



**UNIVERSIDA TÉCNICA “LUIS VARGAS TORRES” DE ESMERALDAS**

**FACULTAD DE INGENIERÍAS**

**CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN**

**TRABAJO DE TITULACIÓN**

Previo a la obtención del Título de:

**INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN**

**TITULO DEL PROYECTO:**

Desarrollo de sitios web para la gestión del currículum vitae normalizado (CVN) y  
bolsa de empleo para estudiantes y graduados de la UTLVTE

**AUTOR:**

Montaño Solis Luis Ariel

**TUTOR:**

Ing. Richard Alejandro Masías Lara, MSc.

**ESMERALDAS-ECUADOR**

**2025**

## **DECLARACIÓN DE DERECHOS DE AUTORÍA**

Yo, **Luis Ariel Montaño Solis**, con **C.C.0803700657** declaro bajo juramento que el trabajo de integración curricular, es de mi autoría y que ninguna otra persona o profesional lo ha expuesto previamente, y que he consultado y referenciado todas las bibliografías que se anexan en este documento.

Con la presente declaración transfiero los derechos de propiedad intelectual correspondientes de este trabajo, a la Universidad Técnica “Luis Vargas Torres” de Esmeraldas, según lo establecido por la Ley de Propiedad Intelectual, con la aprobación del Decreto N°. 508 y Registro Oficial N° 320 por la normativa institucional vigente.

Atentamente,

---

**Luis Ariel Montaño Solis**

**C.C. 0803700657**

## **CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN**

Yo, **Richard Alejandro Masías Lara**, con **C.C. 0803487453**, Docente de la Universidad Técnica “Luis Vargas Torres” de Esmeraldas, declaro que bajo mi tutoría fue desarrollado el trabajo de integración curricular titulado: **“Desarrollo de sitios web para la gestión del currículum vitae normalizado (CVN) y bolsa de empleo para estudiantes y graduados de la UTLVTE”**, realizado por **Luis Ariel Montaño Solis**, con **C.C.0803700657**, bajo la modalidad de Titulación “Proyectos de Integración”, que cumple con todos los requisitos determinados por al Universidad Técnica “Luis Vargas Torres” de Esmeraldas.

Atentamente,

---

**Ing. Richard Alejandro Masías Lara, MSc**

**C.C. 0803487453**

**TRIBUNAL EXAMINADOR DE GRADUACIÓN**

**FACULTAD DE INGENIERÍAS**

**CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN**

Los miembros del tribunal del Proyecto Investigación titulado: “**Desarrollo de sitios web para la gestión del currículum vitae normalizado (CVN) y bolsa de empleo para estudiantes y graduados de la UTLVTE**”, y su propuesta, elaborada por **Luis Ariel Montaño Solis**, como requisito previo a la investidura de Ingeniero en Tecnologías de la Información, una vez evaluado lo aprueben.

Atentamente,

---

**Ing. Jonathan Cárdenas Ruperti, MSc**

**C.C. 0802371567**

---

**Ing. Guillermo Cedeño Rodríguez, MSc**

**C.C. 0802010959**

---

**Ing. Viviana Aparicio Izurieta, MSc**

**C.C. 0802771535**

## **DEDICATORIA**

Se que la mayor parte de las dedicatorias son dirigidas primeramente a Dios; esta dedicatoria también la lleva, pero principalmente el presente trabajo se lo dedico a **mi yo del pasado**, a ese niño que nunca imaginó llegar tan lejos, que siempre sueña, nunca se rinde, siempre busca la manera de salir adelante, no solo por satisfacción propia, si no por su familia.

A ese niño que, con la ayuda de Dios, sigue cumpliendo sus objetivos, queriendo ser un buen ejemplo para sus siete hermanas menores, sus cinco hermanos mayores, sus sobrinos, primos y demás seres queridos, demostrando que, aunque algo sea muy difícil, siempre se puede lograr.

**A mi padre**, que, aunque trabaje duró sigue dándonos su amor, sonrisa y apoyo. Ese apoyo que me hizo el hombre que soy ahora.

**A mi madre**, que con su cuidado amor y comprensión nos apoya a mí y a mis hermanos incondicionalmente.

**A los docentes**, que siempre tuvieron fe en que si podría hacerlo, aquellos que me dieron su apoyo, sus estrategias de aprendizaje.

**A mis amigos**, los que siempre me apoyaron cuando lo necesité.

Este logro pienso disfrutarlo con todos ustedes.

Siempre seguiré mejorando, para llegar a ser grande...

**UNO DE LOS GRANDES.**

## **AGRADECIMIENTOS**

Agradezco a ese niño que nunca que se rindió y se sigue esforzando, también a Dios, por ser mi guía y fortaleza en cada paso. Muchas gracias por darme, salud, sabiduría y cada una de las oportunidades que me han permitido cumplir poco a poco mis metas.

A mi familia, por su gran amor, confianza y apoyo. Gracias por enseñarme el valor del esfuerzo e impulsarme a seguir adelante.

A todos aquellos que me formaron como persona, amigos y docentes. Les agradezco mucho.

Este y todos los trabajos que se vienen representan ese esfuerzo compartido. A todos ustedes,  
**GRACIAS DE TODO CORAZÓN.**

## **RESUMEN**

En la actualidad, las Instituciones de Educación Superior (IES) enfrentan el reto de fortalecer la empleabilidad de sus estudiantes y graduados mediante el uso de herramientas tecnológicas que faciliten la gestión de información profesional y la conexión con el mercado laboral. En ese contexto, la Universidad Técnica “Luis Vargas Torres” de Esmeraldas (UTLVTE) carecía de una solución digital que permitiera normalizar/estandarizar las hojas de vida de sus estudiantes, egresados y graduados, también gestionar oportunidades laborales de forma integrada. Por ello, el siguiente trabajo de titulación tiene como objetivo desarrollar dos plataformas web interconectadas: una para la gestión del Curriculum Vitae Normalizado y otra para Bolsa de Empleo, orientadas a optimizar la organización, actualización y uso de la información profesional académica, así como mejora la vinculación entre los egresados y el sector productivo. La investigación adoptó un enfoque descriptivo-experimental y cualitativo, utilizando la metodología ágil Xtreme Programin (XP), la cual permitió un desarrollo iterativo y adaptable a los requerimientos de los usuarios. Se emplearon tecnologías modernas para el backend (Laravel), frontend (Vue) y un gestor de base de datos MySQL, garantizando escalabilidad, seguridad y rendimiento. Los resultados demostraron que la plataforma CVN consiguió estandarizar el formato curricular institucional, generándolos en formato PDF, mientras que la Bolsa de Empleo facilitó la publicación, búsqueda y postulación a ofertas laborales, añadiendo como plus el apartado de emprendimientos. Mediante API RESTful se logró integrar ambas plataformas, esto permitió la interoperabilidad de los datos, reduciendo la redundancia de información y mejorando la experiencia de usuario. Las pruebas funcionales y de usabilidad confirmaron la eficiencia, seguridad y estabilidad de las plataformas, con mecanismos de autenticación mediante tokens JWT (Json Web Token) y cifrado de

contraseñas. Finalmente, la implementación de ambas plataformas contribuye al fortalecimiento de transformación digital institucional, potenciando la empleabilidad, el seguimiento a graduados y la vinculación universidad – empresa, además de sentar las bases para futuras ampliaciones del sistema y nuevas investigaciones en el ámbito tecnológico universitario.

**Palabras clave:** Currículum Vitae Normalizado, Bolsa de Empleo, Laravel, Vue, MySQL, Metodología XP, empleabilidad, interoperabilidad, transformación digital.

## **ABSTRACT**

Currently, Higher Education Institutions (IES) face the challenge of strengthening the employability of their students and graduates through the use of technological tools that facilitate the management of professional information and connection to the labor market. In this context, the Technical University "Luis Vargas Torres" of Esmeraldas (UTLVTE) lacked a digital solution that allowed for the normalization/standardization of the resumes of its students, graduates, and alumni, as well as the management of job opportunities in an integrated manner. Therefore, the following thesis aims to develop two interconnected web platforms: one for the management of the Standardized Curriculum Vitae and another for the Job Bank, aimed at optimizing the organization, updating, and use of academic professional information, as well as improving the connection between graduates and the productive sector. The research adopted a descriptive-experimental and qualitative approach, using the agile Xtreme Programming (XP) methodology, which allowed for iterative development adaptable to user requirements. Modern technologies were used for the backend (Laravel), frontend (Vue), and a MySQL database manager, ensuring scalability, security, and performance. The results demonstrated that the CVN platform standardized the institutional resume format, generating them in PDF format, while the Job Board facilitated the publication, search, and application for job openings, adding a startup section as an added benefit. A RESTful API enabled the integration of both platforms, enabling data interoperability, reducing information redundancy, and improving the user experience. Functional and usability tests confirmed the efficiency, security, and stability of the platforms, with authentication mechanisms using JWT tokens (JSON Web Tokens) and password encryption. Finally, the implementation of both platforms contributes to strengthening the institutional digital transformation, enhancing employability, graduate

follow-up, and university-business relationships. It also lays the groundwork for future system expansions and new research in the field of university technology.

**Keywords:** Standardized Curriculum Vitae, Job Board, Laravel, Vue, MySQL, XP Methodology, employability, interoperability, digital transformation.

## ÍNDICE GENERAL

<i>DEDICATORIA</i> .....	<i>IV</i>
<i>AGRADECIMIENTOS</i> .....	<i>VI</i>
<i>RESUMEN</i> .....	<i>VII</i>
<i>ABSTRACT</i> .....	<i>IX</i>
<i>ÍNDICE DE FIGURAS</i> .....	<i>XIV</i>
<i>INTRODUCCIÓN</i> .....	<i>I</i>
Descripción de la situación problemática.....	2
Formulación del problema.....	3
Importancia de la investigación.....	3
Objetivos de la investigación .....	5
<i>CAPÍTULO I. MARCO TEÓRICO</i> .....	<i>6</i>
1.1..... Antecedentes de la investigación	
.....	6
1.2.....Bases teóricas	
.....	8
1.2.1.....Curriculum Vitae Normalizado (CVN)	
8	
1.2.2.....Bolsa de Empleo	
9	
1.2.3..... Introducción a las aplicaciones web modernas	
9	
1.2.4..... Vue.js	
12	
1.2.5..... Laravel	

1.2.6.....Comparación de frameworks:	
18	
1.2.7.....Diseño de Base de datos y migraciones	
28	
1.2.8.....Control de versiones Git y GitHub	
29	
1.2.9.....Despliegues y pruebas	
29	
1.3.....Definición de términos básicos	
31	
1.3.1.....Tecnologías y herramientas	
31	
1.3.2.....Componentes del Sistema	
31	
1.3.3.....Arquitectura y comunicación	
32	
<i>CAPÍTULO II. METODOLOGÍA</i>	33
2.1. Diseño metodológico.....	33
2.2. Técnicas de recolección de datos .....	50
<i>CAPÍTULO III. RESULTADOS Y DISCUSIÓN</i>	51
<i>CONCLUSIONES</i>	57
<i>RECOMENDACIONES</i>	58
<i>REFERENCIAS BIBLIOGRÁFICAS</i>	59
<i>ANEXOS</i>	67

## ÍNDICE DE TABLAS

	Pág.
<i>Tabla 1. Comparativa de frameworks frontend.....</i>	<i>19</i>
<i>Tabla 2. Comparativa de frameworks backend .....</i>	<i>24</i>
<i>Tabla 3. Requerimientos funcionales.....</i>	<i>34</i>
<i>Tabla 4. Requerimientos no funcionales.....</i>	<i>35</i>
<i>Tabla 5. Priorización de Requerimientos .....</i>	<i>39</i>
<i>Tabla 6. Actividades con iteraciones .....</i>	<i>40</i>
<i>Tabla 7. Actividades con iteraciones .....</i>	<i>41</i>
<i>Tabla 8. Componentes Backend.....</i>	<i>46</i>
<i>Tabla 9. Repositorio GitHub de las plataformas.....</i>	<i>52</i>

## ÍNDICE DE FIGURAS

	Pág.
<i>Figura 1. Arquitectura Cliente Servidor.....</i>	<i>10</i>
<i>Figura 2. API REST (API RESTful).....</i>	<i>11</i>
<i>Figura 3. Representación de la relación entre frontend y el backend.....</i>	<i>12</i>
<i>Figura 4. Organización en árbol de componentes anidados.....</i>	<i>13</i>
<i>Figura 5. Diagrama UML de la plataforma CVN .....</i>	<i>37</i>
<i>Figura 6. Diagrama UML de la plataforma Bolsa de Empleo.....</i>	<i>38</i>
<i>Figura 7. Tablas para la plataforma Bolsa de Empleo .....</i>	<i>42</i>
<i>Figura 8. Tablas para la plataforma CVN .....</i>	<i>43</i>
<i>Figura 9. Arquitectura MVVM - Vue .....</i>	<i>43</i>
<i>Figura 10. Página principal .....</i>	<i>44</i>
<i>Figura 11. Página principal – Inicio de Sesión.....</i>	<i>44</i>
<i>Figura 12. Arquitectura MVC - Laravel.....</i>	<i>45</i>
<i>Figura 13. Arquitecturas MVVM (frontend) y MVC(backend).....</i>	<i>46</i>
<i>Figura 14. Aplicación Bitsive SSH Client para conectar al servidor AlmaLinux.....</i>	<i>48</i>
<i>Figura 15. Acceso a las carpetas de apache en donde estarán las plataformas.....</i>	<i>48</i>
<i>Figura 16. Archivo de configuración para el Alias de la plataforma CVN y Bolsa de Empleo.....</i>	<i>49</i>
<i>Figura 17. Inicio de sesión exitoso para el usuario Estudiante y Administrador-CVN.....</i>	<i>49</i>
<i>Figura 18. Obtención de datos .....</i>	<i>50</i>

<i>Figura 19. Pruebas peticiones HTTP .....</i>	50
<i>Figura 20. Gráfica de Funcionalidad y Rendimiento de la Plataforma CVN55</i>	
<i>Figura 21. Gráfica de Funcionalidad y Rendimiento de la Plataforma Bolsa de Empleo.....</i>	55

## **INTRODUCCIÓN**

En las últimas décadas, la educación superior se ha extendido masivamente. Se matriculan más estudiantes que nunca, y la matrícula mundial en educación superior ha pasado del 19% en 2000 al 43% en 2023. Pero aquí el truco: que haya más estudiantes en la universidad no significa automáticamente que se gradúen más estudiantes. De hecho, la tasa de graduación sólo ha subido del 17% al 26% en el mismo periodo y la proporción de graduados que consiguen buenos empleos es aún menor. Se trata de una gran brecha que requiere una cuidadosa consideración, tanto por parte de las universidades como de los responsables de políticas públicas [1]. Es así como, la Unión Europea ha impulsado iniciativas como el Europass Curriculum Vitae (CV), un formato de curriculum estandarizado que facilita la presentación de información laboral y académica [2]. De igual modo, plataformas como LinkedIn considerada como la mayor red profesional online a nivel mundial para encontrar el trabajo o las prácticas perfectas, conectar y fortalecer las relaciones profesionales y aprender las aptitudes necesarias para tener éxito.

Actualmente en Ecuador, varias Instituciones de Educación Superior (IES) disponen de servicios institucionales para apoyar la empleabilidad de sus estudiantes y graduados/as. Entre ellas la Universidad Politécnica Salesiana, ofrece una bolsa de trabajo para apoyar la empleabilidad de sus estudiantes y graduados de nivel de grado y posgrado, también posee una opción para adjuntar el CV y para llenar ciertos campos del CV [3]. Esta y la mayoría de las plataformas de empleo de las universidades del Ecuador poseen las mismas funcionalidades, proporcionan el llenado de un CV dentro del mismo sistema lo cual dificulta tener un CV más completo y que cubra los puntos importantes que un CV debe tener para que respondan a las necesidades reales del mercado y sus bolsas de trabajo no integran la generación y publicación de emprendimientos de sus estudiantes.

Es así como la Universidad Técnica “Luis Vargas Torres” de Esmeraldas enfrentando el desafío de proporcionar a sus estudiantes, egresados/as y graduados/as herramientas que faciliten su transición al ámbito profesional. Por ellos, el presente proyecto tiene como objetivo del desarrollo de dos plataformas digitales, una que gestione el Curriculum Vitae Normalizado (CVN) en un formato estándar institucional, y otra de Bolsa de Empleo que no solamente gestione la búsqueda de trabajo para los egresados/as y graduados/as si no que permita que estos mismos usuarios puedan publicar, generar y postularse a ofertas de emprendimientos.

### **Descripción de la situación problemática**

La Universidad Técnica “Luis Vargas Torres de Esmeraldas”, actualmente posee una inexistencia de sitios/plataformas que puedan gestionar de forma estandarizada los perfiles profesionales de los estudiantes y los graduados que los conecte de manera eficiente con las ofertas de trabajo. Esta falta de iniciativa provoca dificultades en la inserción profesional, fragmentación en la búsqueda de empleo y una débil vinculación con la universidad y el entorno laboral. Aunque se sabe que existen bolsas de empleo genéricas al igual que páginas que generan currículos, existe un vacío de conocimiento sobre el diseño e implementación de una solución específica y adaptada a las necesidades contextuales de la institución y su impacto real.

Dado que, la UTLVTE no cuenta con una plataforma que gestione currículos estandarizado y conecte a graduados con el mundo laboral, se generan tres desafíos críticos:

1. Dificultades en la inserción profesional debido a la dispersión de perfiles no normalizados

2. Fragmentación en la búsqueda de empleo al depender de herramientas externas no adaptadas al contexto institucional; y
3. Una relación débil entre la universidad, sus egresados/as y/o graduados/as y el mercado laboral, limitando el seguimiento de trayectorias profesionales y la retroalimentación para mejorar la oferta académica.

### **Formulación del problema**

¿De qué manera el desarrollo de dos plataformas web, una para la gestión del CVN y otra para la bolsa de empleo universitaria, puede mejorar la vinculación entre los estudiantes y graduados de la Universidad Técnica “Luis Vargas Torres” de Esmeraldas con el sector productivo local?

### **Importancia de la investigación**

El acceso a estas plataformas estará disponible para todos/as las/los estudiantes, egresados/as y graduados/as de la Universidad Técnica “Luis Vargas Torres” de Esmeraldas, este sistema permitirá la creación, almacenamiento, actualización y estandarización de la información profesional, así conectar de forma directa a los graduados con oportunidades laborales publicadas por empresas e instituciones que tengan convenio con la UTLVTE, y que puedan los egresados/as y graduados/as crear ofertas en base a sus emprendimientos. Al dar este “plus” en las plataformas, no solo podrán postularse a ofertas de empresas, sino también, postularse a ofertas de emprendimiento.

Cierto es que, para que la bolsa de empleo tenga muchas ofertas de empresas e instituciones se necesita que estás en conjunto con la universidad se comprometan a

publicarlas en la plataforma y aceptar a graduados/as de la instrucción. Del mismo modo, para que el CVN esté en constante actualización se debe tener un compromiso, primero de los estudiantes, egresados/as y graduados/as, segundo los docentes de las unidades académicas. Estas limitantes hacen que plataformas de estos tipos queden en el olvido.

El desarrollo de la plataforma Bolsa de Empleo contribuye significativamente en la evaluación universitaria para la acreditación, específicamente en el indicador veinte tres de seguimiento a graduados, el cual dice que el seguimiento a graduados es el proceso a través del cual las universidades y escuelas políticas gestionan la información referente a la empleabilidad pertinente, campos ocupacionales y los niveles de satisfacción de los graduados de las carreras o programas ofertados por las UEP. Estos procesos permiten orientar la toma de decisiones institucionales en torno a la oferta académica, gestión del currículum, los niveles de inserción laboral de sus graduados y fortalecer los procesos de aseguramiento interno de la calidad y de mejora continua [4]. Pero no solo se debe basar en la inserción laboral, si no en la construcción de un perfil más profesional, a esto es donde interviene la creación de un CVN que normalice la información simplificando la gestión de datos y permitiendo una evaluación justa y equitativa de los perfiles académicos y profesionales. Esta medida no solo mejoraría la eficiencia de la universidad, sino que también contribuiría a la equidad en la toma de decisiones académicas y de empleo.

El desarrollo de un CVN normalizado y una bolsa de empleo universitaria posicionaría a la Universidad Técnica “Luis Vargas Torres” de Esmeraldas como una institución pionera en la región, destacándola como una entidad comprometida con el éxito de sus estudiantes y su inserción en el mercado laboral.

## **Objetivos de la investigación**

### **-Objetivo general:**

Desarrollar dos plataformas digitales (un sistema de Currículum Vitae Normalizado -CVN- y una Bolsa de Empleo) mediante el uso de tecnologías full-stack modernas y un diseño centrado en el usuario, con el fin de optimizar la gestión de la información profesional-académica, y facilitar la vinculación eficiente entre egresados y las oportunidades laborales pertinentes.

### **-Objetivos específicos:**

1. Identificar los requerimientos y expectativas clave de los usuarios (estudiantes, egresados/as y graduados/as) para definir las funcionalidades y características esenciales que guiarán el desarrollo de las plataformas.
  
2. Diseñar la arquitectura de las plataformas CVN y Bolsa de Empleo, que optimice la gestión de la información y la interacción bidireccional de los usuarios.
  
3. Validar el funcionamiento, usabilidad y rendimiento de las plataformas en un entorno controlado, para garantizar la calidad y la aceptación por parte de los usuarios finales.

## **CAPÍTULO I. MARCO TEÓRICO**

### **1.1. Antecedentes de la investigación**

La empleabilidad de estudiantes y graduados/as constituye una de las prioridades de las IES, dado que forma parte de los procesos de acreditación y de la vinculación en el sector productivo. Es así como, el desarrollo de plataformas digitales de bolsas de empleo y sistemas de gestión del curriculum vitae normalizado se han convertido en una estrategia clave para facilitar la inserción laboral y fortalecer la relación universidad-empresa. En 2021 [5] desarrolló un sistema web de empleos que no solo gestiona ofertas laborales, si no, el seguimiento de egresados/as y graduados/as del Instituto Superior Tecnológico “José Chiriboga Grijalva” (ITCA), para este desarrollo se utilizó la metodología SCRUM y el framework Laravel, siendo este su principal objetivo. El sistema web funcional permitió al instituto recolectar información relevante sobre la inserción laboral, lo que llevó a concluir que esta implementación digital es una solución efectiva para mitigar las dificultades en la empleabilidad y cumplir con los requisitos de la calidad y seguimiento a graduados.

No obstante, es crucial considerar las barreras identificadas en la literatura para garantizar el éxito a largo plazo, por ejemplo, el estudio de [6] identificó un desajuste crítico entre las competencias académicas de los estudiantes y las demandas del mercado actual. Esta investigación, que se basó en encuestas, concluyó que la efectividad de las plataformas digitales está sujeta a una mejor alineación curricular y a la capacitación de los estudiantes para una presentación efectiva de sus perfiles profesionales.

Estos desafíos de inserción laboral se agravan en contextos donde la infraestructura digital es incipiente, lo cual justifica la necesidad de una plataforma institucional como la

que propone este trabajo de investigación. Así lo evidenció la investigación de [7], la misma que diagnostico la falta de un servicio institucionalizado en la provincia de Esmeraldas. Este estudio concluyó que una herramienta específica para el contexto local es fundamental para fortalecer la interacción entre las Instituciones de Educación Superior de Esmeraldas y los empleadores.

Precisamente, una de las herramientas digitales para solventar este desajuste y la fragmentación de perfiles es la estandarización, meta primordial de los CVN. A este propósito, el [8] demuestra la viabilidad de desarrollar una herramienta interna que se ajuste a un estándar externo, esta iniciativa logró implementar una solución para generar documentos en formato PDF y XML, confirmando que al estandarizar un currículum se facilita la gestión de la información curricular y optimiza los procesos de evaluación y acreditación universitaria.

Asimismo, [9] desarrolló una investigación en la Pontificia Universidad Católica del Ecuador Sede Esmeraldas sobre el seguimiento a graduados y egresados de la Escuela de Contabilidad y Auditoría, evidenciando la necesidad de implementar mecanismos digitales que fortalezcan el vínculo entre la universidad y sus titulados. Su tesis destacó la importancia de contar con plataformas institucionales que permitan registrar información de los graduados, conocer su situación laboral y ajustar la oferta académica a las demandas del mercado, contribuyendo a los procesos de mejora continua y acreditación de las instituciones de educación superior.

Finalmente, en [10] presenta una aplicación web para el seguimiento a graduados y bolsa de empleo de un IES, con el propósito de fortalecer el vínculo de la institución con sus egresados y gestionar ofertas laborales, El desarrollo de un aplicación web funcional

cuya validación se realizó mediante encuestas online, las cuales mostraron una alta satisfacción del 85% de los participantes y una utilidad del módulo de bolsa de empleo. Concluyendo que la aplicación web es una solución eficaz para la optimización de los procesos de seguimiento y vinculación, siendo percibida como altamente útil y usable por los usuarios.

## **1.2. Bases teóricas**

En esta sección se presentan los fundamentos conceptuales, los enfoques teóricos y las arquitecturas tecnológicas que sustentan el desarrollo de plataformas web para la gestión del CVN y la bolsa de empleos.

### **1.2.1. Curriculum Vitae Normalizado (CVN)**

#### **1.2.1.1. ¿Qué es un curriculum?**

El curriculum es un concepto que dentro del discurso acerca de la educación denomina y demarca una realidad existente e importante en los sistemas educativos; un concepto que, si bien es cierto que no acoge bajo su paraguas a toda la realidad de la educación, sí que se ha convertido en uno de los núcleos de significación más densos y extensos para comprenderla en el contexto social, cultural, entender las diversas formas en las que se ha institucionalizado [11].

### **1.2.2. Bolsa de Empleo**

#### **1.2.2.1. Conceptos de empleabilidad**

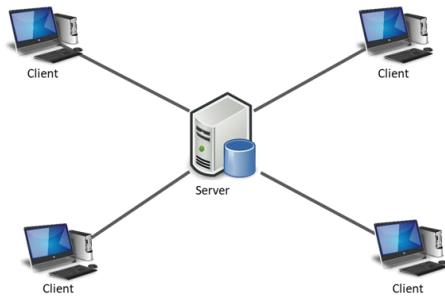
El concepto de empleabilidad ha sido un elemento central en la implementación y diseño de las políticas activas de mercado, en especial en la zona de la Unión Europea y países con sistemas capitalistas coordinados. Desde una noción general, la empleabilidad es la capacidad de los sujetos de adquirir un primer empleo, mantenerlo y obtener uno nuevo [12].

### **1.2.3. Introducción a las aplicaciones web modernas**

#### **1.2.3.1. Arquitectura cliente-servidor.**

La palabra arquitectura se refiere a la estructura física y lógica de cómo está compuesta una red de comunicación o red de distribución de información, cliente se refiere a la máquina que solicita alguna petición o información y el servidor es la máquina que proporciona la información a la máquina cliente, todo esto por medio de una red informática de comunicaciones y que están sujetos por medio de protocolos de comunicación ya sea TCP/IP U OSI. Por lo regular tcp/ip que es la tecnología más popular a nivel mundial [13].

En este contexto podemos decir que, la tecnología Cliente/Servidor es el procesamiento cooperativo de la información por medio de un conjunto de procesadores, en el cual múltiples clientes, distribuidos geográficamente, solicitan requerimientos a uno o más servidores centrales [14].



**Figura 1. Arquitectura Cliente Servidor**

Fuente: [15]

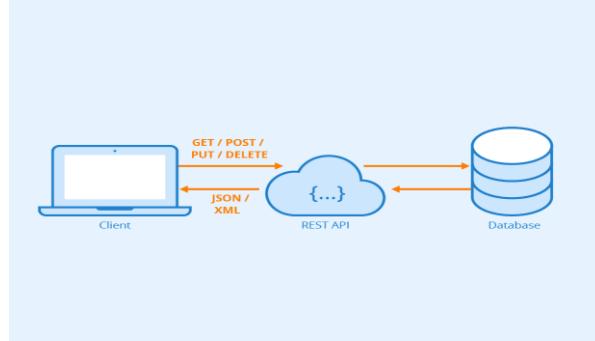
#### 1.2.3.2. Principios RESTful y API

REST significa Transferencia de Estado Representacional (Representational State Transfer), refiriéndose a un estilo de arquitectura de software que cuenta con 6 principios enfocados principalmente en cómo se definen y abordan los recursos en red en la web. La arquitectura

REST es una arquitectura cliente-servidor diseñada para usar HTTP, que se clasifica como un protocolo de comunicación sin estado. Durante los últimos años, el estilo arquitectónico REST se ha utilizado para guiar el diseño y desarrollo de la arquitectura de la Web moderna. En la arquitectura REST, los clientes y servidores intercambian representaciones de recursos utilizando una interfaz y un protocolo estandarizados. Estas características ayudan a proporcionar aplicaciones REST simples y ligeras. Por lo tanto, con respecto al alcance de la confiabilidad, los servicios RESTFUL superan las limitaciones de los servicios Simple Object Access Protocol (SOAP) y logran mejores resultados, especialmente en las comunicaciones móviles [16].

Una API (interfaz de programación de aplicaciones) es un conjunto de herramientas, definiciones y protocolos que se utiliza para integrar los servicios y el software de aplicaciones. Es lo que permite que sus productos y servicios se comuniquen

con otros, sin tener que diseñar permanentemente una infraestructura de conectividad nueva [16]. Si bien, las APIs facilitan la interoperabilidad entre diferentes sistemas, permitiendo a las organizaciones adaptarse a demandas creciente [17].



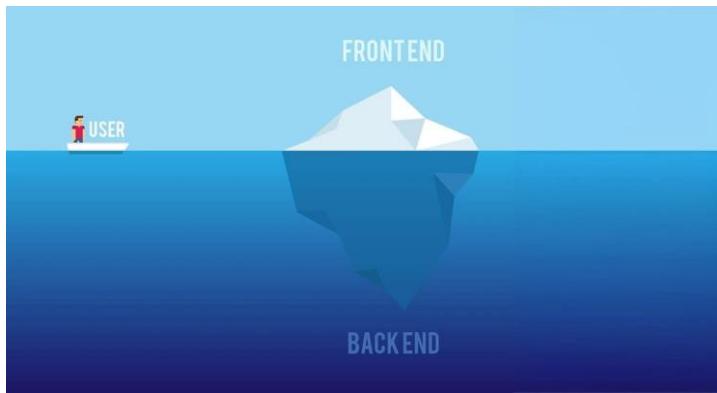
**Figura 2. API REST (API RESTful)**

**Fuente:** [18]

#### 1.2.3.3. Comunicación frontend-backend.

En el libro de Programación Web del Frontend al Backend [19], definen al frontend como la parte visible de una aplicación web, lo que los usuarios ven y con lo que interactúan. El frontend se compone principalmente de lenguajes de programación como HTML, CSS y JavaScript, que permiten la creación de páginas web dinámicas y atractivas. Por otro lado, el backend se encarga de la lógica detrás de una aplicación web. Esto incluye la gestión de la base de datos, la seguridad y la lógica de negocio de la aplicación. Cabe destacar que uno no funciona sin el otro, de manera que, aunque haya desarrolladores que se especialicen en una de los dos, deben conocer cómo se debe programar el otro, ya que pueden aparecer problemas a lo largo del desarrollo de un sistema que deban solucionar [20].

Así se explica en la siguiente imagen.



**Figura 3. Representación de la relación entre frontend y el backend**

Fuente: [20]

#### **1.2.4. Vue.js**

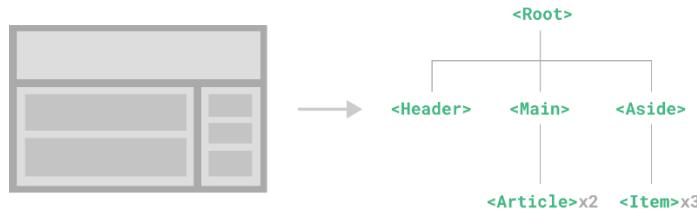
##### **1.2.4.1. Historia y características de Vue.js**

Vue fue creado por Evan You después de trabajar para Google usando AngularJS en varios proyectos. Más tarde, resumió su proceso de pensamiento: "Pensé, ¿qué tal si pudiera extraer la parte que realmente me gustaba de Angular y construir algo realmente liviano?" [21].

Vue.js es un framework frontend de JavaScript que ha crecido rápidamente inspirado en Angular.js, Reactive.js y Rivets.js que proporciona un diseño de interfaz de usuario simplificado, manipulable y con una capacidad de respuesta profunda. Se describe como un framework de modelado Modelo-Vista-Vista Modelo(MVVM), basado en el concepto de vinculación bidireccional de datos a componentes y vistas; es increíblemente rápido, supera la velocidad de otros principales frameworks de JavaScript, y es muy fácil de integraren los proyectos web [22].

#### 1.2.4.2. Componentes, reactividad y ciclo de vida

Los componentes permiten dividir la interfaz de usuario en partes independientes y reutilizables, y considerar cada una de ellas de forma aislada [23]. Es habitual que una aplicación se organice en un árbol de componentes anidados



**Figura 4. Organización en árbol de componentes anidados**

Fuente: [23]

La programación reactiva es un paradigma cada vez más común en el desarrollo de aplicaciones modernas gracias a su capacidad de manejar datos de manera eficiente y reactiva. Este enfoque permite que las aplicaciones respondan automáticamente a las modificaciones de datos, mejorando tanto la eficiencia como la experiencia del usuario [24]. Esta misma dice que, Vue.js ofrece dos enfoques principales para manejar la reactividad: el Options API y el Composition API, cada uno con sus características únicas.

En Vue, los ciclos de vida representan las etapas desde que un componente es creado hasta que es destruido [24]. Por ejemplo, debe configurar la observación de datos, compilar la plantilla, montar la instancia en el DOM y actualizar este cuando los datos cambian [25].

#### 1.2.4.3. Manejo de estado: Vuex

Vuex es una librería que permite incorporar de una forma relativamente sencilla los conceptos de arquitectura Flux a los proyectos desarrollados utilizando el framework Vue.js. La misma proporciona un depósito o store centralizado para todos los componentes de una aplicación [26].

Es decir que, en vez de tener un objeto reactivo dentro de la propiedad “data” de nuestros componentes, Vuex maneja este objeto reactivo dentro de un store, alojado bien alto en nuestra aplicación para que cualquier componente pueda acceder a este objeto en caso de ser requerido, y cuando este se vea actualizado, se actualizarán también los componentes de nuestra aplicación que hagan uso de él [27].

#### 1.2.4.4. Validaciones de formulario en Vue

Los formularios son parte de la mayoría de las aplicaciones web actuales. Permiten recibir datos de los usuarios para luego validarlos y poder interactuar con ellos [28]. Vue.js ofrece funciones para simplificar su creación, gestionar la entrada del usuario y garantizar la precisión de los datos. Puedes comprobar si los campos están completos, siguen patrones específicos o cumplen los requisitos de longitud. Esto se puede hacer mediante directivas como v-model y v-bind [29].

La validación de formularios es soportada de forma nativa por el navegador. Sin embargo, diferentes navegadores manejarán las cosas de una manera que puede ser un poco complicada o trámposa. Incluso cuando la validación es soportada perfectamente, algunas veces habrá la necesidad de personalizar las mismas, por lo cual, una solución basada en Vue podrá ser más apropiada [30].

### 1.2.5. Laravel

#### 1.2.5.1. Historia y características de Laravel.

Laravel fue creado por Taylor Otwell en 2011, pero solo en 2014 comenzó a atraer la atención de los desarrolladores [31].

El framework Laravel utiliza PHP, es conocido por su elegante sintaxis y su enfoque en hacer que el desarrollo web sea un proceso simple y agradable. Proporciona un conjunto de herramientas ricas y completas, incluyendo migraciones de base de datos,

sistema de colas y una potente capa de abstracción de base de datos a través de Eloquent ORM. Este framework también se destaca por su facilidad de uso y su capacidad para acelerar el desarrollo de aplicaciones gracias a su amplia gama de características preconstruidas y paquetes disponibles a través de Composer, el gestor de paquetes de PHP. Su desarrollo rápido y una comunidad en crecimiento puede enfrentarse a limitaciones de rendimiento en aplicaciones de gran escala y alto tráfico, lo que puede requerir ajustes adicionales y una optimización cuidadosa [32].

#### 1.2.5.2. Estructura Modelo Vista Controlador en Laravel.

El Modelo Vista Controlador (MVC) fue definido por Trygve Reenskaug a finales de los 70 en laboratorios de investigación de Xerox [33]. El patrón Modelo, Vista y Controlador (mvc) es el más extendido para el desarrollo de aplicaciones donde se deben manejar interfaces de usuarios, éste se centra en la separación de los datos o modelo, y la vista, mientras que el controlador es el encargado de relacionar a estos dos. Su principal característica es aislar la vista del modelo [34].

En ese contexto, Laravel utiliza el patrón Modelo-Vista-Controlador para organizar el código de la aplicación de manera clara y estructurada [32]. Esta arquitectura aporta muchas soluciones ya preparadas, fáciles de configurar y adaptar a los problemas que todos los desarrolladores web sufren en el proceso de creación [31].

#### 1.2.5.3. Eloquent ORM – mapeo objeto-relacional.

Eloquent ORM (Object-Relational Mapping) es el sistema de mapeo objeto relacional incluido en Laravel, que facilita la interacción con bases de datos utilizando objetos y clases en lugar de consultas SQL directas. Eloquent sigue el patrón de diseño Active Record, permitiendo a los desarrolladores trabajar con registros de la base de datos como si fueran objetos de PHP [31].

#### 1.2.5.4. Controladores, rutas y middlewares.

Los controladores son clases que manejan la lógica de la aplicación y se encargan de procesar las solicitudes del usuario. Los controladores son una parte fundamental de la arquitectura MVC de Laravel, lo que permitirá desarrollar aplicaciones de manera estructurada y escalable [35].

Las rutas en Laravel son una herramienta fundamental para el desarrollo de aplicaciones web. Las rutas permiten definir las URLs de una aplicación y mapearlas a controladores que procesan las solicitudes del usuario [36].

Los middlewares proporcionan un mecanismo conveniente para inspeccionar y filtrar las solicitudes HTTP que entran en tu aplicación. Laravel incluye un middleware que verifica si el usuario de tu aplicación está autenticado. Si el usuario no está autenticado, el middleware redirigirá al usuario a la pantalla de inicio de sesión de tu aplicación. Sin embargo, si el usuario está autenticado, el middleware permitirá que la solicitud avance más en la aplicación [37].

#### 1.2.5.5. Validaciones, peticiones y respuesta en Laravel.

La validación de datos es fundamental para cualquier empresa que gestiona entradas de usuarios, transacciones o datos confidenciales. Puede considerarse como un guardián que garantiza la integridad, precisión y seguridad de los datos que ingresan a sus sistemas. Sin una validación adecuada, es probable que corra el riesgo de procesar datos incorrectos o maliciosos. Esto podría provocar ineficiencias operativas, brechas de seguridad e incluso pérdidas financieras en su empresa [38]. Laravel proporciona varias reglas de validación listas para usar cuando los usuarios de tu aplicación envían datos a través de formularios. Se pueden marcar los campos de entrada como obligatorios,

establecer una longitud mínima o máxima, y exigir entradas únicas (no duplicadas) y direcciones de correo electrónico válidas. El validador de Laravel comprueba si la entrada cumple estas reglas o cualquier otra que especifiques [39].

Un request HTTP, también conocido como petición o solicitud HTTP, es una consulta que se hace a tu backend para ejecutar una acción a nombre de un usuario [40]. En Laravel, una solicitud permite recuperar los datos de entrada enviados dentro de un formulario. Se presenta en formato de solicitud HTTP. Los datos de la solicitud suelen contener los campos del modelo utilizado para recopilar los datos [41].

La clase Response de Laravel configura la salida de tu aplicación web al ser devuelta al cliente. Representa la respuesta HTTP completa, incluyendo encabezados, contenido y códigos de estado, generada al procesar una solicitud [42].

Todas las rutas y controladores deben devolver una respuesta que se enviará de vuelta al navegador del usuario. Laravel proporciona varias formas diferentes de devolver respuestas, la respuesta más básica es devolver una cadena desde una ruta o controlador [43].

#### 1.2.5.6. Seguridad: autenticación, autorización, protección contra

ataques de falsificación de solicitudes entre sitios (CSRF).

Muchas aplicaciones web ofrecen una forma para que sus usuarios se autentiquen con la aplicación e "inicien sesión". Implementar esta función en aplicaciones web puede ser una tarea compleja y potencialmente arriesgada. Por esta razón, Laravel se esfuerza por proporcionarte las herramientas necesarias para implementar la autenticación de manera rápida, segura y fácil. Es así como, Laravel incluye servicios integrados de autenticación y sesión a los que generalmente se accede mediante las fachadas Auth y Session [44].

Las falsificaciones de solicitudes entre sitios son un tipo de exploit malicioso

mediante el cual se ejecutan comandos no autorizados en nombre de un usuario autenticado. Laravel genera automáticamente un token CSRF para cada sesión de usuario activa administrada por la aplicación. Este token se utiliza para verificar que el usuario autenticado es quien realiza las solicitudes a la aplicación. Dado que este token se almacena en la sesión del usuario y cambia cada vez que esta se regenera, una aplicación maliciosa no puede acceder a él [45].

### **1.2.6. Comparación de frameworks:**

#### **1.2.6.1. Concepto de frameworks**

La palabra Framework, se puede traducir como “marco de trabajo”, a partir de esta mera traducción podemos decir que el framework es un esquema o patrón tecnológico que se usa en el entorno informático y contiene herramientas y librerías para el desarrollo o la implementación de una nueva aplicación o web, además también permite mantener un orden concreto y preciso en el desarrollo de estas [20].

#### **1.2.6.2. Comparativa de frameworks para el frontend: Vue.js, React y Angular**

A continuación, se presenta un cuadro comparativo de los frameworks para el frontend: Vue.js, React y Angular:

**Tabla 1. Comparativa de frameworks frontend**

Framework	Ventajas	Desventajas
<b>Ligereza y rendimiento:</b>		
	una característica muy significativa subrayada por muchos Vue.js Desarrolladores es su reducido tamaño y rendimiento. El paquete principal comprimido sólo pesa 18 KB. El framework está diseñado para ser eficiente sin optimizaciones adicionales - la mayoría de las APIs de Vue son tree-shakable [46].	<b>Barreras lingüísticas</b> Uno de los problemas que su mayor comunidad se encuentra en China, lo que puede resultar problemático en el caso de algunos paquetes o investigaciones [46].
		<b>Riesgo de exceso de flexibilidad:</b> Vue ofrece una gran flexibilidad, lo que ofrece mucha libertad a la hora de utilizar distintos enfoques. Los grandes proyectos pueden acarrear problemas de incoherencia del código [46].
Vue.js	<b>Simplicidad y curva de aprendizaje:</b> Vue es fácil de entender y asimilar. La curva de aprendizaje es menos pronunciada y los conceptos de desarrollo son fáciles de entender [46].	
	<b>Integración sencilla:</b> Vue es fácil de integrar, a diferencia de otros Marcos JavaScript - Puede utilizarse no sólo para crear aplicaciones, sino también para incluir componentes en aplicaciones existentes [46].	<b>Menor número de componentes y plugins:</b> En comparación con el React, Vue.js desarrolladores puede sentir un poco la falta de plugins y librerías [46].
	<b>Reactividad:</b> Vue 3 introduce una nueva característica -	

---

API de Composición, que es similar a React Hooks. Vuex ya no es necesario, ya que Composables puede gestionar la gestión del estado [46].

---

	<b>Falta de</b>
	<b>documentación oficial</b>
<b>Fácil de aprender:</b> Sintaxis intuitiva basada en HTML. Ideal para quienes ya conocen JavaScript	<b>unificada:</b> El rápido desarrollo y colaboración abierta pueden dificultar encontrar guías completas [47].
<b>Alta flexibilidad y rendimiento:</b> React se integra fácilmente con otras librerías o frameworks y responde bien bajo carga [47].	<b>Curva de aprendizaje media:</b> Aunque la sintaxis es simple, dominar React requiere comprensión profunda de su ecosistema [47].
<b>DOM virtual eficiente:</b> Mejora el rendimiento al minimizar cambios innecesarios en el DOM real [47].	<b>Demasiada libertad estructural:</b> La ausencia de un patrón de proyecto predeterminado obliga al equipo a definir muchas decisiones arquitectónicas [47].
<b>Componentes reutilizables:</b> Facilita el desarrollo modular, mantenimiento y escalabilidad de grandes aplicaciones [47].	<b>Necesidad de integración con otras tecnologías:</b> React solo
<b>Biblioteca de código abierto:</b> Comunidad activa con actualizaciones constantes [47].	

---

gestiona la UI, por lo que necesitarás otras herramientas para el routing, manejo de estado o APIs [47].

---

	<p><b>Marco integral:</b> Angular es un framework completo con amplias funciones para crear aplicaciones empresariales [48].</p> <p><b>Enlace de datos bidireccional:</b> Angular ofrece enlace de datos bidireccional, lo que significa que cuando un desarrollador modifica algo en el modelo, la vista se sincroniza en tiempo real [48].</p> <p><b>Arquitectura modular:</b> Una de las ventajas de usar un framework modular como Angular es el acceso a componentes de código reutilizables [48].</p> <p><b>Comunidad fuerte y apoyo:</b> Angular ha formado una gran comunidad de desarrolladores a lo largo de los años [48].</p> <p><b>Inyección de dependencia:</b> Angular ofrece una forma integrada de injectar dependencias en el código [48].</p>	<p><b>Curva de aprendizaje pronunciada:</b> Angular no es un framework apto para principiantes. Su complejidad, incluyendo características como la inyección de dependencias, desalienta a algunos usuarios [48].</p> <p><b>Código verboso y complejo:</b> Angular no se caracteriza por ser conciso. Algunas funciones pueden requerir varias capas de código, incluyendo varios componentes y dependencias [48].</p> <p><b>Problemas de rendimiento en aplicaciones grandes:</b> Aunque Angular promociona su enfoque centrado en el rendimiento, las aplicaciones complejas requieren una optimización</p>
--	--	---

---

<p><b>Integración con TypeScript:</b> Una de las características más potentes de Angular es su profunda integración con TypeScript, un superconjunto de JavaScript con tipado estático desarrollado por Microsoft [48].</p> <p><b>Alto rendimiento:</b> Angular ofrece varias características que garantizan un rendimiento excelente [48].</p> <p><b>Entorno de pruebas robusto:</b> Angular simplifica el proceso de control de calidad con herramientas integradas para pruebas unitarias y de extremo a extremo [48].</p> <p><b>Mejora continua:</b> Angular está en constante evolución, lo que significa que los desarrolladores pueden contar con nuevas funciones y mejoras [48].</p>	<p>cuidadosa [48].</p> <p><b>Actualizaciones y cambios frecuentes:</b> Angular lanza actualizaciones importantes según un calendario establecido cada seis meses, además de cambios periódicos menores [48].</p> <p><b>Marco de opinión:</b> Angular suele describirse como "estricto", lo que exige que los desarrolladores se ajusten a patrones de desarrollo específicos [48].</p> <p><b>Desafíos del SEO:</b> Debido a la naturaleza de las aplicaciones de una sola página, Angular presenta desafíos específicos para el SEO [48].</p>
---	---

---

Como se puede observar en Tabla 1. Comparativa de frameworks frontend, Vue.js se destaca por su ligereza, simplicidad y alto rendimiento, cualidades que permiten construir interfaces dinámicas y reactivas con un consumo mínimo de recursos. Su núcleo comprimido apenas pesa 18 KB, lo que se traduce en una carga rápida y eficiente incluso

en dispositivos de baja capacidad, algo esencial para plataformas web institucionales que buscan accesibilidad y compatibilidad con diferentes entornos.

A diferencia de Angular, cuyo enfoque integral implica una curva de aprendizaje pronunciada y una estructura de código extensa, Vue.js ofrece una curva de aprendizaje mucho más suave, lo que facilita que tanto desarrolladores experimentados como principiantes puedan adaptarse rápidamente al entorno y comenzar a producir resultados en menos tiempo. Su sintaxis clara y su integración sencilla hacen que Vue.js sea ideal para equipos pequeños o medianos, reduciendo la complejidad del desarrollo sin sacrificar la escalabilidad del sistema.

Por otro lado, si bien React ofrece un excelente rendimiento y una comunidad muy activa, su excesiva libertad estructural puede convertirse en una desventaja, ya que obliga a definir manualmente muchas decisiones arquitectónicas y depender de librerías externas para tareas básicas como el enrutamiento o la gestión del estado. Vue.js, en cambio, ofrece una solución más equilibrada al integrar herramientas oficiales como Vue Router y Vuex, brindando una estructura organizada y coherente que favorece el mantenimiento y la evolución del proyecto a largo plazo.

Otra ventaja importante es su naturaleza progresiva, lo que permite usar Vue tanto para añadir componentes específicos en aplicaciones existentes como para desarrollar proyectos completos desde cero. Además, la introducción de la Composition API en Vue 3 proporciona un manejo más moderno y flexible del estado y la lógica de los componentes, equiparándose a las capacidades de React Hooks, pero manteniendo una sintaxis más intuitiva.

A nivel de comunidad, aunque Vue.js tiene una mayor presencia en Asia, su adopción global ha crecido de manera significativa, y hoy cuenta con amplia documentación en múltiples idiomas, así como con una comunidad activa y numerosos paquetes open source que enriquecen su ecosistema.

Tras analizar cada una de estas ventajas y desventajas de los frameworks Vue.js, React y Angular, se determinó que Vue.js es la opción más adecuada para el desarrollo del frontend del presente proyecto de tesis.

#### 1.2.6.3. Comparativa de frameworks para el backend: Laravel,

#### Django y Spring

A continuación, se presenta un cuadro comparativo de los frameworks para el frontend: Vue.js, React y Angular:

**Tabla 2. Comparativa de frameworks backend**

Framework	Ventajas	Desventajas
	<p><b>Fácil de entender:</b> ya que los documentos son muy legibles y están bien explicados, este framework es de fácil aprendizaje [20].</p> <p><b>Uso de MVC framework:</b> evita las arquitecturas tradicionales donde los desarrolladores solían escribir todo en HTML y código PHP en el mismo fichero [20].</p> <p><b>Reducción en el ciclo de desarrollo de un producto:</b> minimiza considerablemente este ciclo gracias a</p>	<p><b>Desempeño lento:</b> necesita un gran número de consultas a la base de datos, lo cual ralentiza su funcionamiento</p> <p>[20].</p> <p><b>No admite la función de pagos:</b> debido a que tendría que seguir una regulación y cumplimiento del PCI (Payment Card Industry), es preferible diferir esta función a servicios como Paypal [20].</p>
Laravel		

	<p>que las integraciones son más rápidas [20].</p>	<p><b>A menudo no proporciona riqueza a aplicaciones móviles:</b> al tener que recargar páginas completamente, en aplicaciones no es lo más recomendable, pero para páginas web no sería tan pesado [20].</p>
Django	<p><b>Rápido:</b> ha sido desarrollado para ayudar a los desarrolladores a hacer una aplicación lo más rápido posible.</p> <p><b>Herramientas para todo:</b> funciona de tal manera que incluye docenas de extras para ayudar al desarrollador con varias funcionalidades como la autenticación del usuario, mapas de situaciones, contenido para la administración, etc. [20].</p> <p><b>Segura:</b> el desarrollador no tiene que preocuparse de cometer errores relacionados con la seguridad, ya que, por ejemplo, para garantizar el manejo de usuarios y contraseñas está la autenticación del usuario mencionado anteriormente [20].</p>	<p>tiene una estructura muy fija, te dirige hacia patrones que vienen dados por el framework, por lo que no da soltura a la creatividad del desarrollador.</p> <p><b>Capacidad de manejar más de una solicitud a la vez:</b> para páginas web, suele ser la tarea más necesaria y Django no está diseñado para ello [20].</p> <p><b>Incompatibilidad para actualizar versiones frente a la velocidad:</b> tiene tendencia a ser cada vez más grande y pesada, por lo que la velocidad es importante, pero Django está más enfocado en la compatibilidad de versiones anteriores y no tanto en la velocidad [20].</p>
Spring	<p><b>Uso de POJO (Plain Old Java Object):</b> usar POJO para desarrollar</p>	<p><b>Complejidad:</b> como tiene muchas clases y</p>

---

<p>aplicaciones tiene el beneficio de no necesitar un servidor para que funcione [20].</p> <p><b>No necesita ser reinventado:</b> al usar tecnologías conocidas como JDK timers o Java EE entre otros [20].</p> <p><b>Framework de desarrollo bien diseñado:</b> Spring está diseñado por un modelo MVC, el cual puede servir para que otro framework web lo use [20].</p>	<p>herramientas distintas, Spring puede llegar a ser un framework difícil de aprender [20].</p> <p><b>Muchos mecanismos de paralelismo:</b> el paralelismo puede ser de gran ayuda si el desarrollador conoce a la perfección qué métodos y clases serían útiles para un caso en particular [20].</p> <p><b>Gran cantidad de XML:</b> para trabajar con Spring, ha de tener soltura con XML, así que, si se quiere desarrollar una aplicación con dicho framework, hay que estar preparado para los XML [20].</p>
--	---

---

Como se vio en la Tabla 2. Comparativa de frameworks backend, Laravel se destaca por su sencillez y facilidad de aprendizaje, gracias a su documentación clara y extensa, así como a su gran comunidad activa de desarrolladores. Esto representa una ventaja significativa a Spring, cuyo ecosistema, aunque robusto, puede resultar complejo por su curva de aprendizaje elevada y la necesidad de manejar numerosos archivos de configuración XML. A diferencia de ello, Laravel permite una configuración más simple y directa, facilitando el proceso de desarrollo y reduciendo el tiempo de implementación.

Por otro lado, si bien Django ofrece rapidez y muchas herramientas integradas, su estructura rígida limita la flexibilidad del desarrollador y restringe la libertad para definir la arquitectura del sistema según las necesidades del proyecto. En cambio, Laravel basado en

patrón MVC, permite una organización clara del código y una gran libertad de personalización, lo que favorece la escalabilidad y mantenimiento del sistema a largo plazo.

Además, Laravel incorpora múltiples características que optimizan la productividad del desarrollador, como lo es Eloquent ORM para la gestión intuitiva de bases de datos Blade como motor de plantillas dinámicas y Artisan, una poderosa herramienta de línea de comando que automatiza tareas comunes. Estas funcionalidades no solo agilizan el desarrollo, sino que también garantizan una estructura sólida y coherente en el proyecto.

Por eso es que, al analizar las ventajas y desventajas de los frameworks Spring, Django y Laravel, se determinó que Laravel es la opción más adecuada para el desarrollo del presente trabajo de tesis.

#### 1.2.6.4. Laragon

Laragon es un entorno de desarrollo local ligero compatible con múltiples versiones de PHP, Apache, Nginx, MySQL y más. Laragon Portable es una versión simplificada que incluye PHP 5.4, MySQL 5.1 y bitman, y permite añadir versiones más recientes de PHP y MySQL posteriormente. Laragon es un proyecto de código abierto y ambas versiones están disponibles de forma gratuita [49].

### **1.2.7. Diseño de Base de datos y migraciones**

#### **1.2.7.1. MySQL**

MySQL es un sistema open source de administración de bases de datos que es desarrollado y soportado por Oracle [50].

MySQL se ha ganado su lugar en el mundo digital como una base de datos relacional que permite almacenar, organizar y recuperar datos de manera eficiente. MySQL es utilizado por una amplia variedad de organizaciones y aplicaciones en todo el mundo [51].

#### **1.2.7.2. Modelado entidad-relación.**

En el artículo de [52], describe al modelo entidad-relación como un método para la representación gráfica de datos y sus relaciones dentro de un sistema, el cual se utiliza para planificar la estructura lógica de las bases de datos antes de su implementación técnica.

Entidades: Representan objetos o conceptos importantes en un sistema, como clientes, productos o pedidos. Suelen representarse como rectángulos [52].

Atributos: Los atributos describen las propiedades de una entidad, como el nombre de un cliente o el precio de un producto. Se visualizan mediante puntos suspensivos junto a la entidad correspondiente [52].

Relaciones: Las relaciones muestran la conexión entre dos o más entidades, como un cliente y un pedido. Estos vínculos se representan mediante rombos [52].

#### **1.2.7.3. Migraciones en Laravel.**

Las migraciones son como un control de versiones para su base de datos, permitiendo a su equipo definir y compartir la definición del esquema de la base de datos de la aplicación. La Schema fachada de Laravel proporciona soporte independiente de la base de datos para crear y manipular tablas en todos los sistemas de bases de datos compatibles con Laravel. Normalmente, las migraciones utilizan esta fachada para crear y modificar tablas y columnas de la base de datos [53].

### **1.2.8. Control de versiones Git y GitHub**

#### **1.2.8.1. Git**

Git es un sistema de control de versiones distribuido, creado en 2005 por Linus Torvalds, el mismo desarrollador detrás de Linux. Su principal objetivo es permitir a los equipos trabajar de forma eficiente en proyectos de software, controlando los cambios en el código y facilitando la colaboración [54].

#### **1.2.8.2. GitHub.**

GitHub es una plataforma en la nube que almacena y gestiona repositorios Git. Fue fundada en 2008 y adquirida por Microsoft en 2018. Su propósito es simplificar la colaboración entre desarrolladores y mejorar la calidad del software a través de herramientas como pull requests, gestión de issues y automatización de despliegues [54].

### **1.2.9. Despliegues y pruebas**

#### **1.2.9.1. Servidores Linux.**

Un servidor Linux es un servidor que ejecuta su propio sistema operativo (SO) de código abierto Linux, un SO popular creado para algunas de las aplicaciones empresariales más exigentes del mundo [55].

#### 1.2.9.2. AlmaLinux 9.

AlmaLinux 9 es la nueva versión mayor de la que se ha convertido en la sucesora de CentOS y derivada de Red Hat Enterprise Linux (RHEL) más audaz de todas las que ya había, pero también de las que surgieron tras el ocaso de la vertiente comunitaria. En resumen, AlmaLinux 9 recoge de RHEL 9 todas las actualizaciones de esta, incluyendo las optimizaciones de rendimiento y mejoras de seguridad aplicadas, además de las nuevas versiones de componentes como lenguajes de programación, frameworks y otras herramientas enfocadas en el desarrollo [56].

#### 1.2.9.3. SSH.

El protocolo SSH permite que dos computadoras establezcan una conexión segura y directa dentro de una red potencialmente insegura como Internet. Este protocolo es necesario para que no puedan acceder terceros al flujo de datos que se transfiere a través de la conexión y la información sensible no caiga en manos equivocadas. Antes de que apareciera Secure Shell, también era posible establecer conexiones directas entre dos ordenadores, pero las aplicaciones que se utilizaban, Telnet, Remote Shell o rlogin, eran inseguras. SSH encripta la conexión entre dos computadoras y permite operar una de ellas desde la otra [57].

#### 1.2.9.4. Apache.

El servidor web Apache es un servidor Web gratuito desarrollado por el Apache Server Project (Proyecto Servidor Apache) cuyo objetivo es la creación de un servidor web fiable, eficiente y fácilmente extensible con código fuente abierto gratuito. Este proyecto es conjuntamente manejado por un grupo de voluntarios localizados alrededor del mundo que a través de Internet planean y desarrollan el servidor y la documentación relacionada con éste [58].

### **1.3. Definición de términos básicos**

#### **1.3.1. Tecnologías y herramientas**

PHP 8: Lenguaje de programación del lado del servidor, ampliamente utilizado para el desarrollo web debido a su flexibilidad, compatibilidad con múltiples sistemas y gran comunidad de soporte.

Laravel 10: Framework backend basado en PHP que emplea el patrón Modelo-Vista-Controlador (MVC).

Vue.js 3: Framework progresivo de JavaScript enfocado en la construcción de interfaces de usuario reactivas y dinámicas.

MySQL 8: Sistema de gestión de bases de datos relacional de código abierto que permite almacenar, organizar y consultar datos de manera eficiente.

Git: Sistema de control de versiones distribuido que permite registrar los cambios en el código fuente y facilitar el trabajo colaborativo entre desarrolladores.

GitHub: Plataforma basada en la nube que permite alojar repositorios Git, gestionar proyectos y automatizar procesos de desarrollo y despliegue.

Bootstrap 5: Framework de código abierto que proporciona componentes prediseñados en HTML, CSS y JavaScript, permitiendo construir interfaces modernas, responsivas y visualmente atractivas.

#### **1.3.2. Componentes del Sistema**

Gestión del Curriculum Vitae Normalizado (CVN):

Módulo que permite a los usuarios registrar, editar y generar su currículum en un formato estandarizado, garantizando la interoperabilidad con otras plataformas académicas y cumpliendo con estándares institucionales.

Bolsa de empleo:

Funcionalidad que conecta a graduados con empleadores, facilitando la publicación, búsqueda y postulación a ofertas laborales o de emprendimiento.

Gestión de usuarios: Módulo encargado de administrar los diferentes tipos de usuarios del sistema (estudiantes, graduados, empresas, administradores), asignando roles y permisos específicos.

Módulo de autenticación: Sistema de registro e inicio de sesión desarrollado en Laravel, que garantiza la seguridad de acceso y la validación de credenciales de los usuarios.

Módulo de reportes: Componente que genera informes sobre el estado de empleabilidad, número de ofertas publicadas, postulaciones y registros de CV, permitiendo la toma de decisiones basada en datos.

### **1.3.3. Arquitectura y comunicación**

Arquitectura cliente-servidor:

Modelo en el cual el cliente (frontend) solicita servicios o datos al servidor (backend) a través de una red utilizando protocolos de comunicación como HTTP o TCP/IP.

API RESTful: Interfaz que permite la comunicación entre sistemas mediante solicitudes HTTP, siguiendo principios de transferencia de estado representacional (REST). Facilita la interoperabilidad entre frontend y backend.

Comunicación frontend-backend: Proceso en el que el cliente (Vue.js) realiza peticiones HTTP al servidor (Laravel) a través de una API REST, recibiendo datos en formato JSON para ser representados dinámicamente en la interfaz del usuario.

## **CAPÍTULO II. METODOLOGÍA**

### **2.1. Diseño metodológico**

El presente proyecto adopta un diseño de tipo descriptivo-experimental, ya que busca desarrollar dos plataformas web funcionales, una para la gestión del Curriculum Vitae Normalizado y otra para la Bolsa de Empleo Universitaria que respondan a los requerimientos y necesidades de los estudiantes y graduados de la UTLVTE.

Asimismo, se aplica un enfoque cualitativo de tipo descriptivo, orientado a comprender la percepción y experiencia de los usuarios frente al uso de las plataformas web desarrolladas para la gestión del CVN y Bolsa de Empleo.

Para el desarrollo de ambas plataformas se utilizó la metodología de Programación Extrema (XP), por su orientación a la calidad del software, la comunicación constante con el usuario y la entrega rápida de versiones funcionales. Esta metodología fue seleccionada debido a su capacidad para adaptarse a requerimientos cambiantes y asegurar que el producto final responda a las expectativas reales de la comunidad universitaria.

Este proceso metodológico se estructuró en cinco etapas principales:

#### **Fase 1. Exploración**

Para esta etapa se recopilaron los requerimientos funcionales y no funcionales de las plataformas, tomando como base las encuestas realizadas a estudiantes y el reporte de evaluación del CASES en 2023. Se lograron analizar las necesidades de los distintos actores, entre ellos: estudiantes, egresados y graduados. A continuación, se presenta una lista de los requerimientos funcionales y no funcionales:

**Tabla 3. Requerimientos funcionales**

N.º	Requerimiento	Descripción
01	RF- Obtención de información personal de estudiantes desde la base de datos institucional	El sistema debe obtener datos de los estudiantes y graduados desde la base de datos institucional
01	RF- Inicio de sesión y autenticación	Los usuarios deben poder iniciar sesión utilizando credenciales validadas en el backend de Laravel.
02	RF- Gestión de perfiles	Cada usuario debe poder visualizar y editar su información personal, académica y profesional.
03	RF- Creación de Curriculum Vitae Normalizado (CVN)	El sistema debe permitir a los usuarios ingresar sus datos personales, formación académica, experiencia, habilidades y generar el CV en formato estándar institucional PDF
04	RF- Registro y publicación de emprendimientos	El sistema debe permitir a los usuarios registrar y publicar emprendimientos
05	RF- Validación de campos	Los formularios deben validar campos obligatorios antes de permitir el guardado.
06	RF- Administración de emprendimientos	El administrador del sistema debe aprobar o rechazar emprendimientos
07	RF- Publicación de ofertas laborales	Las empresas y estudiantes registrados deben poder publicar, editar e inhabilitar ofertas laborales o de emprendimiento.
08	RF- Búsqueda y filtrado de ofertas	Los usuarios deben poder buscar y filtrar ofertas por categoría o por vigencia
09	RF- Postulación a ofertas	Los graduados deben poder postularse a las ofertas publicadas, siempre y cuando posean un CVN

	RF-	Visualización de	Las empresas deben poder consultar los
10		postulaciones	postulantes registrados en sus ofertas.
			El administrador debe poder generar
	RF-	Gestión institucional	reportes estadísticos de emprendimientos, ofertas
11			y postulaciones para fines de seguimiento y
			vinculación
			El sistema debe notificar al usuario
	RF-	Notificaciones	mediante correo electrónico o alertas internas
12			sobre acciones relevantes (registro, postulación, aceptación, etc.).
	RF-		Las plataformas CVN y Bolsa de
13		Integración entre plataformas	Empleo deben interoperar mediante API REST, compartiendo datos de usuarios y CVs

**Tabla 4. Requerimientos no funcionales**

<b>N.º</b>	<b>Requerimiento</b>	<b>Descripción</b>
01	RNF-Rendimiento	El sistema debe responder a las peticiones de los usuarios en menos de 3 segundos bajo carga moderada.
01	RNF-Escalabilidad	La arquitectura debe permitir el crecimiento de la base de datos y el aumento de usuarios sin afectar el rendimiento.
02	RNF-Seguridad	Las contraseñas deben almacenarse cifradas y las peticiones autenticadas mediante tokens (JWT).
03	RNF-Usabilidad	Las interfaces deben ser intuitivas, responsivas y accesibles desde cualquier dispositivo.
04	RNF-Mantenibilidad	El código debe estar estructurado siguiendo el patrón MVC para facilitar futuras actualizaciones.
05	RNF-Portabilidad	El sistema debe poder implementarse en diferentes servidores Apache o Nginx sin cambios mayores.
06	RNF-Compatibilidad	Las plataformas deben ser compatibles con navegadores modernos (Chrome, Firefox, Edge).
07	RNF-Disponibilidad	El sistema debe garantizar una disponibilidad del 99% en entorno de producción.
08	RNF-Interoperabilidad	Los sistemas CVN y Bolsa de Empleo deben intercambiar información mediante servicios API RESTful estandarizados.
09	RNF-Postulación a ofertas	Los graduados deben poder postularse a las ofertas publicadas, siempre y cuando posean un CVN
10	RNF-Seguridad de datos	Debe cumplirse la confidencialidad y protección de datos personales conforme a la normativa vigente.
11	RNF-Control de versiones	El desarrollo debe ser gestionado mediante GitHub para asegurar trazabilidad de cambios y versiones.

El sistema debe incluir documentación técnica y manual de usuario para su mantenimiento y capacitación institucional.

Con estos requerimientos obtenidos, se muestra a continuación el diagrama Unified Modeling Language (UML), la cual representa las funcionalidades de los usuarios dependiendo su rol.

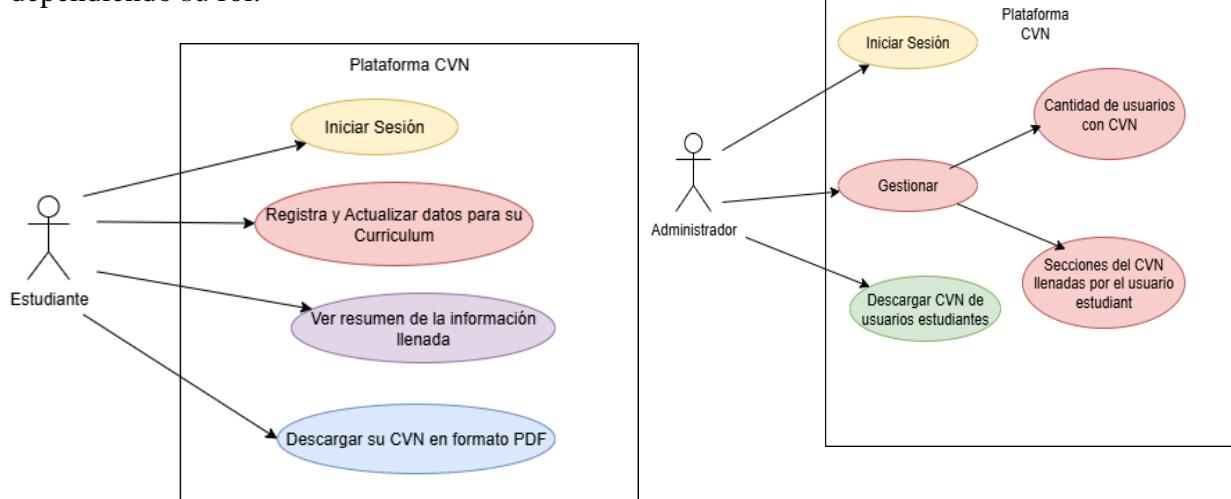
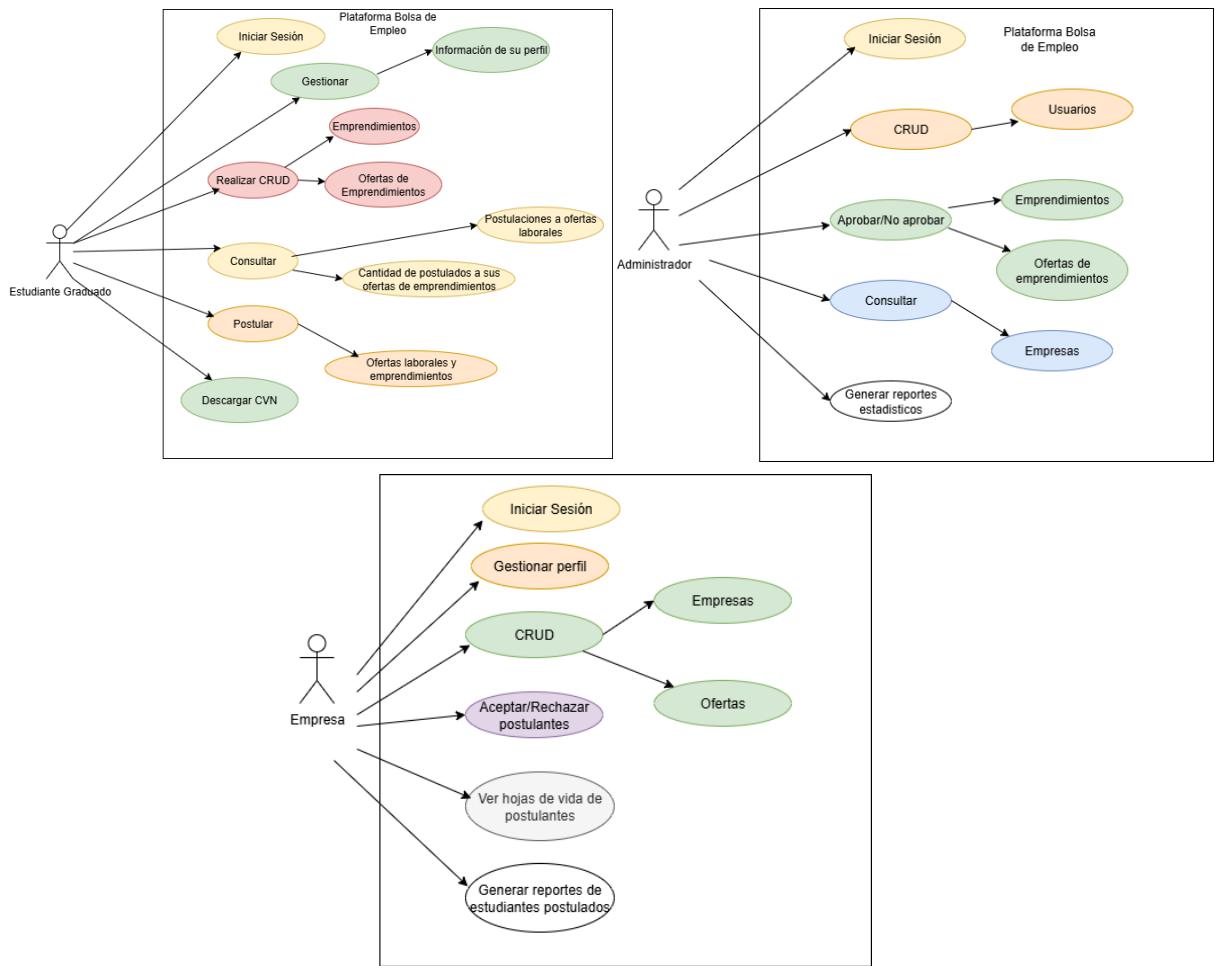


Figura 5. Diagrama UML de la plataforma CVN



**Figura 6. Diagrama UML de la plataforma Bolsa de Empleo**

## Fase 2. Planificación.

Una vez definidos los requerimientos funcionales y no funcionales, en esta etapa se priorizan las funcionalidades del sistema, tiempos de desarrollo y se organizan tareas en iteraciones o versiones incrementales, buscando cumplir con los objetivos establecidos.

### Priorización de requerimientos.

A partir de los requerimientos funcionales, se realizó una clasificación por prioridad, donde se consideraron tres niveles: Alto, Medio y Bajo

**Tabla 5. Priorización de Requerimientos**

Nivel	Módulos priorizados	Descripción
Alto	Inicio de sesión, gestión de usuarios, obtención de datos de estudiantes, creación del CVN, creación y publicación de emprendimientos y ofertas, postulación a empleos y a emprendimientos.	Funcionalidades principales para garantizar la operatividad inicial de ambas plataformas.
Medio	Generación de reportes estadísticos, notificaciones automáticas, integración CVN– Bolsa de Empleo	Mejoran la interacción, comunicación y trazabilidad entre usuarios y empresas.
Bajo	Módulos de estadísticas visuales, configuraciones personalizadas de usuario, mejoras visuales del dashboard	Optimizan la experiencia y apariencia general del sistema.

### **Definición de tarea e iteraciones.**

Para este punto, el desarrollo se estructuró en iteraciones cortas, siguiendo el enfoque incremental de XP. Cada iteración permitió obtener una versión funcional del sistema, verificando su correcto funcionamiento antes de continuar con la siguiente.

La siguiente tabla resume las principales tareas de esta fase:

**Tabla 6. Actividades con iteraciones**

<b>N.<sup>o</sup></b>	<b>Iteración</b>	<b>Actividades/tareas</b>
1		Configuración de entornos de desarrollo, backend: Laravel y frontend: Vue.js, conexión con MySQL, estructura inicial de proyectos y repositorios GitHub.
2		Creación de tablas necesarias para ambas plataformas
3		Implementación del módulo de autenticación: login, roles y permisos.
4		Desarrollo de la plataforma CVN, con sus respectivos módulos: información personal, académica, profesional, habilidades, generación de CV.
5		Desarrollo de la plataforma Bolsa de Empleo, con sus respectivos módulos: publicación, búsqueda y postulación a ofertas.
6		Integración de ambas plataformas mediante API REST, pruebas de interoperabilidad y validación de datos compartidos.
7		Pruebas de rendimiento, usabilidad y validación de formularios. Manuales de usuario.

### **Fase 3. Diseño**

En esta fase se define la arquitectura del sistema, la estructura de los componentes, la interfaz de usuario y la interacción entre el frontend y el backend. Aquí se garantiza que el desarrollo del software siga un modelo coherente, mantenable y escalable, permitiendo que las funcionalidades priorizadas en la fase de planificación se implementen de manera ordenada.

#### **Diseño de la arquitectura del sistema.**

Las plataformas se diseñaron bajo el enfoque Cliente-Servidor, siguiendo una arquitectura RESTful para la comunicación entre el frontend y el backend.

La siguiente tabla muestra el componente, tecnología utilizada y su función principal para el diseño de ambas plataformas.

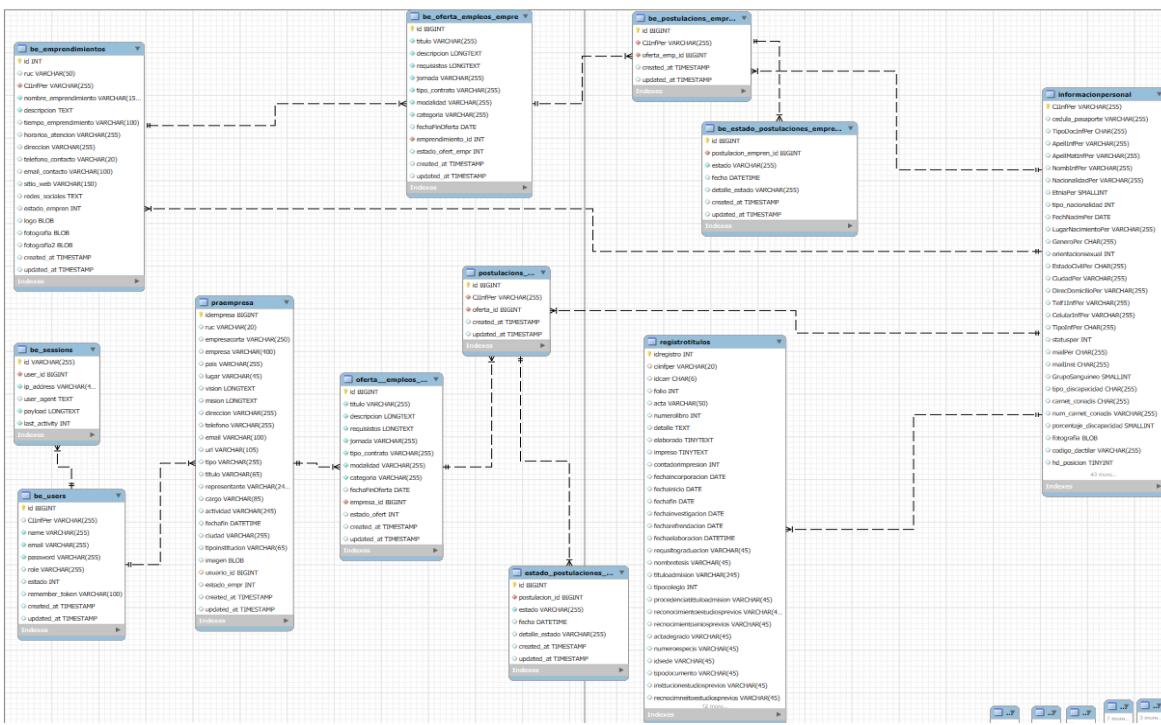
**Tabla 7. Actividades con iteraciones**

Componente	Tecnología usada	Función principal
Frontend	Vue.js 3 + Bootstrap 5	Interfaz de usuario, formularios, vistas dinámicas y manejo del estado.
Backend	Laravel 10	Gestión de usuarios, autenticación, controladores, API REST y validación de datos.
Servidores Locales	Laragon y AlmaLinux	Pruebas locales del funcionamiento de las plataformas en un servidor Apache
Base de datos	MySQL 8	Almacenamiento de información
Control de versiones	Git y GitHub	Control de cambios, commits y respaldo del código fuente.

### **Diseño de la base de datos.**

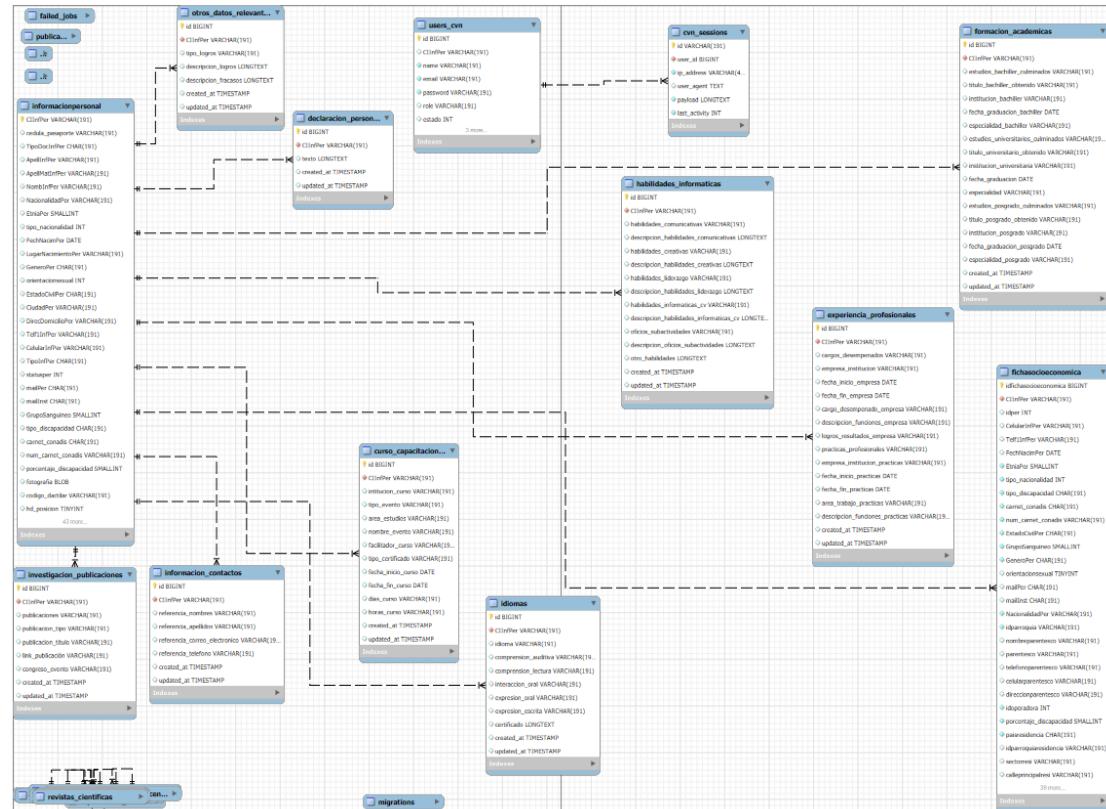
Como motor de base de datos se usa MySQL Workbench, aunque Laragon proporciona un motor de base de datos liviano (HeidiSQL). El diseño de la base de datos se realizó aplicando el modelo Entidad-Relación, garantizando la integridad y relación adecuada entre las entidades.

Cada una de las tablas fue normalizada para evitar redundancias y asegurar la



consistencia de datos.

**Figura 7. Tablas para la plataforma Bolsa de Empleo**



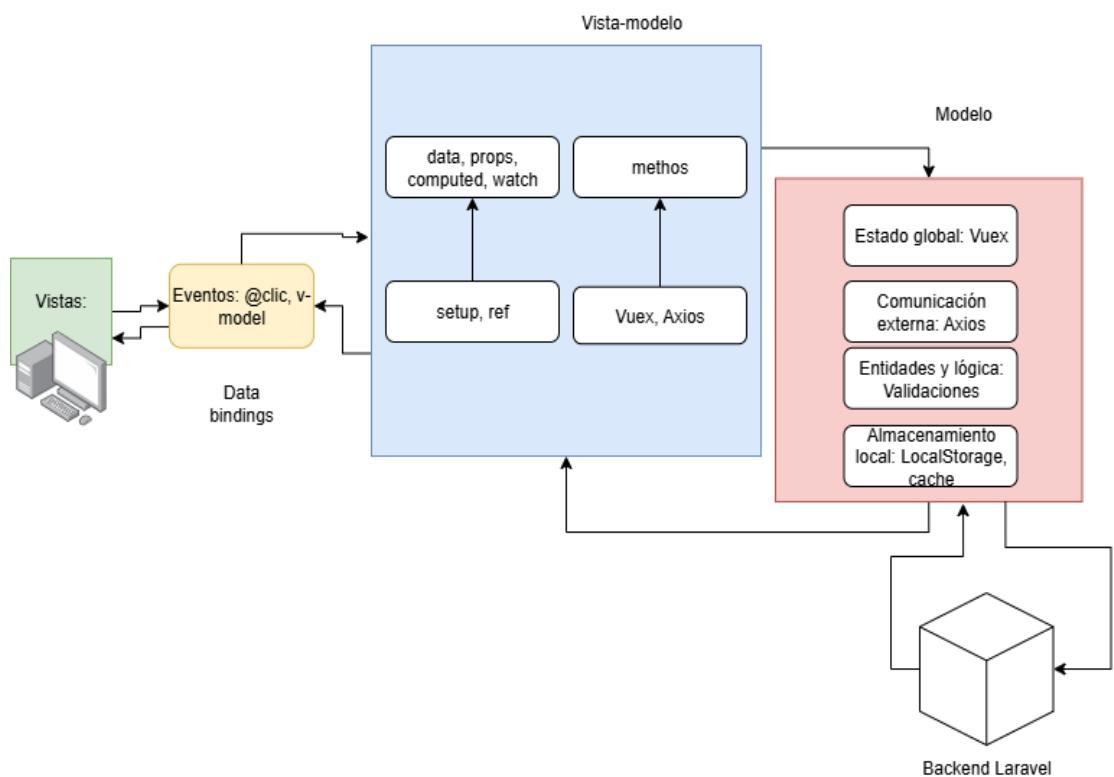
**Figura 8. Tablas para la plataforma CVN**

Estas tablas se crean automáticamente mediante migraciones en Laravel, lo cual permite versionar los cambios en la estructura de la base de datos y mantener sincronizados los entornos de desarrollo.

### Diseño del frontend.

El diseño del frontend se centró en la experiencia del usuario y la interfaz visual, aplicando principios de diseño intuitivo, minimalista y responsive.

Vue utiliza la arquitectura MVVM, esta arquitectura se utilizó para el desarrollo de ambas plataformas.



**Figura 9. Arquitectura MVVM - Vue**

**Nota:** El modelo gestiona los datos y la lógica de negocio, el vistamodelo actúa como intermediario que vincula los datos y eventos mediante enlaces reactivos, y la vista muestra la información al usuario y captura sus iteraciones, manteniendo bidireccionalidad entre las capas. Al final estas se enlazan con el backend Laravel para realizar peticiones.

## Plataforma CVN



Figura 10. Página principal

## Plataforma Bolsa de Empleo

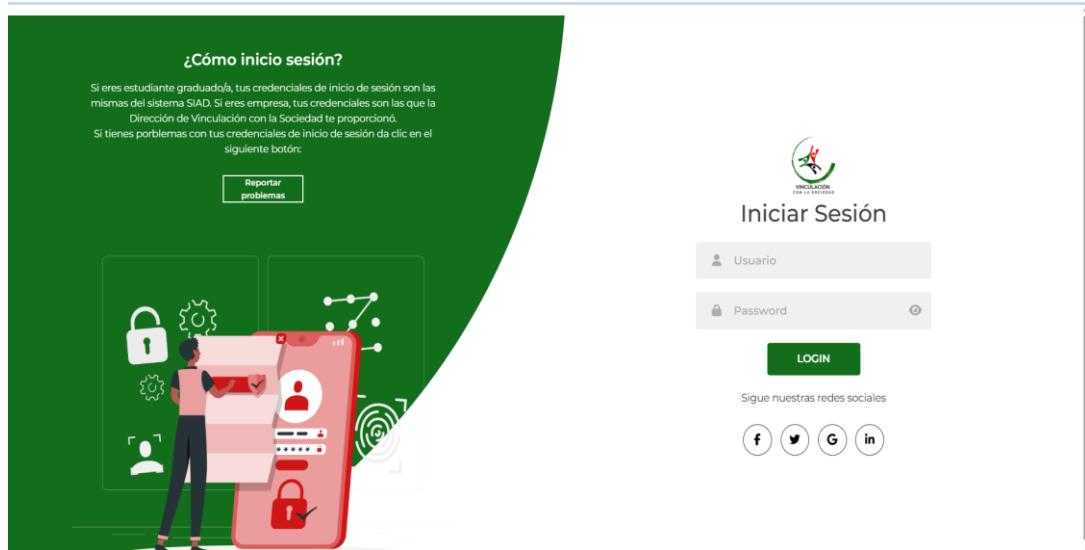


Figura 11. Página principal – Inicio de Sesión

## Diseño del backend.

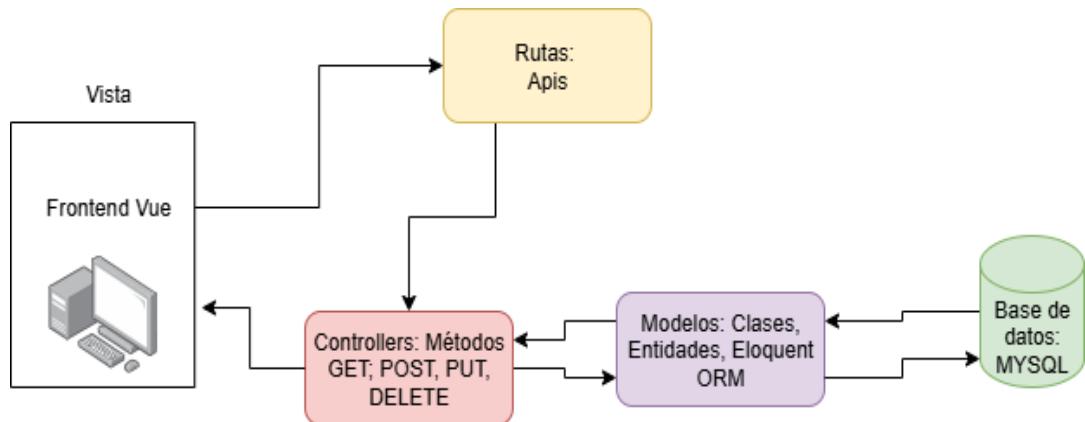


Figura 12. Arquitectura MVC - Laravel

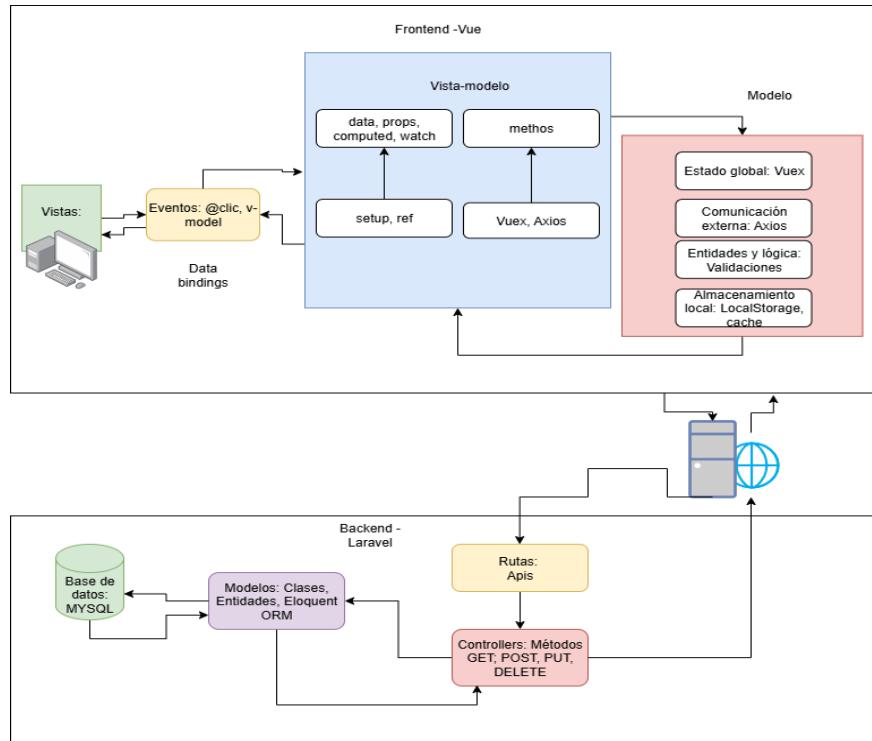
**Nota:** EL frontend interactúa con las rutas (APIS), las cuales direccionan las peticiones hacia los controladores.

Estos controladores ejecutan los métodos correspondientes, ya sean: GET, POST, PUT, DELETE. Luego el controlador se comunica con el modelo, el cual gestiona la lógica de datos y operaciones en la base de datos MySQL.

Al crear un proyecto en laravel, éste ya viene con la estructura MVC. Para cada entidad se crearon modelos y controladores para que estos puedan interactuar con la base de datos mediante Eloquent ORM, y rutas API que definen los puntos de acceso al frontend.

Para permitir la interoperabilidad entre ambas plataformas, se diseñaron Endpoints RESTful que permiten realizar peticiones HTTP como: GET, POST, PUT y DELETE, este último se utiliza, pero no es conveniente que los datos se eliminen definitivamente ya que es necesario tener respaldos, por ende, se usan estados 0 y 1 para habilitar o deshabilitar un registro.

Laravel actúa como un servidor API, mientras que Vue.js consume toda la información y la representa visualmente en el navegador.



**Figura 13. Arquitecturas MVVM (frontend) y MVC(backend)**

**Nota:** El diagrama muestra la interacción entre el frontend en Vue.js con MVVM y el backend en Laravel. Vue gestiona la interfaz, los eventos y el estado global, mientras que Laravel maneja las rutas, controladores y modelos que se comunican con la base de datos MySQL, permitiendo un flujo estructurado y modular entre la vista y los datos.

#### Fase 4. Codificación

En esta fase se materializan las estructuras, módulos y funcionalidades diseñadas en las anteriores fases. Aquí se transformaron los diagramas, modelos y arquitecturas en código ejecutable.

No se presentarán todos los códigos de todo el proyecto, sino, códigos de secciones importantes.

#### Estructura de codificación del backend:

Como se dijo en el diseño, Laravel proporciona arquitectura MVC al crear el proyecto. Cada módulo se estructuró en carpetas separadas dentro del frameworks, lo que garantiza la mantenibilidad y escalabilidad del código.

**Tabla 8. Componentes Backend**

Componente	Ubicación	Descripción
Modelos	/app/Models/	Representan las entidades de la base de datos.
Controladores	/app/Http/Controllers/	Lógica de negocio y las peticiones HTTP.
Rutas	/routes/api.php	Definen los endpoints RESTful para el consumo desde el frontend.
Migraciones	/database/migrations/	Permiten crear y modificar tablas de forma controlada
Middleware	/app/Http/Middleware	Gestionan la autenticación, roles y permisos de acceso.

## Fase 5. Pruebas

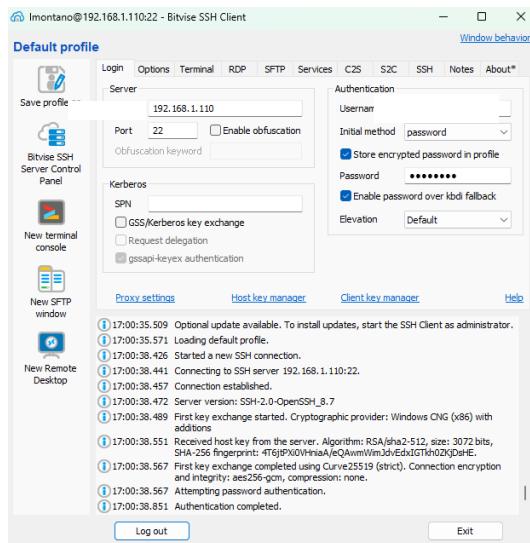
En esta fase de pruebas se logró verificar la calidad, funcionalidad, rendimiento y usabilidad del sistema desarrollado. Para ello se procedió a realizar pruebas locales desde el servidor Laragon y corregir pequeñas inconsistencias como el exceso de solicitudes al servidor. Después se procedió a subir ambas plataformas a un servidor local de AlmaLinux, este servidor usando Apache, el servidor resultó ser muy óptimo para la carga de las plataformas web, se crearon archivos de configuración para poder acceder a ambas plataformas mediante la ip del servidor y diferenciarlas con un slash, ejemplo: 192.168.50.1/cvn y 192.168.50.1/b\_e. Las pruebas unitarias, funcionales, de integración y usuario ayudaron a verificar el correcto funcionamiento de los métodos, controladores y validaciones en Laravel, validar todas las funcionalidades completas de ambas plataformas, evaluar la comunicación entre plataformas y que si un usuario de tipo estudiante no tenga un CVN no pueda postular a una oferta laboral o de emprendimiento.

Las autenticaciones de usuarios funcionaron correctamente, redireccionando/mostrando contenido de la página dependiendo del Rol del usuario

logueado.

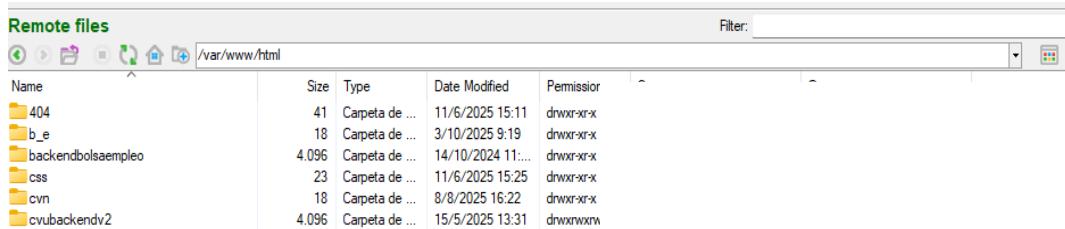
Ambas plataformas usaron algo adicional, el envío de correos. Esta funcionalidad ayuda a que los usuarios no solo sean notificados dentro de la plataforma, sino, en sus correos electrónicos, ya sean estos institucionales o personales.

Se adjuntan imágenes de las pruebas funcionales de ambas plataformas.



**Figura 14. Aplicación Bitsive SSH Client para conectar al servidor AlmaLinux**

**Nota:** La aplicación Bitsive SSH, permite establecer conexión al servidor creado



**Figura 15. Acceso a las carpetas de apache en donde estarán las plataformas**

**Nota:** En esta figura se muestran las carpetas a las cuales nuestro servidor Apache tendrá acceso.

```

# Para CVN
Alias /cvn /var/www/html/cvn/dist

<Directory /var/www/html/cvn/dist>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

# Para CVNBAK
Alias /cvubackendv2 /var/www/html/cvubackendv2/public

<Directory /var/www/html/cvubackendv2/public>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

# Para B_E BACK
Alias /b_e /var/www/html/b_e/dist

<Directory /var/www/html/b_e/dist>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

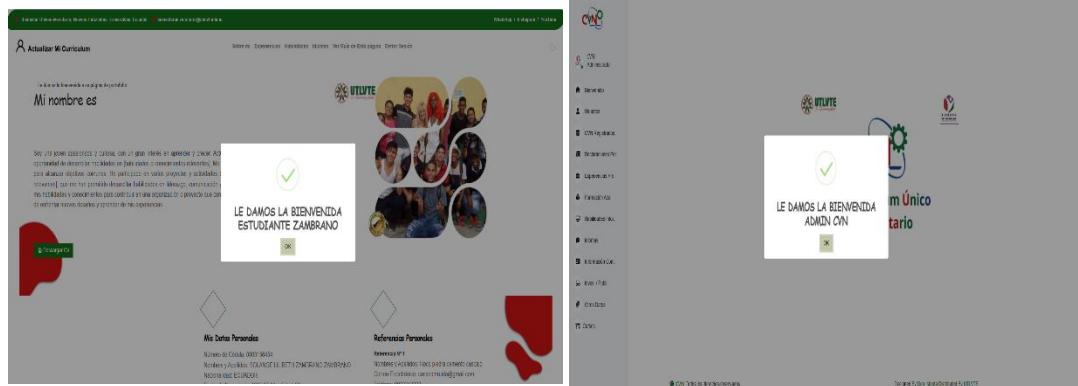
# Para Bolsa de Empleo BACK
Alias /backendbolsaempleo /var/www/html/backendbolsaempleo/public

<Directory /var/www/html/backendbolsaempleo/public>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

```

**Figura 16. Archivo de configuración para el Alias de la plataforma CVN y Bolsa de Empleo**

**Nota:** Estos archivos de configuración ayudan a que si una aplicación usa la misma dirección ip se diferencien con un alias



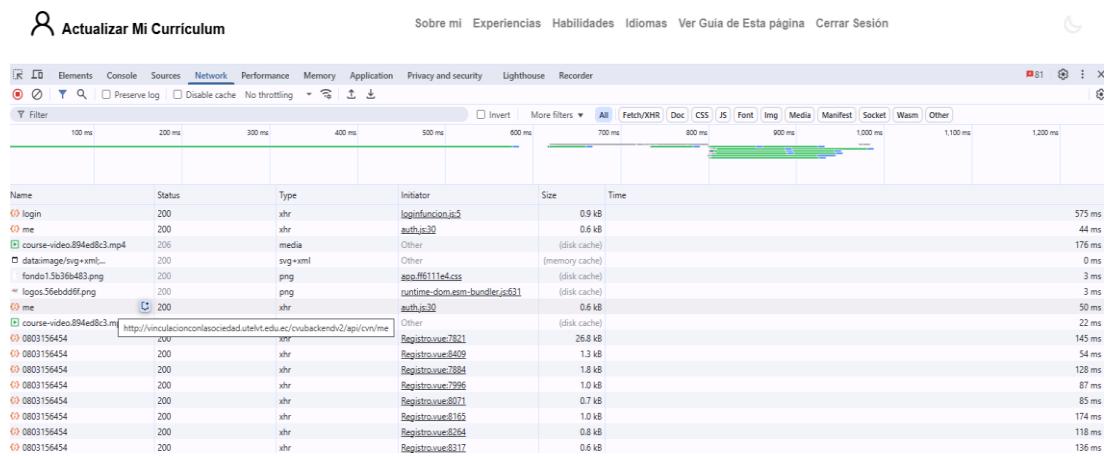
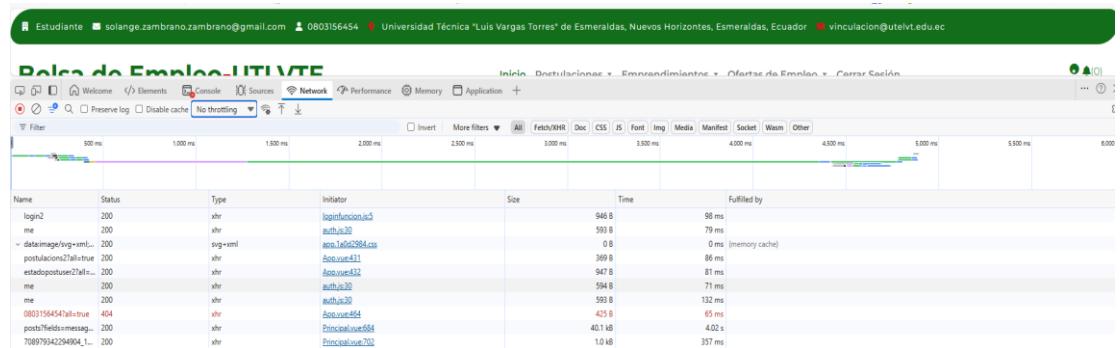
**Figura 17. Inicio de sesión exitoso para el usuario Estudiante y Administrador-CVN**

**Nota:** Al iniciar sesión un usuario estudiante, tendrá acceso a módulos de registro y actualización del curriculum

ID	Nombre	Apellido Paterno	Apellido Materno	Estado del CVN
0804040321	ESTUPIÑAN	ESTUPIÑAN	GABRIEL OMAR	Completo
0804040328	CHI SPID	CHI SPID	KARINA AMBERINA	Completo
0802022239	CORTÉZ	GODDÍY	KARLA MIREYEDES	Completo
0804060710	RAMÍREZA	SAWAPIRIA	JEFTH AL-HRITI	Completo
0803066514	ZAMBRANO	ZAMBRANO	SOLANGE LILITH	Completo
7200329742	ALVAREZ	YUMBO	STEVEN MARCIO	Completo

**Figura 18. Obtención de datos**

**Nota:** Se obtienen datos mediante la solicitud GET enviadas al backend



**Figura 19. Pruebas peticiones HTTP**

## 2.2. Técnicas de recolección de datos

Para la fase inicial de exploración , orientada a la identificación de los requerimientos funcionales, no funcionales y expectativas de los usuarios, se optó por realizar entrevistas semiestructuradas y observaciones directas.

Al aplicar las entrevistas semiestructuradas se pudo conocer las percepciones sobre las dificultades existentes en la gestión de sus curriculum, su proceso de búsqueda de empleo y expectativas frente a una plataforma institucional. Entre los temas abordados se incluyeron: la utilidad de un curriculum normalizado, la importancia de tener un curriculum bien estructurado, los problemas detectados en plataformas externas de empleo

y las funciones que consideraban más relevantes para una bolsa de empleo universitaria.

Durante las pruebas piloto de ambas plataformas, donde los usuarios interactuaron con los módulos desarrollados. Este procedimiento permitió identificar de manera empírica las fortalezas y debilidad del sistema, lo que incluye: patrones de uso, errores recurrentes, tiempos de respuesta y nivel de satisfacción en la navegación.

La retroalimentación informal obtenida durante los talleres de demostración, donde los usuarios compartieron sugerencias sobre la disposición de los menús, validación de formularios y claridad de las instrucciones.

La combinación de estos instrumentos para el enfoque cualitativo permitió contrastar la información, aportando mayor validez al proceso de desarrollo y ajuste de las plataformas.

### **CAPÍTULO III. RESULTADOS Y DISCUSIÓN**

Para este capítulo se presentan: los resultados del desarrollo de las plataformas, resultados funcionales e integración entre plataformas y las discusiones de estos resultados.

#### **Resultados del desarrollo de ambas plataformas.**

El desarrollo de las plataformas propuestas CVN y Bolsa de empleo, permitió obtener sitios web que responden a los requerimientos técnicos y funcionales establecidos en el segundo capítulo de este documento. Ambas plataformas se construyeron utilizando tecnologías modernas y de código abierto, como lo es: Laravel 10 para el backend, Vue.js 3 para el frontend y MySQL8 como gestor de base de datos.

La metodología ágil XP facilitó el desarrollo incremental con avances de versiones funcionales y la incorporación de mejoras derivadas de la retroalimentación obtenida durante el proceso de prueba, esto garantizó un software de buena calidad, modular, escalable y de fácil mantenimiento. La siguiente tabla muestra los enlaces de los repositorios de las plataformas:

**Tabla 9. Repositorio GitHub de las plataformas**

Plataformas	Enlaces
CVN	Backend: <a href="https://github.com/luisolis20/CvnBAckend">https://github.com/luisolis20/CvnBAckend</a> Frontend: <a href="https://github.com/luisolis20/CVNFrontend">https://github.com/luisolis20/CVNFrontend</a>
Bolsa de Empleo	Backend: <a href="https://github.com/luisolis20/B_E-backkend">https://github.com/luisolis20/B_E-backkend</a> Frontend: <a href="https://github.com/luisolis20/B_E-frontend2">https://github.com/luisolis20/B_E-frontend2</a>

En la fase de codificación, se implementaron los módulos de autenticación, gestión de usuarios generación de currículos, registro de empresas y/o emprendimientos, publicación de ofertas y postulación, siguiendo la arquitectura MVC proporcionada por Laravel y la arquitectura MVVM de Vue. Para tener un control de versiones, garantizar la transparencia, la trazabilidad y la posibilidad de revisión del trabajo desarrollado, los repositorios completos fueron alojados en GitHub, donde se encuentra el código fuente del backend y frontend de ambas plataformas.

En cuanto a la plataforma CVN, lo resultados obtenidos se centraron en la correcta gestión, almacenamiento y generación de la información profesional y académica de los usuarios. La plataforma permite que todos los estudiantes ya sean egresado o graduados obtener sus datos personales, registrar y/o actualizar su: formación académica, experiencia profesional, publicaciones, habilidades y otros apartados de su perfil profesional, generando automáticamente un documento en formato PDF y estandarizado. Si bien, estos resultados representan un avance significativo en la normalización del curriculum institucional, contribuyendo a la homogeneización de los perfiles profesionales de los estudiantes de la UTLVTE.

Asimismo, la plataforma Bolsa de empleo obtuvo resultados satisfactorios en la gestión de ofertas laborales y de emprendimiento, publicación de vacantes, registro de

empresas, postulaciones y validación de usuarios. Las pruebas funcionales demostraron una correcta interoperabilidad entre ambos sistemas permitiendo que los datos de los CVN de los usuarios estudiantes se integren de manera automática en las postulaciones. Así los usuarios estudiantes pueden aplicar a ofertas laborales y/o emprendimientos sin necesidad de volver a ingresar datos de su currículum dentro de esta misma plataforma, con eso se optimiza la experiencia de uso y la eficiencia del sistema.

Durante el proceso de pruebas de rendimiento y usabilidad, se validó la seguridad de los datos mediante el uso de tokens JWT para la autenticación y la encriptación de contraseñas mediante algoritmos hash, asegurando la protección de la información sensible. Las entrevistas semiestructuradas y observaciones directas aplicadas a los estudiantes, egresados y graduados proporcionaron información valiosa sobre las principales necesidades de usabilidad, organización de datos y funcionalidad esperada. Gracias a estos insumos, se priorizó el diseño de una interfaz intuitiva y fácil de usar, se pudo identificar las dificultades iniciales en la navegación y ajustar la disposición de los menús, mejorando significativamente la experiencia del usuario final.

Otro resultado que destaca es la correcta implementación de roles y permiso dentro de las plataformas, garantizando que cada tipo de usuario ya sea: estudiante, empresa, administrador tenga acceso únicamente a las funcionalidades que le corresponde. Al realizar esta segmentación por roles refuerza la integridad del sistema y la trazabilidad de cada una de las acciones ejecutadas por el usuario.

Para cerrar estos resultados, el proceso de despliegue a servidores locales y su posterior migración a un entorno AlmaLinux con Apache confirmó la portabilidad y disponibilidad de ambas plataformas. Las plataformas demostraron estabilidad en entornos de producción reales, cumpliendo con los requisitos no funcionales de compatibilidad,

disponibilidad e interoperabilidad.

### **Resultados funcionales e integración entre plataformas.**

La integración entre ambas plataformas se logró mediante el uso de API RESTful, lo cual permitió el intercambio de información en formato JSON entre el backend (Laravel) y el frontend (Vue).

Desde el punto de vista funcional de las plataformas, se comprobó el correcto flujo de información. Los estudiantes, egresados y graduados pueden llenar los datos necesarios para su CVN y utilizarlo automáticamente para postularse a ofertas laborales o de emprendimientos, mientras que las empresas y usuarios estudiantes con emprendimiento pueden gestionar sus ofertas, visualizar CVN de postulantes y descargarlos.

De igual manera, el sistema de notificaciones internas y por correo electrónico fue validado, confirmando su correcto funcionamiento para informar a los usuarios sobre eventos relevantes, como nuevos emprendimientos y/u ofertas registradas, postulaciones o aprobaciones de emprendimiento y/u ofertas. Este punto incrementa la interacción dentro de la plataforma y mantiene informada a la comunidad universitaria en tiempo real.

En cuanto a la usabilidad de las plataformas, los resultados evidencian que las interfaces gráficas, diseñadas con Bootstrap, proporcionan una experiencia intuitiva y adaptada a diferentes dispositivos (responsive) y accesibilidad. La dinamicidad y validación de formularios en tiempo real optimizan la captura de información y reducen errores de ingreso de datos. En la siguiente imagen se presenta de manera grafica la usabilidad y rendimiento de ambas plataformas:



Figura 20. Gráfica de Funcionalidad y Rendimiento de la Plataforma CVN



Figura 21. Gráfica de Funcionalidad y Rendimiento de la Plataforma Bolsa de Empleo

## **Discusión de resultados**

Los resultados obtenidos en el presente trabajo concuerdan con los hallazgos previos sobre el impacto de las plataformas digitales en la empleabilidad universitaria. Por ejemplo, [5] demostró que los sistemas web basados en Laravel y metodologías ágiles contribuyen a optimizar el seguimiento a graduados y recolección de información laboral, también fortalece la vinculación con el entorno productivo. En este mismo ámbito, los resultados de esta investigación evidencian que la implementación de una bolsa de empleo universitaria propia promueve la autonomía tecnológica de la institución.

El mismo autor detalla que, la falta de estandarización en los currículum académicos dificulta la evaluación equitativa de los perfiles profesionales. Es así como, la implementación de la plataforma CVN constituye un aporte innovador, al permitir que todos los estudiantes, egresados y graduados de la UTLVTE presenten su información bajo un formato uniforme, lo cual garantiza transparencia y coherencia institucional.

La correcta interoperabilidad entre las plataformas desarrolladas demuestra la viabilidad de integrar soluciones web basadas en PI RESTful, una práctica recomendada en la literatura reciente sobre arquitectura de software distribuido. Dicha implementación refuerza la idea de que el uso de frameworks modernos, como Vue.js (frontend) y Laravel (backend), facilita la comunicación entre componentes heterogéneos, asegurando un desempeño y experiencia fluida para el usuario final.

Otro aspecto relevante se relaciona con la seguridad y protección de datos personales, el uso de tokens JWT, cifrado de contraseñas y autenticación mediante middleware en Laravel permitió cumplir con las normativas de seguridad informática. Esto marca una diferencia sustancial respecto a sistemas tradicionales que no contemplan mecanismos robustos de acceso ni auditoria de actividades.

## **CONCLUSIONES**

El desarrollo e implementación de las plataformas CVN y Bolsa de Empleo para la UTLVTE cumplió con los objetivos de la investigación, estableciendo un sistema integral para la gestión de información profesional y la vinculación efectiva entre la comunidad académica y el sector productivo.

La plataforma CVN estandarizó la presentación de currículums bajo un formato institucional, homogeneizando los perfiles y fortaleciendo los procesos de acreditación y seguimiento de estudiantes y egresados. Por su parte, la Bolsa de Empleo demostró su efectividad al conectar a los usuarios con oportunidades laborales y de emprendimiento, optimizando los procesos de postulación.

Desde la perspectiva técnica, la metodología ágil XP permitió iteraciones rápidas y una mejora continua, mientras que la arquitectura con Laravel y Vue.js demostró ser robusta, segura y de alto rendimiento. La integración mediante API RESTful aseguró una interoperabilidad fluida y una experiencia de usuario unificada.

Este proyecto representa un avance significativo en la transformación digital de la UTLVTE, promoviendo la vinculación con el sector empresarial, el seguimiento de graduados y el fortalecimiento de la empleabilidad.

## **RECOMENDACIONES**

Fortalecer la Gobernanza y Mantenimiento del Sistema. Establecer un proceso institucional formal y periódico para la actualización de los CVN, asignando responsabilidades a docentes, coordinadores y decanos. Asegurar el mantenimiento continuo y la capacitación permanente del personal técnico y administrativo para garantizar la operatividad y actualización de las plataformas.

Expandir la Cobertura y Accesibilidad de las Plataformas. Ampliar la funcionalidad del CVN para incluir a docentes y personal administrativo, y desarrollar una versión móvil de ambas plataformas (CVN y Bolsa de Empleo) para incrementar la accesibilidad y facilitar su uso desde cualquier dispositivo.

Fomentar la Vinculación con el Sector Productivo y Egresados. Promover activamente la participación de empresas, instituciones públicas y egresados con emprendimientos para ampliar la oferta laboral en la Bolsa de Empleo, fortaleciendo así la relación entre la universidad y su entorno económico.

## **REFERENCIAS BIBLIOGRÁFICAS**

- [1] UNESCO, “Promover la empleabilidad de los estudiantes a través de la educación superior.” [Online]. Available: <https://www.iesalc.unesco.org/es/articles/promover-la-empleabilidad-de-los-estudiantes-traves-de-la-educacion-superior>
- [2] N. Llevat, “¿Qué es el Currículum Europeo o Europass y para qué sirve?” [Online]. Available: <https://www.ifp.es/blog/que-es-el-curriculum-europeo-o-europass-y-para-que-sirve#:~:text=Acceso%20a%20empleos%2C%20becas%20y,m%C3%A1s%20accesibles%20las%20oportunidades%20internacionales.>
- [3] Universidad Politécnica Salesiana, “Bolsa de Trabajo - Universidad Politécnica Salesiana.” [Online]. Available: [https://www.ups.edu.ec/en/bolsa\\_de\\_trabajo?utm\\_source=chatgpt.com](https://www.ups.edu.ec/en/bolsa_de_trabajo?utm_source=chatgpt.com)
- [4] CACES, “Modelo de evaluación caces 2023,” vol. d, pp. 1–23, 2023, [Online]. Available: <https://www.caces.gob.ec/wp-content/uploads/2023/12/Modelo-de-Evaluaci%C3%B3n-Externa-UEP-2023-1.pdf>
- [5] T. Tocagón and M. Benjamín, “Desarrollo de un sistema web de bolsa de empleos y seguimiento de egresados y graduados del instituto superior tecnológico José Chiriboga Grijalva (ITCA) mediante la implementación de una arquitectura de microservicios con laravel framework,” 2021, [Online]. Available: <https://repositorio.utn.edu.ec/handle/123456789/10871>
- [6] I. Gutiérrez, K. Lara, and A. Valencia, “Impacto en la inserción laboral de las plataformas de empleo para estudiantes universitarios,” Polo del Conoc., vol. 9, no. 11, pp. 1213–1235, 2024, [Online]. Available: <https://polodelconocimiento.com/ojs/index.php/es/article/view/8372/pdf>

[7] A. M. José Gilberto, E. R. Zeus Honorio, O. V. Maylee Lisbeth, and G.-W. Mariuxi Elizabeth, “Creación de una página web y bolsa de empleo para los Institutos Superiores Tecnológicos de Esmeraldas,” vol. 5, no. 10, pp. 868–902, 2020, doi: 10.23857/pc.v5i10.1857.

[8] ULPGC, “Currículum Vítæ Normalizado (CVN),” 2023, [Online]. Available: <https://biblioteca.ulpgc.es/cvn>

[9] G. Toala, “Seguimiento a graduados y egresados de la Escuela de contabilidad y auditoría de la PUCESE,” 2020, [Online]. Available: [https://pedagogiaignaciana.com/biblioteca-digital/biblioteca-general?view=file&id=2755:seguimiento-a-graduados-y-egresados-de-la-escuela-de-contabilidad-y-auditoria-de-la-puce-esmeraldas&catid=8&filename=Toala G. 2020 Seguimiento graduadoContabilidad y Au](https://pedagogiaignaciana.com/biblioteca-digital/biblioteca-general?view=file&id=2755:seguimiento-a-graduados-y-egresados-de-la-escuela-de-contabilidad-y-auditoria-de-la-puce-esmeraldas&catid=8&filename=Toala%202020%20Seguimiento%20graduadoContabilidad%20y%20Au)

[10] J. Eduardo, M. Gavilanes, A. Mercedes, and C. Vargas, “APLICACIÓN WEB PARA EL SEGUIMIENTO INSTITUTO SUPERIOR TECNOLÓGICO WEB APPLICATION FOR GRADUATE TRACKING AND JOB BOARD MANAGEMENT AT THE INSTITUTO SUPERIOR TECNOLÓGICO,” 2025, [Online]. Available: <https://ciencialatina.org/index.php/cienciala/article/view/19098/27283>

[11] J. G. Sacristán, “¿Qué significa el currículum?,” pp. 18–43, 2010, [Online]. Available:

[https://www.scielo.org.mx/scielo.php?script=sci\\_arttext&pid=S1665-109X2010000100009](https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1665-109X2010000100009)

[12] N. Didier, “Capital Humano Nominal, Empleabilidad y Credencialismo,” Rev. Iberoam. Psicol. Cienc. y Tecnol., vol. 7, no. 2, pp. 19–27, 2014, doi: 10.33881/2011-1786.rip.%x.

- [13] M. C. Lopez, “Arquitectura Clientes Servidor - Sistema de información distribuido,” 2020, [Online]. Available: <https://es.scribd.com/document/483177824/Arquitectura-Clientes-Servidor>
- [14] A. R. Garcia, S. B. Filipe, C. Fernandes, C. Estevão, and G. Ramos, “ARQUITECTURA CLIENTE – SERVIDOR,” 2012, [Online]. Available: <https://iscteposcolula.wordpress.com/wp-content/uploads/2012/02/cte-servidor.pdf>
- [15] O. Blancarte, “Arquitectura Cliente-Servidor.” [Online]. Available: <https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/cliente-servidor>
- [16] J. C. Villanueva Guzmán, E. Gómez Domínguez, P. Zamora Reséndiz, A. Priego Clemente, and E. Landero García, “APIs RESTful para interoperabilidad entre aplicación Móvil y aplicación Web,” Cienc. Lat. Rev. Científica Multidiscip., vol. 9, no. 3, pp. 6795–6821, 2025, doi: 10.37811/cl\_rcm.v9i3.18325.
- [17] L. Z. Wellington Paul, M. B. Liceth Monserrate, P. Z. William Israel, and F. S. Edwin Gustavo, “Diseño Estratégico de APIs Escalables y Seguras para la Integración de Sistemas y Aplicaciones”.
- [18] A. Marketing, “Definición de API REST: ¿Qué son las API REST (API RESTful)?” [Online]. Available: <https://www.astera.com/es/type/blog/rest-api-definition/>
- [19] C. P. Ricardo Javier, B. A. Miguel Fabricio, and M. O. Aldo Patricio, Programación Web del Frontend al Backend. 2023. [Online]. Available: <https://dialnet.unirioja.es/servlet/libro?codigo=933116>
- [20] N. M. Egea, “Frameworks de desarrollo web back-end,” MoleQla Rev. Ciencias la Univ. Pablo Olavide, ISSN-e 2173-0903, N°. 36, 2020, no. 36, p. 13, 2020, [Online]. Available:

<https://dialnet.unirioja.es/servlet/articulo?codigo=7220429&info=resumen&idioma=SPA>

[21] colaboradores de Wikipedia, “Vue.js.” [Online]. Available:

<https://es.wikipedia.org/wiki/Vue.js>

[22] P. R. J. Celi, M. O. Aldo Patricio, M. F. B. Andrade, and S. X. E. Chancay, “Desarrollo de Aplicaciones Web Utilizando Vue.Js, Axios y PHPWeb Application Development Using Vue.Js, Axios and PHP,” vol. 4, 2023, [Online].

Available: <https://revista.gnerando.org/revista/index.php/RCMG/article/view/94/86>

[23] Vue.js, “Componentes básicos.” [Online]. Available:

<https://vuejs.org/guide/essentials/component-basics>

[24] S. Burbano, “Reactividad y programación en vue: efectos y ciclos de vida.” [Online]. Available: <https://platzi.com:8443/cursos/reactividad-vuejs/siguientes-pasos-con-vuejs/>

[25] Vue, “Ganchos de ciclo de vida.” [Online]. Available:

<https://vuejs.org/guide/essentials/lifecycle>

[26] N. L. Ayala Frasnelli and A. D. Ruiz Diaz Medina, “Un acercamiento teórico y práctico a la implementación de Flux,” Impacto. Rev. Cienc. y Tecnol., vol. 1, no. 1, pp. 47–56, 2021, [Online]. Available:

<http://revista.cyt.uni.edu.py/index.php/revista/article/view/19>

[27] M. Iriarte, “Estado Global con Vuex en Vue.js.” [Online]. Available:

<https://somospnt.com/blog/302-estado-global-con-vuex-en-vue-js>

[28] J. GUTIÉRREZ, “Validación de formularios con Vue.” [Online].

Available: <https://javiergutierrez.trade/validacion-de-formularios-con-vue/>

[29] A. Zia, “Formularios y validación de Vue.js.” [Online]. Available:

<https://matifzia.medium.com/vue-js-forms-and-validation-a3f0e3ed8b35>

- [30] Vue.js, “Validación de Formularios.” [Online]. Available: <https://es.vuejs.org/v2/cookbook/form-validation>
- [31] A. Ledo, “Guía completa de Eloquent ORM en Laravel,” 2024, [Online]. Available: <https://es.scribd.com/document/546072214/La-historia-de-Laravel>
- [32] E. P. Gladys Geoconda, Q. C. Luis René, C. G. Abraham Paul, and L. A. Santiago Javier, “Frameworks del lado del Servidor: Caso de Estudio Node JS, Django y Laravel,” 2024, [Online]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=9966614>
- [33] Á. R. España León, M. I. González Valero, J. T. Mejía Viteri, J. A. Campi Mayorga, and I. I. Campi Mayorga, “Patrón MVC, un componente para la implementación de una Estrategia Informática para mejorar gestión de datos en el área de estadística: Caso de Estudio Hospital Maternidad Babahoyo.,” Episteme Rev. Digit. Ciencia, Tecnol. e Innovación, vol. 3, no. 4, dic, p. 14, 2016, [Online]. Available: <http://186.46.158.26/ojs/index.php/EPISTEME/article/view/280>
- [34] C. Sagredo, J. Gamaliel, T. Espinosa, M. Reyes, L. García, and M. De Lourdes, “Automatización de la codificación del patrón modelo vista controlador ( mvc ) en proyectos orientados a la Web,” vol. 1, no. Mvc, pp. 239–250, 2012.
- [35] V. A. Flores, “Controladores en Laravel ¿Qué son y para que sirven?” [Online]. Available: <https://codersfree.com/courses-status/aprende-laravel-desde-cero/controladores-en-laravel-que-son-y-para-que-sirven>
- [36] V. A. Flores, “Rutas en Laravel ¿Qué son y para que sirven?” [Online]. Available: <https://codersfree.com/courses-status/aprende-laravel-desde-cero/rutas-en-laravel-que-son-y-para-que-sirven>
- [37] Documentación Laravel, “Middleware.” [Online]. Available:

<https://documentacionlaravel.com/docs/11.x/middleware>

[38] R. Siddique, “Laravel Validation: A Complete Guide for Your Data Handling.” [Online]. Available: <https://eluminoustechologies.com/blog/laravel-validation/>

[39] Jeremy Holcombe, “Validación de Datos en Laravel : Conveniente y Potente,” no. October 2025, 2025, [Online]. Available: <https://kinsta.com/es/blog/laravel-validacion/>

[40] J. Ramos, “Cómo usar Form Requests y Policies en Laravel.” [Online]. Available: <https://programacionymas.com/blog/form-requests-y-policies#:~:text=Primero%20que%20todo%2C%20qu%C3%A9%20es,%20el%20registro%20de%20nuevos%20datos.>

[41] S. Raji, “¿Qué es una solicitud de Laravel?” [Online]. Available: <https://medium.com/@ormrepo/what-is-a-laravel-request-aa6d072e276a>

[42] P. Prajapati, “¿Qué es la clase de respuesta de Laravel?” [Online]. Available: <https://www.elightwalk.com/blog/laravel-response-class>

[43] Documentación Laravel, “Respuestas HTTP,” 2023, [Online]. Available: <https://laravel-docs.com/es/docs/10.x/responses>

[44] Documentación Laravel, “Autenticación.” [Online]. Available: <https://laravel-docs.com/es/docs/10.x/authentication>

[45] Documentación Laravel, “Protección CSRF.” [Online]. Available: <https://laravel.com/docs/12.x/csrf>

[46] F. Tobiasz, “Uso del Vue: ventajas e inconvenientes.” [Online]. Available: <https://thecodest.co/es/blog/pros-y-contras-de-vue/>

[47] Admin\_solbyte, “¿Qué es React JS? Ventajas e Inconvenientes.” [Online]. Available: [https://www.solbyte.com/blog/react-js-ventajas-e-inconvenientes/#Ventajas\\_de\\_usar\\_React\\_JS](https://www.solbyte.com/blog/react-js-ventajas-e-inconvenientes/#Ventajas_de_usar_React_JS)

- [48] K. Shields, “Angular Development Pros and Cons: Is It Right for Your Web Application.” [Online]. Available: [https://designlio.translate.google/blog/angular-development-web-application?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=es&\\_x\\_tr\\_hl=es&\\_x\\_tr\\_pto=tc](https://designlio.translate.google/blog/angular-development-web-application?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)
- [49] M. Davey, “The 4 Best Local WordPress Development Environments in 2020: XAMPP vs MAMP vs Local vs DesktopServer.” [Online]. Available: [https://deliciousbrains.com.translate.google/xampp-mamp-local-dev/?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=es&\\_x\\_tr\\_hl=es&\\_x\\_tr\\_pto=tc](https://deliciousbrains.com.translate.google/xampp-mamp-local-dev/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)
- [50] B. Jackson, “¿Qué Es MySQL? Una Explicación para Principiantes.” [Online]. Available: <https://kinsta.com/es/blog/que-es-mysql/>
- [51] Arsys, “¿Qué es MySQL? Explicación y características.” [Online]. Available: <https://www.arsys.es/blog/mysql#tree-1>
- [52] M. Görög, “Modelo entidad-relación: explicado de forma sencilla e implementado de forma práctica.” [Online]. Available: <https://www.collaboard.app/en/blog/entity-relationship-model/>
- [53] Documentación Laravel, “Base de datos: Migraciones,” 2025, [Online]. Available: <https://laravel.com/docs/12.x/migrations>
- [54] N. Pariente, “Git y GitHub: qué son y por qué todo desarrollador web debería usarlos.” [Online]. Available: <https://codespaceacademy.com/git-y-github-que-son-por-que-usarlos/#contenido>
- [55] M. Flinders, “¿Qué es un servidor Linux?” [Online]. Available: <https://www.ibm.com/mx-es/think/topics/linux-servers>
- [56] J.Pomeyrol, “AlmaLinux 9 amplía su oferta para el escritorio.” [Online]. Available: <https://www.muylinux.com/2022/05/30/almalinux-9/>
- [57] Equipo editorial de IONOS, “¿Qué es SSH (Secure Shell)?”

[Online]. Available: [https://www.ionos.com/es-](https://www.ionos.com/es-us/digitalguide/servidores/herramientas/protocolo-ssh/)

us/digitalguide/servidores/herramientas/protocolo-ssh/

[58] J. Márquez Díaz, L. Sampedro, and F. Vargas, “Instalación y configuración de Apache, un servidor Web gratis,” un Serv. Web gratis Ing. y Desarro., no. 12, pp. 10–23, 2002, [Online]. Available:

<https://www.redalyc.org/pdf/852/85201202.pdf>

[59] L. Serra, “Los 4 puntos para implementar el plan estratégico de una empresa.” [Online]. Available: <https://empresas.blogthinkbig.com/como-implementar-plan-estrategico-empresarial/>

## ANEXOS

### Anexo 1: Capacitaciones realizadas a estudiantes y graduados sobre la plataforma CVN y Bolsa de Empleo, 20 y 21 de agosto del 2025

