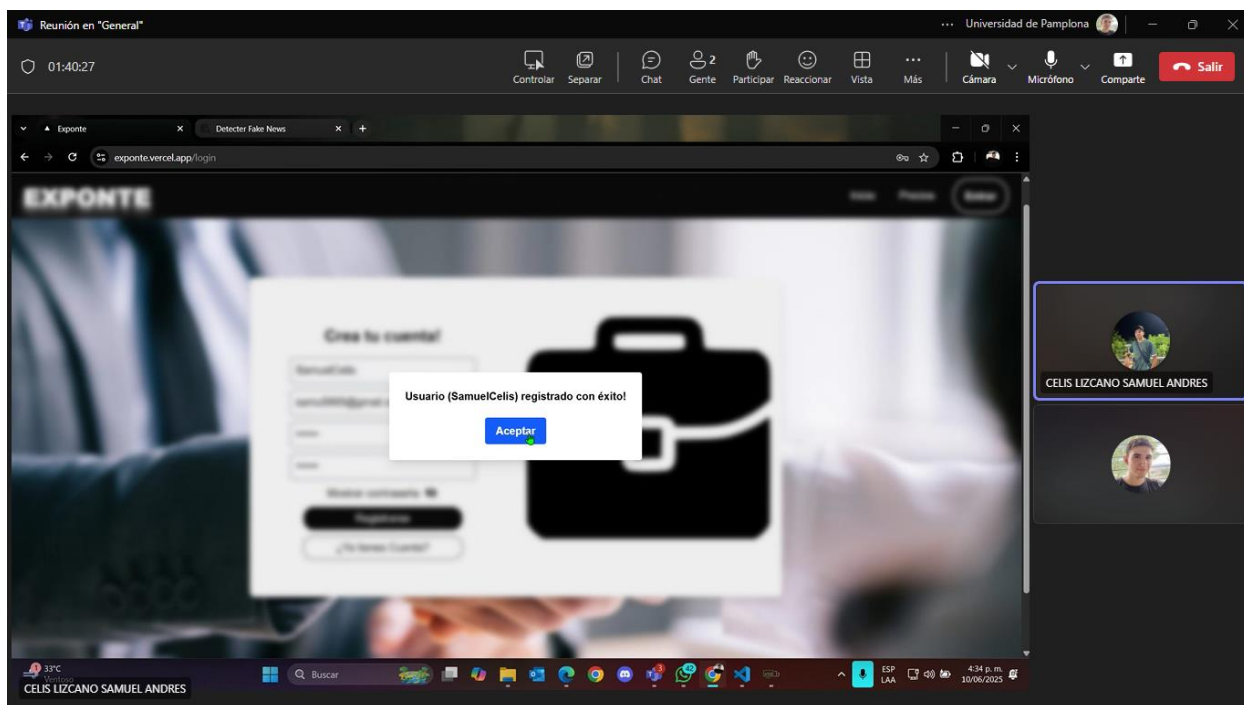


Nota: La anterior ilustración muestra las pruebas de usuario con un estudiante de ingeniería de sistema, Fuente tomada del Autor.

La ilustración 38 presentada muestra el proceso de prueba relacionado con el registro de usuario dentro de la aplicación web, realizado por un estudiante del programa de Ingeniería de Sistemas. Durante esta fase, se evaluó la funcionalidad del formulario de registro, la validación de datos ingresados y la experiencia general del usuario al momento de crear una cuenta en el sistema. Esta prueba permitió identificar aspectos clave como la claridad de los campos, la retroalimentación ante errores y la eficiencia del proceso. Los resultados obtenidos sirvieron como insumo para ajustar detalles técnicos y mejorar la usabilidad.

**Ilustración 38,** *Prueba del registro de usuario con un estudiante*

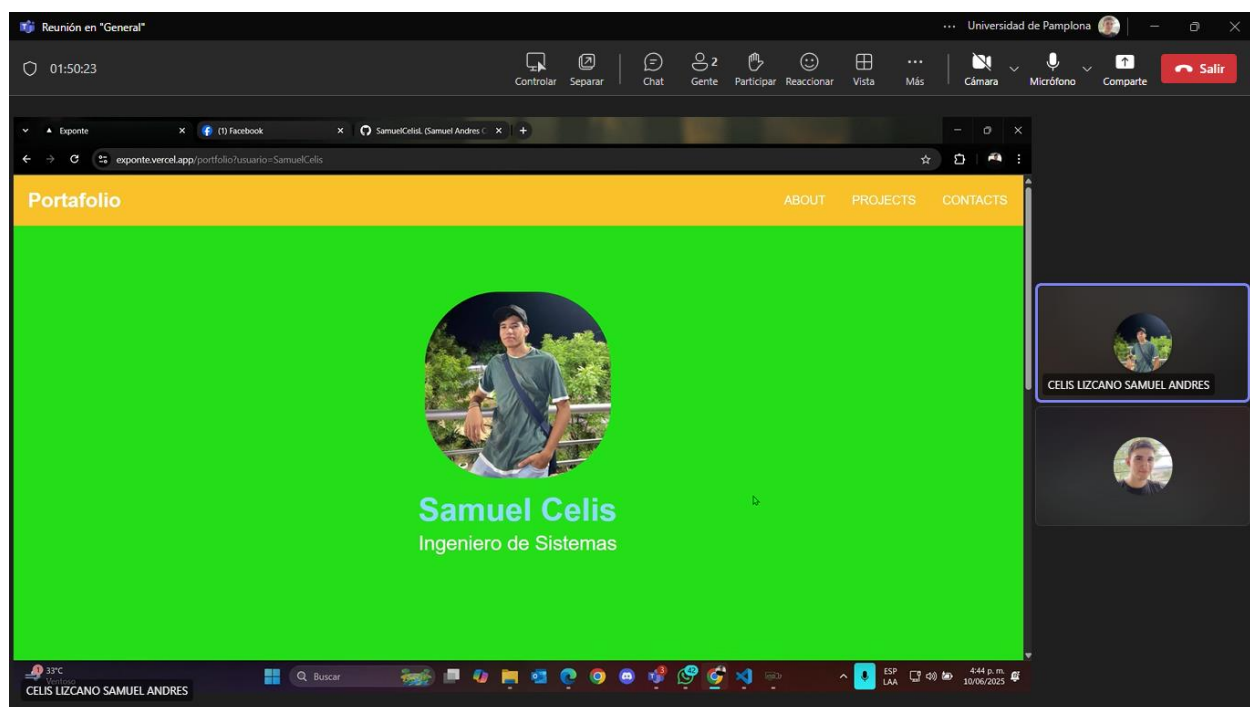




Nota: La anterior ilustración muestra las pruebas de registro de usuario con un estudiante de ingeniería de sistema, Fuente tomada del Autor.

La ilustración 39 corresponde a la fase de pruebas en la que un estudiante del programa de Ingeniería de Sistemas generó su portafolio académico utilizando la aplicación web. En esta etapa se evaluó la funcionalidad del sistema para compilar, organizar y presentar los elementos del portafolio de manera clara y estructurada. Se analizaron aspectos como la disposición del contenido, la facilidad para seleccionar los proyectos a incluir y la calidad del resultado final. Esta prueba permitió validar que la herramienta cumple con su propósito de facilitar la presentación profesional de trabajos académicos.

**Ilustración 39, Prueba de generación de portafolio con un estudiante**



Nota: La anterior ilustración muestra las pruebas de generación del portafolio de un estudiante de ingeniería de sistema, Fuente tomada del Autor.

#### 10.4.2 Retroalimentación:

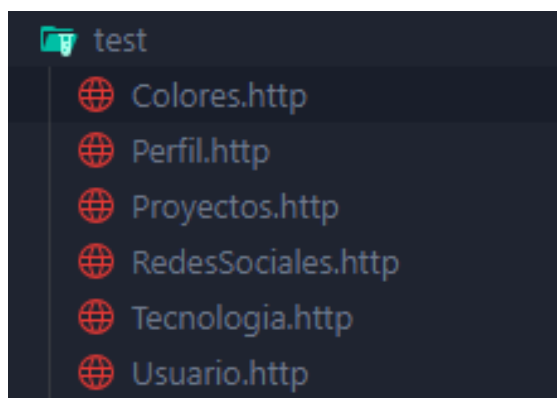
Durante las pruebas realizadas con los estudiantes, se recopilaban diversas observaciones y retroalimentaciones que resultaron valiosas para el proceso de mejora de la aplicación. Entre los comentarios más relevantes se destacaron aspectos como la claridad y adecuación de los nombres en los formularios, así como la falta de restricciones al ingresar espacios en campos donde no deberían permitirse. También se mencionaron sugerencias relacionadas con la practicidad de uso y elementos visuales de la interfaz. Estas observaciones permitieron identificar detalles técnicos y de diseño que fueron corregidos posteriormente, contribuyendo a una experiencia de usuario más fluida, intuitiva y profesional.



### 10.4.3 Pruebas de software:

La ilustración 40 muestra el proceso de prueba de la API, enfocándose en la interacción de cada entidad correspondiente a las tablas de la base de datos. Durante esta fase se verificó que las operaciones de consulta, creación, actualización y eliminación funcionaran correctamente en cada módulo del sistema. Asimismo, se validó la comunicación entre la capa de presentación y la capa de navegación, asegurando una integración efectiva entre el frontend y el backend. Estas pruebas permitieron identificar errores lógicos y asegurar que el flujo de datos entre las distintas capas del sistema fuera coherente y eficiente.

**Ilustración 40,** *Pruebas de la API en cada entidad de la base de datos*



Nota: La anterior ilustración muestra las pruebas de la API en cada entidad que corresponde a las tablas de la base de datos y cada comunicación de la capa de presentación con la capa de navegación, Fuente tomada del Autor.

La ilustración 41 corresponde a las pruebas realizadas sobre la API en la entidad "usuario", donde se evaluaron operaciones fundamentales como la creación, consulta, modificación y obtención de registros. Estas pruebas permitieron verificar que los endpoints respondieran correctamente ante diferentes solicitudes y que la lógica implementada cumpliera



con los requisitos funcionales establecidos. Además, se analizó el manejo de posibles errores, la validación de datos y la consistencia de las respuestas. Este proceso fue esencial para garantizar la estabilidad y fiabilidad del sistema en uno de sus módulos más críticos.

#### Ilustración 41, Pruebas de todas las consultas de la entidad Usuario

```

1  ### * Obtener usuario por ID
2
3  GET http://localhost:3001/api/usuario/1
4
5
6  ### * Crear un nuevo usuario
7  POST http://localhost:3001/api/usuario
8  Content-Type: application/json
9
10 {
11   "nombre_usuario": "Nohora",
12   "email": "nora@gmail.com",
13   "password": "123nora"
14 }
15
16 ### * Verificar el email y la contraseña de un usuario
17 POST http://localhost:3001/api/usuario/verificar
18 Content-Type: application/json
19
20 {
21   "email": "nora@gmail.com",
22   "password": "321nora"
23 }
24
25 ### * Actualizar nombre de usuario
26 PUT http://localhost:3001/api/usuario/nombre/1
27 Content-Type: application/json
28
29 {
30   "nombre_usuario": "Nohora1"
31 }
32
33 ### * Actualizar contraseña de usuario
34 PUT http://localhost:3001/api/usuario/password/1
35 Content-Type: application/json
36
37 {
38   "password": "321nora"
39 }
40
41 ### * Obtener usuario por nombre de usuario
42 GET http://localhost:3001/api/usuario/nombre/juanpablo

```

Nota: La anterior ilustración muestra las pruebas de la API en la entidad usuario con consultas como creación de usuario, obtención de un usuario, modificación de un usuario, etc. Fuente tomada del Autor.



**Ilustración 42,** *Pruebas de todas las consultas de la entidad Proyectos*

```

1  ### * Crear un proyectos
2  POST http://localhost:3001/api/proyecto
3  Content-Type: application/json
4
5  [
6    {
7      "titulo": "Proyecto de prueba 2",
8      "descripcion": "Descripción del proyecto de prueba 2",
9      "imagen": "http://ejemplo.com/imagen.jpg",
10     "linkGithub": "http://github.com/usuario/proyecto",
11     "linkDemo": "http://ejemplo.com/demo",
12     "usuario_id": 1
13   }
14 ]
15
16 ### * Obtener todos los proyectos de un usuario
17 GET http://localhost:3001/api/proyecto/1
18
19 ### * Modificar un proyectos
20 PUT http://localhost:3001/api/proyecto/1
21 Content-Type: application/json
22
23 [
24   {
25     "id": 3,
26     "titulo": "Proyecto de prueba 3 modificado",
27     "descripcion": "Descripción del proyecto de prueba 3",
28     "imagen": "http://ejemplo.com/imagen.jpg",
29     "linkGithub": "http://github.com/usuario/proyecto",
30     "linkDemo": "http://ejemplo.com/demo",
31     "tecnologias": [
32       {
33         "id": 2,
34         "nombre_tecnologia": "typescript"
35       }
36     ]
37   }
38 ]
39
40 ### * Eliminar un proyecto
41 DELETE http://localhost:3001/api/proyecto/11

```

Nota: La anterior ilustración muestra las pruebas de la API en la entidad proyectos con consultas similares a las del usuario, Fuente tomada del Autor.



## 11. Conclusiones

El análisis de tecnologías frontend revela que frameworks como React, Vue y Angular ofrecen características sólidas para construir aplicaciones interactivas e inmersivas. React destaca por su ecosistema de componentes reutilizables y su integración fluida con bibliotecas especializadas como Three.js y WebXR, lo que lo hace ideal para experiencias multimedia avanzadas. Además, tecnologías complementarias como TailwindCSS permiten un diseño ágil, personalizable y responsivo, adaptado tanto a dispositivos móviles como de escritorio. La elección de un framework frontend no solo debe centrarse en su rendimiento, sino también en su comunidad activa, documentación y compatibilidad con herramientas modernas de desarrollo. Según los resultados de la encuesta de Stack Overflow, React sigue siendo el framework más utilizado por los desarrolladores, consolidando su liderazgo en el desarrollo frontend. (Stack Overflow, 2023).

En cuanto al backend, tecnologías como Node.js, Django y Spring Boot ofrecen soluciones robustas y escalables para el manejo de datos complejos, autenticación y almacenamiento multimedia en aplicaciones orientadas a portafolios digitales. Node.js, con su arquitectura asincrónica y compatibilidad con JavaScript, resulta especialmente eficiente en el desarrollo de APIs RESTful que permiten la gestión fluida de contenidos en tiempo real. La integración con bases de datos como MongoDB y PostgreSQL permite almacenar desde archivos estáticos hasta estructuras de datos dinámicas sin comprometer el rendimiento. Además, la elección de una tecnología backend debe considerar su facilidad de integración con el frontend, así como el soporte disponible y las capacidades de escalabilidad que aseguren un crecimiento sostenido del sistema. Node.js se consolida como una solución versátil que





equilibra rendimiento y facilidad de desarrollo en proyectos web modernos. (TechEmpower, 2023).

La implementación de una arquitectura por capas en el desarrollo de la aplicación permite organizar el sistema en niveles con responsabilidades claras, lo cual mejora la mantenibilidad del código, la escalabilidad de la solución y la separación de preocupaciones. Esta estructura modular facilita las tareas de prueba, depuración y despliegue de nuevas funcionalidades sin afectar el sistema completo. Además, promueve buenas prácticas de diseño que son ampliamente reconocidas en el desarrollo profesional de software, lo que contribuye a la calidad técnica del proyecto. Por ejemplo, separar la capa de presentación de la lógica de negocio asegura una mayor flexibilidad para adaptar la interfaz sin comprometer los procesos internos. Las arquitecturas en capas son útiles cuando los desarrolladores quieren implementar funcionalidades que puedan evolucionar de forma independiente en distintas partes del sistema. (Sommerville I. , 2016).

Integrar Next.js con TypeScript en el desarrollo frontend ofrece una serie de beneficios que mejoran la calidad y eficiencia del código. TypeScript proporciona tipado estático, lo que permite detectar errores en tiempo de compilación y facilita la refactorización del código. Por su parte, Next.js ofrece funcionalidades como el renderizado del lado del servidor (SSR) y la generación de sitios estáticos (SSG), que mejoran el rendimiento y la optimización para motores de búsqueda. La combinación de ambas tecnologías resulta en aplicaciones más robustas, mantenibles y escalables. Como se destaca en un artículo de Medium "incorporar TypeScript en un proyecto de Next.js puede mejorar significativamente el flujo de trabajo de desarrollo al ofrecer una mejor herramienta, seguridad de tipos y una mayor mantenibilidad" (Farihatul, 2023).





El desarrollo de este sistema permitió crear una plataforma funcional y visualmente amigable que facilita al usuario la gestión y personalización de su portafolio profesional. A través de un panel principal (**dashboard**) bien estructurado, se ofrecen secciones clave como presentación, proyectos, redes sociales, configuración de colores y datos de usuario. Esta organización contribuye a una experiencia intuitiva y flexible, permitiendo que cada usuario configure su espacio de forma personalizada, según sus necesidades e intereses. La posibilidad de visualizar el portafolio generado en tiempo real, incluyendo enlaces a redes y proyectos con demos y repositorios, enriquece significativamente su valor como herramienta de presentación profesional. Una buena experiencia de usuario permite que el contenido sea útil, utilizable, deseable, encontrable, accesible y creíble, (Nielsen Norman Group, 2021).

Desde el punto de vista técnico, el sistema demuestra una arquitectura backend robusta, organizada en rutas, controladores, esquemas y servicios que separan claramente las responsabilidades del código, facilitando su mantenimiento y escalabilidad. La definición de navegación con la entidad usuario, así como la estructura de base de datos modelada con ChartDB, respaldan la coherencia del flujo de datos y la correcta relación entre entidades. En conjunto, este proyecto no solo logra cumplir con los objetivos funcionales propuestos, sino que también establece una base sólida para futuras ampliaciones y mejoras tanto en el diseño visual como en la lógica de negocio y almacenamiento de información. Una arquitectura de software bien definida permite mejorar la calidad, la eficiencia y la evolución del sistema a lo largo del tiempo, (Bass, 2022).

La implementación y validación de la aplicación web demostraron ser elementos clave para su mejora continua y su adecuación al entorno educativo. A través de las pruebas realizadas con estudiantes del programa de Ingeniería de Sistemas, se identificaron aspectos



funcionales y de usabilidad que fueron corregidos y optimizados, como la validación de formularios, la experiencia visual y el flujo de generación del portafolio académico. Esto evidencia la importancia de involucrar a los usuarios finales en las etapas finales del desarrollo para garantizar una herramienta más eficaz y centrada en sus necesidades reales. Además, el uso de pruebas estructuradas en la API permitió comprobar el correcto funcionamiento de cada entidad del sistema, fortaleciendo la comunicación entre las capas del modelo arquitectónico. La validación de operaciones como creación, consulta y modificación de usuarios aseguró la integridad de los datos y la eficiencia del sistema en tiempo real. Este enfoque metodológico se alinea con las buenas prácticas del desarrollo ágil y centrado en el usuario, donde la retroalimentación temprana es esencial para el éxito del producto, (Pressman, 2020).

La experiencia práctica de los usuarios aportó una visión real del uso de la aplicación, lo que facilitó decisiones más precisas en cuanto a su ajuste y refinamiento. Las pruebas realizadas sobre la API confirmaron que el sistema responde adecuadamente a las operaciones fundamentales, especialmente en módulos críticos como el de usuarios. La validación de los datos, el manejo de errores y la conexión entre capas reflejan una arquitectura sólida y escalable. Estas acciones refuerzan la importancia de realizar pruebas funcionales y de integración como parte fundamental del ciclo de vida del desarrollo de software (Sommerville, 2016), asegurando así la confiabilidad y sostenibilidad del sistema en el entorno académico.

## 12. Bibliografía

Ahmad, M. &. (2021). A Review on the Usage of Scrum in Global Software Development.  
*Journal of Software: Evolution and Process*, 33(3), e2292.



AVSVM cloud. (9 de Mayo de 2024). *Scrum: qué es y por qué se ha convertido en una de las metodologías ágiles más populares*. Obtenido de <https://ausum.cloud/scrum-metodologia-agil-mas-popular-en-empresas/>

Barrett, H. C. (2007). Researching Electronic Portfolios and Learner Engagement: The REFLECT Initiative. *The Journal of Adolescent & Adult Literacy*, 50(6), 436–449.

Bass, L. C. (2022). *Software Architecture in Practice (4th ed.)*. Addison-Wesley.

BeBetter Developers. (2025). *Diseño y desarrollo de Páginas web*. Obtenido de BeBetter: <https://bebetterdevelopers.com/>

Bierman, G. A. (2023). TypeScript: Safety and Scalability in Large JavaScript Projects. *ACM SIGPLAN Notices*.

Canva. (11 de 05 de 2025). *Haz un portafolio digital gratis online*. Obtenido de Canva: [https://www.canva.com/es\\_mx/crear/portafolios-digitales/](https://www.canva.com/es_mx/crear/portafolios-digitales/)

ChartDB. (8 de Junio de 2025). *ChartDB*. Obtenido de <https://app.chartdb.io/>

Creasotol. (2025). *Mejor empresa de desarrollo de sitios web en Cúcuta*. Obtenido de Creasotol: <https://creasotol.com/cucuta/mejor-empresa-de-desarrollo-de-sitios-web-en-cucuta.php>

EL CONGRESO DE COLOMBIA. (18 de Agosto de 1999). *LEY 527 DE 1999*. Obtenido de Funcion Publica: <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=4276>



EL CONGRESO DE COLOMBIA. (17 de Octubre de 2012). *Ley 1581 de 2012*. Obtenido de

Funcion Publica:

<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981>

Emadamerho-Atori, N. (7 de Octubre de 2024). *Angular vs. React vs. Vue.js: Comparing*

*performance*. Obtenido de blog logrocket: <https://blog.logrocket.com/angular-vs-react-vs-vue-js-comparing-performance/>

Farihatul, M. (9 de junio de 2023). *How to use TypeScript with Next.js and the benefits of using*

*TypeScript*. Obtenido de Medium: <https://medium.com/@farihatulmaria/how-to-use-typescript-with-next-js-and-the-benefits-of-using-typescript-c075a2b8e239>

Gobernación de Norte de Santander. (2025). *Plan Estratégico de Tecnologías de la Información (PETI) 2024-2027*. Obtenido de nortedesantander:

[https://administrador.nortedesantander.gov.co/wp-content/uploads/2025/01/PETI\\_2024\\_V15.pdf](https://administrador.nortedesantander.gov.co/wp-content/uploads/2025/01/PETI_2024_V15.pdf)

Hernández-Carrera, R. M.-R.-P. (2023). Digital portfolios and employability: A case study in higher education. *Education and Information Technologies*, 28, 963–980.

JetBrains. (2023). *The State of Developer Ecosystem*. Obtenido de JetBrains:

<https://www.jetbrains.com/lp/devecosystem-2023/>

Li, X. Z. (2022). Design Pattern Optimization in Cloud-based Web Applications: A Singleton Pattern Case Study. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 207–213.



Logan. (1 de Febrero de 2025). *The Most Demanded Frontend Frameworks in 2024*. Obtenido de devjobsscanner: <https://www.devjobsscanner.com/blog/the-most-demanded-frontend-frameworks/>

López-Meneses, E. V.-C.-R. (2022). Portafolios digitales en educación superior: percepción del estudiantado sobre su eficacia académica y profesional. *Revista de Educación a Distancia (RED)*, 22(70).

Nielsen Norman Group. (2021). *The Definition of User Experience (UX)*. Obtenido de nngroup: <https://www.nngroup.com/articles/definition-user-experience/>

OpenAI. (2025). *ChatGPT (version del 19 de marzo)*. Obtenido de ChatGPT-4-turbo: <https://chatgpt.com/>

Pressman, R. S. (2020). *Ingeniería del software: Un enfoque práctico (9.ª ed.)*. McGraw-Hill.

Ramírez, L. &. (2022). Innovación académica y visibilidad estudiantil en la educación superior. *Revista de Tecnología Educativa*, 45–58.

Refactoring Guru. (2021). *Refactoring Guru*. Obtenido de Singleton: <https://refactoring.guru/es/design-patterns/singleton>

Singh, S. (29 de Julio de 2023). *Node.js vs Spring Boot vs Django: Which One Is Best for Beginners?* Obtenido de medium: <https://sandydev.medium.com/node-js-vs-spring-boot-vs-django-which-one-is-best-for-beginners-8782d3be54>

solweb. (2025). *Diseño y desarrollo de Páginas web*. Obtenido de solweb: <https://solwebonline.com/>

Sommerville. (2016). *Ingeniería del software (10.ª ed.)*. Pearson Educación.



*"Formando líderes para la construcción de un nuevo país en paz"*

Universidad de Pamplona  
Pamplona - Norte de Santander - Colombia  
Tel: (+57) 3152129105 - 3160211175



Sommerville, I. (2016). *Ingeniería de Software* (10ª ed.). *Pearson Educación.*, 169.

Stack Overflow. (2023). *Developer Survey Results*. Obtenido de Stack Overflow:

<https://survey.stackoverflow.co/2023/>

Suratica. (1 de Julio de 2024). *Qué es el backend y por qué infravaloras a los desarrolladores*.

Obtenido de Suratica: <https://www.suratica.es/que-es-el-backend/>

TechEmpower. (2023). *Framework Benchmarks Round 23*. Obtenido de techempower:

<https://www.techempower.com/benchmarks/>

Tilkov, S. &. (2022). Node.js: Using JavaScript to Build High-Performance Network Programs.

*IEEE Internet Computing*, 26(1), , 78–84.

Torres, J. (2023). Plataformas digitales para la proyección de proyectos universitarios. *Revista*

*Colombiana de Educación en Ingeniería*, 77–89.

UNESCO. (25 de Septiembre de 2015). *Asamblea General Naciones Unidas*. Obtenido de

Transformar nuestro mundo: La Agenda 2030 para el Desarrollo Sostenible.

Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura.:

[https://unctad.org/system/files/official-document/ares70d1\\_es.pdf](https://unctad.org/system/files/official-document/ares70d1_es.pdf)

Vercel. (2023). *Next.js Documentation*. Obtenido de Vercel: <https://nextjs.org/docs>

Wiz. (11 de 05 de 2025). *Portafolio Digital, Hacer un Portafolio Online*. Obtenido de Wiz:

<https://es.wix.com/portafolio-digital>