

13

SISTEMA WEB

**PARA LA GESTIÓN DE BOLSA DE TRABAJO EN LA
UNIVERSIDAD IBEROAMERICANA DEL ECUADOR**



SISTEMA WEB

PARA LA GESTIÓN DE BOLSA DE TRABAJO EN LA UNIVERSIDAD IBEROAMERICANA DEL ECUADOR

WEB SYSTEM FOR JOB PLACEMENT MANAGEMENT AT THE IBERO-AMERICAN UNIVERSITY OF ECUADOR

Tonysé de la Rosa-Martín¹

E-mail: severus.trm@gmail.com

ORCID: <https://orcid.org/0000-0002-0881-6034>

Jair Castillo¹

E-mail: jaiirsk7@gmail.com

ORCID: <https://orcid.org/0009-0001-9209-7852>

¹ Universidad Iberoamericana del Ecuador. Ecuador.

Cita sugerida (APA, séptima edición)

Rodríguez Muñoz, R., & Socorro Castro, A. R. (2021). Reflexiones respecto a modelos y procesos de gestión de Ciencia, Tecnología e Innovación. *Revista Metropolitana de Ciencias Aplicadas*, 9(3), 114-126.

Fecha de presentación: 19/02/2026

Fecha de aceptación: 21/03/2026

Fecha de publicación: 01/05/2026

RESUMEN

El acceso oportuno y confiable a oportunidades laborales es crucial para la inserción profesional de estudiantes y egresados, por lo que las instituciones de educación superior deben articular la formación académica con las dinámicas del mercado laboral para fortalecer la empleabilidad y el desarrollo profesional, donde las tecnologías de la información y la comunicación asumen un rol estratégico al optimizar la gestión, organización y difusión de ofertas mediante plataformas centralizadas y accesibles. La Universidad Iberoamericana del Ecuador enfrenta la ausencia de un sistema automatizado para su bolsa de trabajo, lo cual provoca dispersión de la información y acceso limitado a ofertas actualizadas, afectando los procesos de inserción laboral; ante esto, la investigación propone desarrollar un sistema web orientado a facilitar el acceso a oportunidades organizadas por carrera, garantizar datos actualizados y reducir la carga operativa. Para lograrlo se plantea automatizar la recopilación de ofertas mediante técnicas de web scraping, y diseñar un prototipo con frontend en React, backend en Node.js y servicios en la nube para autenticación y almacenamiento, además de ejecutar pruebas funcionales que verifiquen el cumplimiento de los requerimientos y aseguren la calidad del sistema. Así entonces, la solución propuesta constituye una alternativa viable y escalable que fortalece la vinculación entre formación y mercado laboral, mejora la accesibilidad a oportunidades y genera beneficios tanto para los estudiantes como para la comunidad universitaria.

Palabras clave:

Web Scraping, automatización, gestión, bolsa de empleo.

ABSTRACT

Timely and reliable access to job opportunities is crucial for the professional integration of students and graduates. Therefore, higher education institutions must align academic training with labor market dynamics to strengthen employability and professional development. Information and communication technologies (ICTs) play a strategic role in this process by optimizing the management, organization, and dissemination of job postings through centralized and accessible platforms. The Ibero-American University of Ecuador faces the absence of an automated job board, leading to information dispersion and limited access to updated job postings, thus impacting the job placement process. In response, this research proposes developing a web-based system to facilitate access to opportunities organized by degree program, ensure up-to-date data, and reduce the operational burden. To achieve this, the project proposes automating the collection of job postings using web scraping techniques and designing a prototype with a React frontend, a Node.js backend, and cloud services for authentication and storage. Functional tests will also be conducted to verify compliance with requirements and ensure system quality. Thus, the proposed solution constitutes a viable and scalable alternative that strengthens the link between education and the labor market, improves access to opportunities, and generates benefits for both students and the university community.

Keywords:

Web scraping, automation, management, job board.

INTRODUCCIÓN

Uno de los principales desafíos en la dimensión profesional de los recién egresados radica en la transición del entorno académico al ámbito laboral, especialmente en un contexto caracterizado por una alta competitividad y el acelerado avance de las tecnologías digitales. Este escenario intensifica la dinámica de articulación entre la oferta formativa de las universidades y las demandas del mercado laboral, evidenciando un proceso aún en consolidación, en el cual la inserción profesional se convierte en un mecanismo fundamental para la transferencia efectiva del conocimiento hacia la sociedad. En este sentido, incluso la divulgación académica adquiere mayor relevancia cuando se vincula con dinámicas laborales que fortalecen la visibilidad, la pertinencia y el prestigio de las instituciones de educación superior, lo que pone en evidencia la necesidad de contar con herramientas que faciliten la conexión entre los estudiantes, los graduados y el sector productivo.

En este contexto, la Universidad Iberoamericana del Ecuador (UNIB.E) enfrenta una limitación significativa al no disponer de un sistema tecnológico integral que permita agrupar, gestionar y difundir de manera estructurada las oportunidades laborales dirigidas a sus graduados. Esta carencia puede convertirse en una barrera para la eficacia de los procesos de intermediación laboral, ya que debilita el rol de la institución como primer eslabón en la inserción profesional de sus egresados. A su vez, esta situación dificulta el seguimiento sistemático del impacto que los programas académicos tienen en el desempeño profesional de sus titulados, limitando la posibilidad de generar procesos de mejora continua basados en evidencia.

Desde esta perspectiva, el presente proyecto se fundamenta en su contribución directa al fortalecimiento de la empleabilidad de los graduados y a la optimización de los procesos internos de análisis institucional. La implementación del sistema propuesto representa una oportunidad estratégica para mejorar la administración de la bolsa de trabajo universitaria, consolidar vínculos más sólidos con el sector empresarial, reafirmar el compromiso social de la academia en la generación y transferencia de conocimiento, y promover la actualización constante de los programas educativos en función de las demandas del entorno profesional.

Hoy en día, las universidades enfrentan el desafío de integrar la tecnología como un componente esencial de su propia estructura institucional, especialmente en procesos que han evolucionado hacia la optimización de tiempos y la medición objetiva de resultados e impactos en sus distintas áreas de vinculación con la sociedad. En este contexto, las instituciones de educación superior desempeñan un papel relevante y decisivo en el desarrollo social, ya que, si bien tradicionalmente su función principal ha sido la formación y generación de conocimiento,

actualmente también adquieren una dimensión estratégica en la proyección profesional del estudiante. En este sentido, elementos como la visibilidad institucional, la credibilidad académica y la formación en valores contribuyen a la inserción de los nuevos profesionales en la sociedad, siendo el mercado laboral el principal escenario donde dicha proyección se materializa.

Por otro lado, el estudiante graduado se enfrenta a un entorno laboral caracterizado por una creciente competitividad y por constantes transformaciones derivadas de factores sociales y económicos que condicionan su inserción profesional. La ausencia de canales formales de intermediación laboral genera una desconexión entre la academia y el sector productivo, lo que limita el impacto social de la universidad. En este sentido, la inexistencia de una plataforma centralizada dificulta el acceso equitativo a oportunidades laborales, generando desigualdades en la inserción profesional y afectando tanto el desarrollo individual de los egresados como el fortalecimiento del desarrollo económico local y nacional. Esta situación también aleja a las instituciones del cumplimiento del Objetivo de Desarrollo Sostenible N.º 8, el cual promueve el trabajo decente y el crecimiento económico inclusivo de manera medible y sostenible (Organización de las Naciones Unidas, 2024).

Asimismo, esta ineficiencia operativa limita la capacidad de las universidades, y en particular de sus estudiantes, para liderar procesos de gestión tecnológica educativa, manteniendo ecosistemas digitales desactualizados frente a las exigencias actuales de escalabilidad, seguridad y mantenimiento propias del desarrollo de software moderno. En consecuencia, surge la necesidad de implementar un sistema web para la administración de la bolsa de trabajo en instituciones como la Universidad Iberoamericana del Ecuador (UNIB.E), el cual integre módulos de registro, búsqueda, publicación de ofertas, seguimiento de postulaciones y generación de informes, constituyéndose en un elemento mínimo indispensable para fortalecer su rol dentro de la sociedad contemporánea.

En este artículo se describe la estructura de un sistema web orientado a mejorar la gestión de la bolsa de trabajo en la Universidad Iberoamericana del Ecuador.

METODOLOGÍA

El presente proyecto se desarrolló bajo un enfoque de ingeniería de software aplicada, orientado a la construcción de una solución digital para la gestión de la bolsa de trabajo de la Universidad Iberoamericana del Ecuador. Este enfoque permitió abordar el problema desde una perspectiva sistemática, estructurada y orientada a la resolución de necesidades reales, integrando principios de la informática moderna, la arquitectura de sistemas web y la gestión eficiente de datos.

En este sentido, la investigación se fundamenta en un proceso de desarrollo basado en metodologías ágiles, específicamente el marco de trabajo Scrum, el cual permite gestionar la complejidad del sistema de forma iterativa e incremental. Este enfoque facilita la adaptación continua a los requerimientos funcionales, asegurando la entrega progresiva de valor y la validación constante con los usuarios finales. Tal como señalan Schwaber & Sutherland (2024); y Sommerville (2020) los procesos ágiles permiten responder de manera flexible a los cambios del entorno mediante ciclos cortos de desarrollo, retroalimentación y mejora continua.

El proceso metodológico se estructuró en cuatro fases principales: levantamiento de requerimientos, diseño del sistema, implementación y validación funcional. En la primera fase se identificaron las necesidades del sistema mediante el análisis del contexto institucional y la definición de historias de usuario, permitiendo establecer las funcionalidades esenciales como autenticación, publicación de ofertas, búsqueda y gestión de vacantes laborales.

En la fase de diseño se definió la arquitectura del sistema bajo un modelo cliente-servidor, integrando tecnologías modernas como React para el frontend, Node.js para el backend y Firebase como plataforma de servicios en la nube. Esta arquitectura permitió establecer una separación clara de responsabilidades, favoreciendo la escalabilidad, mantenibilidad y seguridad del sistema.

La fase de implementación se llevó a cabo mediante el desarrollo modular del sistema, incorporando componentes reutilizables y aplicando el patrón de diseño Modelo Vista Controlador (MVC) para organizar la lógica de negocio, la interfaz de usuario y el manejo de datos. Asimismo, se integraron servicios de Firebase para autenticación, almacenamiento y gestión en tiempo real de la información, garantizando una experiencia de usuario fluida y segura.

Para la recolección automática de información laboral se implementó un módulo de web scraping, apoyado en herramientas como Playwright, lo que permitió la extracción dinámica de datos desde fuentes externas. Este proceso aseguró la actualización constante de las ofertas laborales dentro del sistema, optimizando la gestión de la información disponible para los usuarios.

En cuanto a la gestión de datos, se utilizó una base de datos NoSQL proporcionada por Firebase (Cloud Firestore), lo que permitió una estructura flexible, escalable y adecuada para la naturaleza dinámica de los datos del sistema. Complementariamente, se empleó el formato JSON como estándar de intercambio de información entre el frontend y el backend, facilitando la interoperabilidad entre los distintos módulos del sistema.

Finalmente, la fase de validación consistió en la revisión del funcionamiento del sistema a partir de las historias de usuario previamente definidas, evaluando la

correcta ejecución de los procesos de publicación, consulta, autenticación y generación de reportes. Este proceso permitió verificar la coherencia entre los requerimientos funcionales y el producto desarrollado, asegurando el cumplimiento de los objetivos propuestos.

DESARROLLO

La construcción de una solución digital robusta para la gestión de la empleabilidad requiere una comprensión integral de los fundamentos que sustentan la informática moderna. En este sentido, resulta necesario analizar de manera articulada los principios de la ingeniería de software, las arquitecturas de los sistemas web y la eficiencia en la gestión de bases de datos, ya que estos elementos, al integrarse, permiten transformar requerimientos abstractos en soluciones tecnológicas funcionales y operativas. Este proceso de desarrollo no se limita únicamente a la codificación, sino que se enmarca en enfoques metodológicos estructurados que garantizan la calidad, mantenibilidad y escalabilidad del producto final, destacándose en este contexto la adopción de metodologías ágiles y, en particular, del marco de trabajo Scrum, el cual facilita la gestión iterativa de la complejidad técnica.

En relación con esta articulación conceptual, la literatura especializada coincide en que el éxito de los sistemas contemporáneos depende de la adecuada integración entre los enfoques metodológicos y las soluciones tecnológicas. En este sentido, Gaviria & Velásquez (2024) señalan que la ingeniería de software ha evolucionado desde modelos rígidos hacia enfoques ágiles que priorizan la entrega continua de valor, lo cual implica que la selección adecuada de la arquitectura del sistema y de los mecanismos de gestión de datos no solo mejora el rendimiento técnico, sino que también fortalece la capacidad de adaptación frente a los cambios del entorno institucional y las necesidades de los usuarios finales. Bajo esta perspectiva, se presentan a continuación los conceptos fundamentales que sustentan la arquitectura técnica del presente proyecto.

Un sistema web puede entenderse como una arquitectura cliente-servidor diseñada para operar directamente desde un navegador, lo que le otorga una alta disponibilidad y accesibilidad. Su relevancia técnica radica en su naturaleza como sistema distribuido, cuya función principal, según Sommerville (2020), consiste en proporcionar un “acceso seguro y fiable a los datos a través de Internet” (p. 520). Esta característica lo convierte en un componente esencial dentro de las soluciones digitales modernas, especialmente en entornos institucionales donde la disponibilidad de la información es crítica.

Desde una perspectiva aplicada, estas plataformas han evolucionado hasta convertirse en herramientas estratégicas para las organizaciones. En el contexto universitario, IBM (2025) destaca que su implementación resulta clave para “facilitar la automatización de procesos de

vinculación y el seguimiento efectivo de la comunidad estudiantil”. En este marco, el sistema propuesto no se concibe únicamente como un repositorio de información, sino como una solución integral orientada a fortalecer la conectividad institucional y mejorar los procesos de gestión de la empleabilidad bajo un enfoque moderno y eficiente.

Por otra parte, la ingeniería de software constituye una disciplina fundamental en el desarrollo de soluciones digitales de calidad, al integrar principios científicos y tecnológicos orientados a la construcción de sistemas confiables. No se trata únicamente de programar, sino de aplicar un conjunto estructurado de procesos, métodos y herramientas que permiten garantizar la calidad del producto final, tal como señalan Cortes (2019), quienes la describen como un ecosistema integral de “procesos, métodos y herramientas” orientado al desarrollo de software robusto. Este enfoque metodológico permite evitar la improvisación y asegurar un desarrollo sistemático y controlado.

En este contexto, la selección del modelo de desarrollo resulta determinante para la evolución del proyecto. Aunque los enfoques tradicionales ofrecen estructuras secuenciales, las exigencias actuales del entorno tecnológico requieren mayor flexibilidad y capacidad de adaptación. Por ello, los procesos ágiles se presentan como una alternativa adecuada, ya que, según Sommerville (2020), se basan en el “desarrollo iterativo para responder rápidamente a los requerimientos cambiantes” (p. 82). Esta combinación entre rigor técnico y flexibilidad metodológica permite garantizar que el sistema de bolsa de empleo responda de manera efectiva a las necesidades institucionales.

Finalmente, las metodologías ágiles han transformado significativamente el desarrollo de software, al sustituir la planificación rígida por procesos dinámicos centrados en la entrega continua de valor. Bajo este enfoque, el producto se concibe como un sistema evolutivo que se construye en ciclos cortos de desarrollo. En este sentido, Sommerville (2020) señala que estos procesos “se basan en la entrega incremental y el desarrollo iterativo para responder rápidamente a los requerimientos cambiantes” (p. 82); lo que permite una adaptación constante a las necesidades emergentes de los usuarios y de la institución durante el proceso de implementación del sistema.

En este contexto, se ha seleccionado Scrum como marco de trabajo para el desarrollo del sistema, debido a su enfoque basado en la transparencia, la inspección continua y la adaptación progresiva. En este sentido, Schwaber & Sutherland (2024) define Scrum como un “marco de trabajo liviano que ayuda a generar valor a través de soluciones adaptables para problemas complejos” (p. 39); lo cual lo convierte en una metodología idónea para proyectos que requieren flexibilidad y validación constante. Bajo este enfoque, el desarrollo de la bolsa de empleo de la UNIB.E no se concibe como un proceso cerrado o lineal,

sino como una construcción progresiva y colaborativa, en la que se incorporan de manera continua las observaciones de actores clave como administrativos, egresados y empleadores. De esta manera, cada sprint permite entregar incrementos funcionales que responden de forma directa a las necesidades reales del usuario final.

La metodología Scrum se caracteriza por ser un modelo incremental de trabajo ligero, orientado a facilitar la generación de valor en entornos complejos mediante soluciones adaptativas. Su enfoque permite organizar el desarrollo en ciclos iterativos, promoviendo la mejora continua y la retroalimentación constante del producto. Según Schwaber & Sutherland (2024), este marco de trabajo refleja la evolución de una filosofía basada en la experiencia y la observación hacia un sistema que regula las interacciones del equipo, en lugar de imponer instrucciones rígidas y detalladas. Esta característica resulta fundamental en el contexto del presente proyecto, ya que permite mantener una estructura flexible sin perder el control sobre el avance del desarrollo.

Dentro de este marco, los roles de Scrum desempeñan un papel esencial en la organización del trabajo. El Product Owner actúa como la voz del cliente, siendo el responsable de traducir las necesidades del usuario en requerimientos funcionales que se organizan en el product backlog, el cual representa el conjunto de funcionalidades del sistema. Por su parte, el Scrum Master cumple la función de facilitador del proceso, asegurando que el equipo de desarrollo aplique correctamente la metodología y eliminando posibles impedimentos que afecten su desempeño. El Scrum Team está conformado por los profesionales encargados de la construcción del sistema, incluyendo analistas, diseñadores, programadores y testers, cuyo número puede variar según la complejidad y alcance del proyecto, generalmente entre dos y nueve integrantes.

Adicionalmente, se reconocen los roles auxiliares dentro del marco Scrum, entre los cuales destacan los stakeholders, quienes representan a los actores interesados en el proyecto y que contribuyen con sus requerimientos y retroalimentación constante durante el proceso de desarrollo (Figura 1). Su participación es clave para asegurar que el producto final se alinee con las expectativas institucionales y las necesidades reales del entorno en el que será implementado.

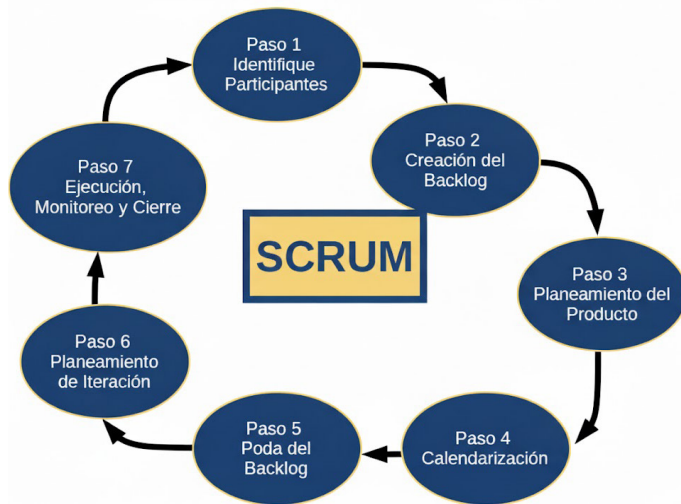


Figura 1. Funcionamiento Scrum.

Scrum integra una serie de elementos fundamentales que estructuran y organizan el proceso de desarrollo dentro de la cadena de trabajo, permitiendo una gestión eficiente, iterativa y orientada a resultados. En primer lugar, el Product Backlog constituye el repositorio principal de requerimientos del sistema, donde se registran todas las funcionalidades esperadas, acompañadas del feedback recopilado en las reuniones con los stakeholders. Este documento se mantiene en constante actualización, lo que garantiza su alineación con las necesidades cambiantes del proyecto y de los usuarios finales.

Por su parte, el Sprint Backlog representa la planificación operativa de cada ciclo de trabajo o sprint, ya que contiene la selección de tareas que el equipo de desarrollo se compromete a implementar en un periodo determinado. En este contexto, se establece que cada sprint debe organizarse dentro de un tiempo controlado, el cual permite mantener la eficiencia del equipo y asegurar entregas incrementales del producto. Esta planificación detallada facilita la organización interna del equipo y la distribución adecuada de las actividades técnicas, asegurando un avance progresivo y controlado del sistema.

Otro elemento clave dentro de Scrum es el Burn Down Chart, el cual se utiliza como herramienta de seguimiento visual del progreso del proyecto. Este gráfico permite representar de manera clara la cantidad de trabajo pendiente en relación con el tiempo disponible dentro de cada sprint, facilitando la identificación del avance real del equipo y permitiendo la toma de decisiones oportunas en caso de desviaciones en la planificación.

Una vez analizados estos elementos, es importante destacar que la metodología Scrum no establece una limitación estricta en cuanto al número de integrantes del equipo de desarrollo. Sin embargo, sí enfatiza la importancia de mantener equipos pequeños, autoorganizados y multidisciplinarios que garanticen una comunicación eficiente

y una entrega continua de valor, siempre siguiendo los procedimientos definidos por el marco de trabajo.

En otro orden de ideas, el sistema de gestión de datos constituye un componente esencial dentro de la arquitectura del sistema. Una base de datos no se limita únicamente al almacenamiento de información, sino que funciona como un sistema estructurado que garantiza la integridad, seguridad y disponibilidad de grandes volúmenes de datos de manera centralizada. Tradicionalmente, el modelo relacional o SQL ha sido el estándar predominante, basado en el uso del Lenguaje de Consulta Estructurado para organizar la información en tablas interrelacionadas mediante claves primarias y foráneas.

Este modelo permite optimizar la recuperación de datos y reducir la redundancia, facilitando una gestión eficiente de la información. Sin embargo, en el contexto actual, caracterizado por el crecimiento exponencial de los datos, su diversidad y velocidad de generación, el modelo relacional presenta limitaciones importantes. La rigidez de sus esquemas puede convertirse en un obstáculo para el rendimiento, especialmente en sistemas que requieren alta escalabilidad y flexibilidad. Además, su dependencia de arquitecturas centralizadas dificulta la expansión horizontal, lo cual es fundamental en aplicaciones modernas.

Como respuesta a estas limitaciones surge el modelo NoSQL, diseñado para manejar datos no estructurados o semi-estructurados con mayor flexibilidad. En este sentido, Kosinski (2025) señala que estas bases de datos “ofrecen una flexibilidad de esquema que permite que la estructura de almacenamiento se adapte dinámicamente a la naturaleza del dato, facilitando una escalabilidad horizontal eficiente” (p. 112); lo que las convierte en una alternativa adecuada para sistemas dinámicos y en constante evolución.

Bajo esta perspectiva, la arquitectura del sistema de bolsa de empleo debe contemplar una estructura capaz de adaptarse a cambios frecuentes en los perfiles de usuarios y en las ofertas laborales, lo que convierte la elección de tecnologías flexibles en una decisión estratégica más que técnica. En este sentido, Firebase se presenta como una plataforma integral desarrollada por Google que opera bajo un modelo serverless, lo que libera a los desarrolladores de la gestión directa de infraestructura, escalabilidad y seguridad.

Firestore se organiza en dos grandes categorías de servicios: Build y Run. Dentro de la categoría Build se encuentran sus bases de datos gestionadas, como Firestore Realtime Database y Cloud Firestore, ambas clasificadas como soluciones NoSQL orientadas a la sincronización en tiempo real y al manejo de estructuras de datos flexibles. Estas se complementan con servicios esenciales como autenticación de usuarios, almacenamiento en la nube y funciones serverless, lo que permite el desarrollo ágil de aplicaciones robustas y escalables. Por otro lado,

la categoría Run proporciona herramientas de monitoreo, análisis y optimización del rendimiento, consolidando así una plataforma integral para el desarrollo de sistemas modernos.

De igual manera, dentro del enfoque de desarrollo se considera la categoría **Run**, la cual se orienta a la ejecución, monitoreo y optimización del sistema una vez que ha sido desplegado en producción. Esta categoría incorpora herramientas como Crashlytics para la detección y análisis de fallos, Google Analytics para el estudio del comportamiento de los usuarios y Firebase Cloud Messaging (FCM) para la gestión de notificaciones. En conjunto, estos servicios permiten asegurar la mejora continua del sistema y su estabilidad operativa en el tiempo (Iones, 2025).

Esta plataforma también destaca por su capacidad de adaptarse a distintos tipos de soluciones tecnológicas, ya que permite almacenar y gestionar información compleja en múltiples contextos de aplicación. En este sentido, Campoverde (2024) señala que su uso resulta adecuado para escenarios de alta demanda y proyectos de gran escala, lo que la posiciona como una alternativa sólida dentro del desarrollo moderno de software.

Por otro lado, el formato JSON (JavaScript Object Notation) se ha consolidado como un estándar fundamental para el intercambio de información entre sistemas. A pesar de su origen en JavaScript, su principal ventaja radica en su independencia de lenguaje, ya que tecnologías como Python, Java, C# y C++ lo soportan de forma nativa. Esta característica permite una comunicación eficiente entre sistemas heterogéneos mediante una estructura ligera, sencilla y fácilmente interpretable.

JSON surge como una evolución natural frente a formatos más complejos como XML, destacándose por su simplicidad y facilidad de lectura tanto para humanos como para máquinas. En este contexto, IBM (2025) afirma que "JSON es un formato de intercambio de datos de texto ligero e independiente del lenguaje que permite la comunicación fluida entre sistemas heterogéneos". En el sistema propuesto, su implementación no solo responde a una necesidad técnica, sino también estratégica, ya que facilita la transmisión eficiente de datos entre el frontend, backend y servicios externos.

En cuanto a la arquitectura del software, el modelo vista controlador (MVC) es ampliamente utilizado para organizar la estructura de las aplicaciones, incluyendo entornos móviles y web. Este patrón divide el sistema en tres capas claramente diferenciadas: Modelo, Vista y Controlador, lo que permite una separación adecuada de responsabilidades y una mejor mantenibilidad del código.

El Modelo se encarga de la gestión de los datos del sistema, permitiendo operaciones como la creación, lectura, actualización y eliminación (CRUD) sobre la información almacenada. La Vista, por su parte, representa la interfaz de usuario, siendo responsable de la presentación de la información mediante tecnologías como HTML, JavaScript y CSS. Finalmente, el Controlador gestiona la lógica de negocio y actúa como intermediario entre el modelo y la vista, procesando las solicitudes del usuario y coordinando las respuestas del sistema (Servicio de Informática ASP.NET MVC 3 Framework, 2024).

Para la construcción de la interfaz de usuario se ha seleccionado React, una biblioteca de código abierto desarrollada por Meta que ha transformado el desarrollo de aplicaciones web modernas gracias a su arquitectura basada en componentes. Este enfoque permite dividir la interfaz en módulos reutilizables, cada uno con su propia lógica y estado, lo que mejora significativamente la escalabilidad y mantenibilidad del sistema.

Uno de los principales aportes de React es el uso del Virtual DOM, el cual optimiza el rendimiento al evitar modificaciones directas sobre el DOM real. En su lugar, React mantiene una representación en memoria y aplica únicamente los cambios necesarios, reduciendo el costo computacional de las actualizaciones de la interfaz. Asimismo, el uso de JSX facilita la creación de componentes al combinar JavaScript con una sintaxis similar a HTML, permitiendo una estructura más intuitiva y expresiva del diseño visual.

Finalmente, en el desarrollo del backend se considera el uso de Node.js, un entorno de ejecución de JavaScript diseñado para operar fuera del navegador, específicamente en el lado del servidor. Este entorno está construido sobre el motor V8 de Google, lo que le permite ejecutar código de manera altamente eficiente. Su arquitectura basada en eventos y su modelo de entrada/salida no bloqueante lo convierten en una tecnología ideal para aplicaciones escalables y con múltiples conexiones simultáneas.

Este motor compila el código JavaScript directamente a código máquina, lo que reduce la latencia y mejora significativamente el rendimiento, convirtiéndose en una de sus principales fortalezas. A esto se suma su modelo de entrada y salida asíncrono y no bloqueante, basado en un único hilo de ejecución y un bucle de eventos que gestiona las operaciones concurrentes de manera eficiente. Este enfoque permite manejar un gran número de conexiones simultáneas sin la necesidad de crear un hilo por cada solicitud, lo que lo convierte en una tecnología altamente eficiente para aplicaciones con múltiples peticiones de red o con interacción en tiempo real (Figura 2).

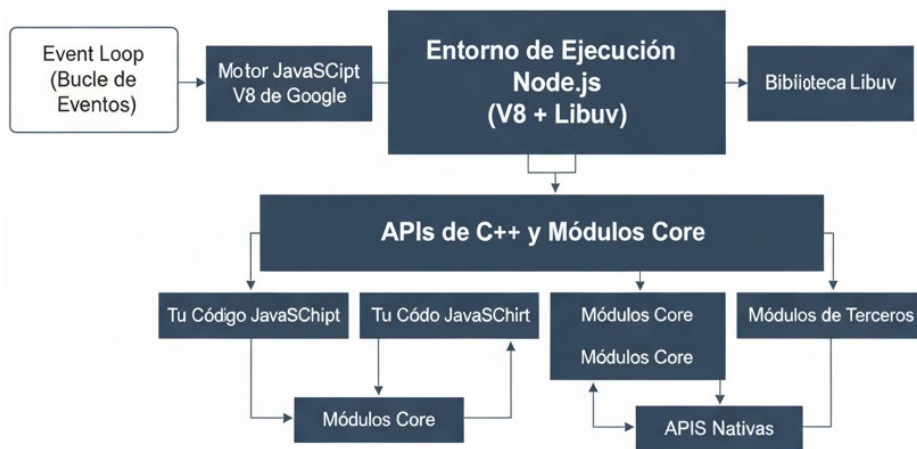


Figura 2. Arquitectura de Node.Js.

En cuanto a su arquitectura interna, Node.js integra varias capas fundamentales que permiten su funcionamiento eficiente. Entre ellas se encuentra el motor V8, encargado de la ejecución del código JavaScript; la biblioteca libuv, que abstrae las operaciones de entrada y salida del sistema operativo (como red y manejo de archivos); y una librería estándar de módulos escritos en JavaScript que proporciona funcionalidades comunes, tales como HTTP, sistema de archivos, buffers y streams.

De manera similar a React en términos de escalabilidad, Node.js también prioriza el rendimiento en sistemas concurrentes. Aunque opera bajo un único hilo principal, es capaz de aprovechar múltiples núcleos del procesador mediante su módulo *cluster*, el cual permite la creación de procesos hijos denominados “workers”. Estos procesos facilitan la distribución de la carga de trabajo, mejorando el rendimiento en aplicaciones de alta demanda. Adicionalmente, su amplio ecosistema de paquetes, gestionado a través de npm (Node Package Manager), permite acceder a miles de librerías reutilizables, desde frameworks como Express hasta herramientas especializadas, lo que lo convierte en una tecnología altamente utilizada en aplicaciones de red en tiempo real, donde la eficiencia y la baja latencia son factores críticos (Kruchten, 2023).

Por otro lado, el *web scraping* se define como una técnica de automatización orientada a la extracción estructurada de información desde sitios web mediante programas que navegan y analizan páginas web de forma sistemática. Según Hernández et al. (2022), esta técnica permite realizar solicitudes HTTP a los servidores, analizar el código HTML o XML de las páginas y extraer datos específicos como precios, descripciones o contactos. Su principal ventaja radica en la capacidad de recopilar grandes volúmenes de información en tiempos reducidos, siendo ampliamente utilizada en análisis de datos, monitoreo de competencia, estudios de mercado e indexación web. Este proceso sigue una secuencia metodológica que incluye la

identificación de URLs, el análisis del código fuente para localizar etiquetas relevantes, el desarrollo del *scraper* y el almacenamiento de la información en formatos estructurados como hojas de cálculo o bases de datos.

Dentro de los frameworks de automatización de navegadores, destaca Playwright, desarrollado por Microsoft. Aunque fue creado inicialmente para pruebas *end-to-end*, ha sido ampliamente adoptado para tareas de web scraping debido a su capacidad para interactuar con aplicaciones web dinámicas basadas en JavaScript. A diferencia de los scrapers tradicionales que solo procesan HTML estático, Playwright ejecuta un navegador real, lo que le permite navegar, interactuar con elementos, completar formularios, esperar cargas dinámicas y ejecutar acciones complejas, siendo especialmente útil en aplicaciones web modernas.

Asimismo, Playwright permite gestionar múltiples contextos de navegador de forma simultánea, ya que es compatible con los principales motores como Chromium, Firefox y WebKit, lo que facilita el scraping en distintos entornos. Incorpora además mecanismos de espera automática que sincronizan el script con la carga de los elementos, reduciendo errores asociados a tiempos de respuesta. Esto resulta particularmente ventajoso en aplicaciones con contenido asíncrono, donde la interacción con elementos dinámicos puede ser compleja. En este sentido, se destaca que “Playwright destaca por su capacidad de automatizar navegadores modernos y manejar aplicaciones de una sola página (SPA) complejas, permitiendo una interacción precisa con elementos que dependen de la ejecución dinámica de scripts” (Universidad Autónoma de Baja California, 2019).

Finalmente, una historia de usuario se define como una descripción breve y clara de una funcionalidad desde la perspectiva del usuario final. Generalmente se estructura mediante la fórmula: *Como [tipo de usuario], quiero [acción o funcionalidad] para [beneficio u objetivo]*.

Tabla 1. Historia de Usuario.

Campo	Historia de Usuario N°1
Número	1
Usuario	Administrador
Nombre historia	Ejecución del scraper para actualización de ofertas laborales
Prioridad en negocio	Alta
Riesgo en desarrollo	Medio
Puntos estimados	6
Iteración asignada	2
Programador responsable	Jair Castillo
Descripción	<ul style="list-style-type: none"> • El administrador debe poder ejecutar manualmente el proceso de scraping desde el panel administrativo. • Al ejecutar el scraper, el sistema debe obtener las ofertas laborales actualizadas desde las fuentes configuradas y almacenarlas en la base de datos. • El sistema debe mostrar el progreso o estado del proceso (por ejemplo, “en ejecución”, “completado” o “error”). • Una vez finalizado, las nuevas ofertas deben reflejarse automáticamente en el módulo de visualización de los usuarios.

Se recomienda implementar validaciones estrictas que eviten la duplicidad de datos, garantizando además que el proceso no interfiera con el funcionamiento normal del sistema ni con la experiencia de los usuarios durante su uso.

Al analizar en profundidad las historias de usuario, fue posible identificar con mayor precisión las prioridades funcionales que deben guiar el desarrollo del sistema de bolsa de empleo. De estas narrativas se desprende que los estudiantes requieren, como principal necesidad, filtros eficientes que permitan segmentar las ofertas por carrera, área de interés o modalidad, sin dejar de lado el acceso rápido y claro a la información detallada de cada vacante. Estas necesidades no son arbitrarias, sino que se alinean directamente con el objetivo central del proyecto: centralizar las oportunidades laborales y facilitar la toma de decisiones.

La organización y análisis de estas historias no representa únicamente una tarea metodológica, sino una base fundamental para delimitar con precisión el alcance del sistema, permitiendo comprender cómo cada funcionalidad, botón o vista responde a un flujo real de interacción del usuario. En este sentido, el estudio de las historias de usuario evidencia un desafío técnico importante: lograr un equilibrio entre funcionalidad robusta y simplicidad en la experiencia de uso.

Se identifica que el usuario no requiere únicamente visualizar listados de ofertas, sino acceder a información de manera eficiente, reduciendo al mínimo el esfuerzo de búsqueda. Este hallazgo refuerza la necesidad de implementar procesos optimizados y una interfaz limpia, orientada a la usabilidad. Al definir claramente los roles, responsabilidades y objetivos de cada historia, se garantiza que el usuario se convierta en el eje central del diseño del sistema.

De este modo, las historias de usuario se consolidan como la base estructural sobre la cual se construye el diseño funcional del prototipo. Cada requerimiento identificado se transforma en una funcionalidad específica, permitiendo un seguimiento detallado durante el desarrollo. Asimismo, estas historias no solo representan el estado actual del sistema, sino que también funcionan como una guía para futuras mejoras basadas en el comportamiento y necesidades de los usuarios. Los resultados obtenidos reflejan fielmente las dinámicas del público objetivo, permitiendo que el sistema se construya sobre una estructura coherente, ordenada y alineada con los objetivos planteados desde el inicio del proyecto (Figura 3).

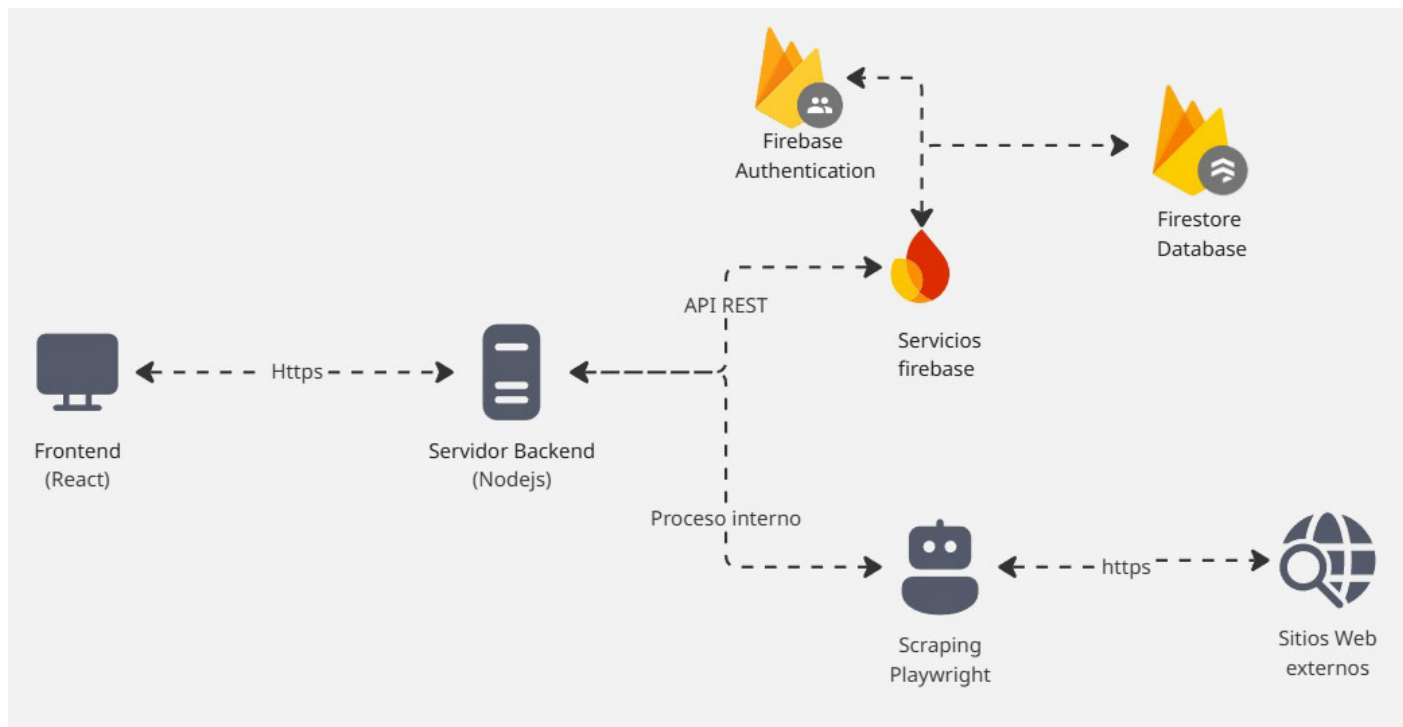


Figura 3. Diagrama de Despliegue.

El Diagrama de Despliegue presentado ilustra el ecosistema técnico diseñado para dar soporte a la bolsa de trabajo de la Universidad Iberoamericana del Ecuador (UNIB.E). En este esquema se evidencia la interacción integrada entre el frontend, el backend y los distintos servicios en la nube. Al tratarse de una aplicación completamente web, la plataforma permite el acceso desde cualquier navegador, donde React desempeña un papel fundamental al construir una interfaz dinámica, responsiva y capaz de adaptarse de forma fluida a las interacciones del usuario.

El intercambio de información entre la capa de presentación y el servidor se realiza a través de una API desarrollada en Node.js. Este componente se encarga de procesar las solicitudes, ejecutar la lógica de negocio y gestionar el módulo responsable de la recolección automatizada de ofertas laborales, garantizando eficiencia en el tratamiento de la información. En lo referente a la seguridad y el control de acceso, esta responsabilidad es delegada a Firebase Authentication, servicio que permite validar credenciales de usuario y proteger el acceso a las funcionalidades administrativas, asegurando la integridad del sistema frente a accesos no autorizados.

Por otra parte, el almacenamiento y la gestión de datos se centralizan en Firebase, plataforma en la que se administran tanto las búsquedas como el historial de ofertas y los registros generados por el sistema. El uso de esta tecnología garantiza la disponibilidad inmediata de la información, así como su persistencia y seguridad. Además, al operar bajo una infraestructura en la nube, el sistema adquiere una alta capacidad de escalabilidad, permitiendo su crecimiento progresivo sin afectar su rendimiento conforme aumenta el número de usuarios o el volumen de datos.

Este diseño arquitectónico se caracteriza por una clara separación de responsabilidades entre sus componentes, lo cual contribuye significativamente a la optimización del rendimiento, la facilidad de mantenimiento y el fortalecimiento de la seguridad del sistema. En este sentido, el Diagrama de Despliegue evidencia cómo la integración de React, Node.js y Firebase conforma una solución tecnológica robusta, escalable y eficiente. Esta sinergia no solo garantiza la estabilidad del sistema web, sino que también constituye la base para el cumplimiento de los objetivos establecidos en el desarrollo del prototipo de bolsa de trabajo universitaria (Figuras 4, 5, 6 y 7).

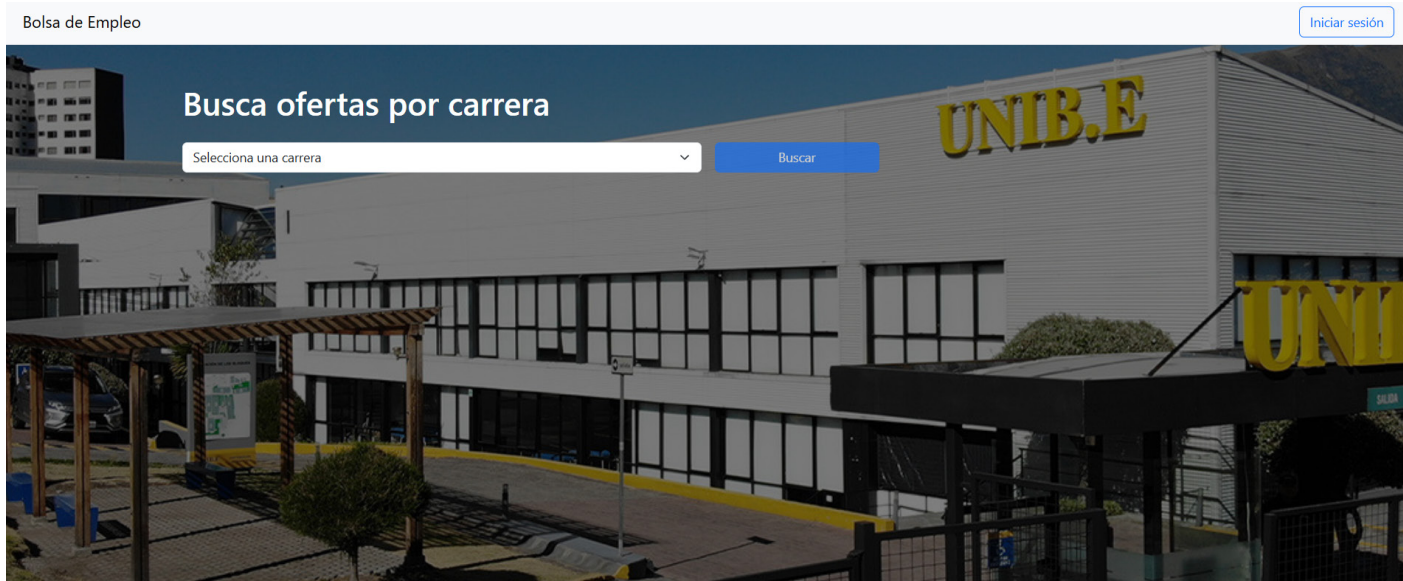


Figura 4. Pantalla de aplicación.

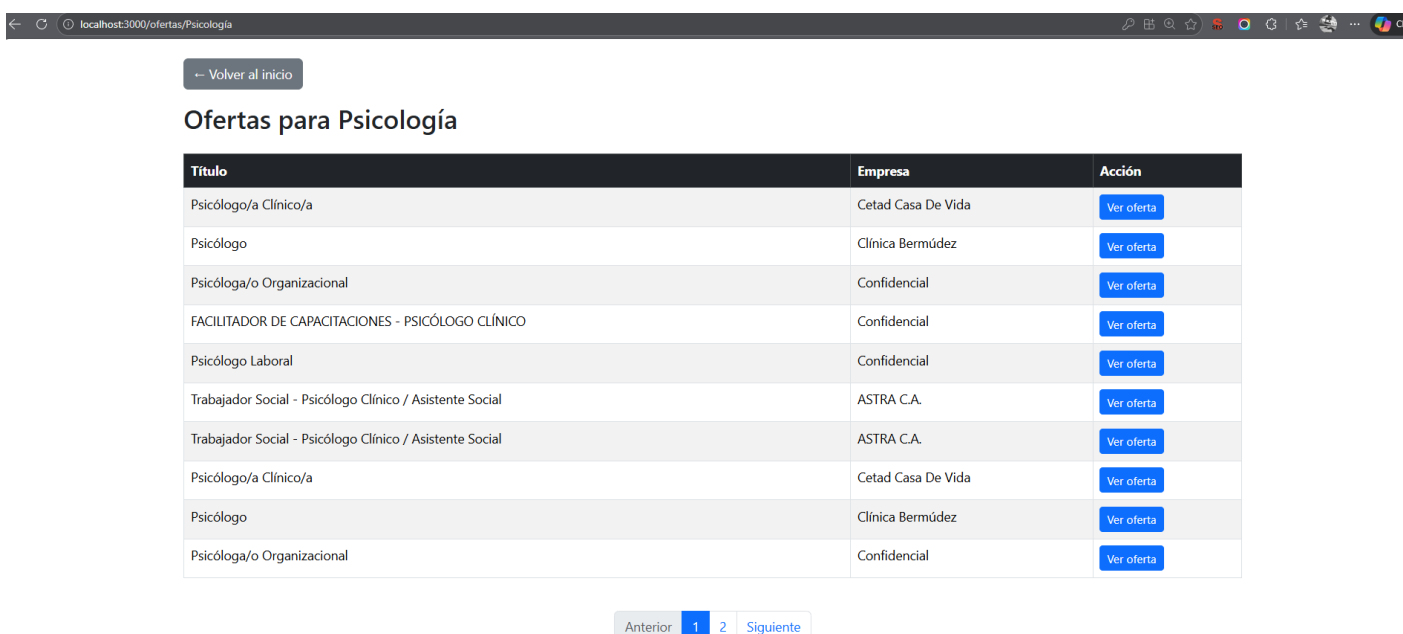


Figura 5. Interfaz de presentación de ofertas.

Ingreso Administrador

Correo

Contraseña

[Ingresar](#)

Figura 6. Interfaz de Login.



Figura 7. Reporte de Ofertas.

En el plano técnico, la solidez de la plataforma descansa en un ecosistema web moderno, donde se combina un frontend en React con un backend robusto en Node.js, cerrando el círculo con los servicios en la nube de Firebase; lo cual no es algo arbitrario; ya que es lo que nos otorga la estabilidad y escalabilidad necesaria para que la herramienta pueda crecer y sumar funciones en el futuro sin complicaciones. Además, delegar la seguridad en Firebase Authentication nos da un control de acceso sumamente confiable, mientras que su base de datos garantiza que la información esté siempre disponible y persistente en tiempo real.

En el apartado económico, el proyecto es factible por una fundamental razón, la mayoría de las herramientas seleccionadas son de software libre o cuentan con planes gratuitos, lo que apoya los costos de licenciamiento, así la inversión se concentra realmente donde aporta valor, que es en el talento humano y las horas de desarrollo. A largo plazo, el sistema se traduce en un ahorro operativo real para la universidad, ya que automatiza procesos que antes se gestionaban manualmente, liberando recursos y optimizando el tiempo institucional.

CONCLUSIONES

Las conclusiones del presente estudio evidencian que, a partir del levantamiento de información y del análisis detallado de las historias de usuario, se logró identificar y consolidar un conjunto de funcionalidades esenciales para el sistema de bolsa de trabajo. Entre estas destacan la visualización de ofertas laborales segmentadas por carrera, la autenticación segura del administrador, la actualización automática de las vacantes y la generación de reportes. Estas funcionalidades responden directamente a las necesidades identificadas en el contexto institucional, garantizando una solución orientada a la eficiencia y a la mejora de la experiencia del usuario.

En cuanto al diseño arquitectónico del sistema, se concluye que la estructura propuesta permite organizar una solución tecnológica robusta, clara y escalable, fundamentada en el modelo cliente-servidor. Esta arquitectura no solo facilita la distribución adecuada de las responsabilidades entre los componentes del sistema, sino que también fortalece aspectos críticos como la seguridad, el rendimiento y la adaptabilidad ante futuros cambios o ampliaciones funcionales.

Asimismo, se determina que la correcta gestión de la información y la comunicación entre los distintos componentes del sistema se logra mediante una arquitectura modular bien definida. La integración de un frontend desarrollado en React, un backend implementado en Node.js y el uso de Firebase como servicio de almacenamiento y autenticación permite establecer un ecosistema tecnológico coherente, eficiente y altamente interoperable, optimizando los procesos internos del sistema.

Por otro lado, el análisis del diagrama de despliegue confirma una adecuada distribución de los recursos en la nube, lo cual garantiza la disponibilidad continua del sistema, su mantenibilidad y su eficiencia operativa. Esta configuración permite que la plataforma pueda responder de manera adecuada a la demanda de usuarios, manteniendo estabilidad y rendimiento en el tiempo.

La implementación del web scraping como mecanismo de recolección de datos contribuye significativamente a la automatización del sistema, al permitir la obtención de ofertas laborales desde fuentes externas de manera constante y actualizada. Este proceso reduce la carga operativa manual, mejora la eficiencia del sistema y asegura la actualización permanente de la información disponible para los usuarios.

REFERENCIAS

- Cortes, A. (2019). Una nueva tecnología pretende transformar el cáncer de pulmón en una enfermedad crónica. *El País*. https://elpais.com/elpais/2019/11/08/ciencia/1573214337_571170.html
- Hernández, L., Acosta, M., Zavaleta, E., & Domínguez, D. (2022). Sistema de seguimiento de egresados como herramienta del desarrollo institucional. *Revista de Investigación*, 14(2), 19–28. <https://progmataeem.mx/progmataeem/index.php/progmataeem/article/view/2022-14-2-03/2022-14-2-03>
- IBM. (2025). *Logical data models*. <https://www.ibm.com/docs/es/ida/9.1.2?topic=modeling-logical-data-models>
- IONOS. (2025). *Firebase: características y funcionamiento*. <https://www.ionos.mx/digitalguide/servidores/how/firebase/>
- Kosinski, J. (2025). *Advanced SQL optimization techniques*. Springer.
- Kruchten, P. K. (2023). *The rational unified process made easy*. Pearson.
- Organización de las Naciones Unidas. (2024). *ODS 8: Trabajo decente y crecimiento económico*. Pacto Mundial de la ONU. <https://www.pactomundial.org/ods/8-trabajo-decente-y-crecimiento-economico/>
- Schwaber, K., & Sutherland, J. (2024). *The Scrum guide*. Scrum.org / Pearson.
- Sommerville, I. (2020). *Software engineering* (10th ed.). Pearson Education Limited.
- Universidad Autónoma de Baja California. (2019). *RUP: Rational Unified Process*. <http://yaqui.mx/uabc/~molquin/as/RUP.htm>
- Universidad de Alicante. (2024). *Modelo Vista Controlador (MVC): documentación ASP.NET MVC 3*. <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/mo-vo-vista-controlador-mvc.html>

Conflictos de interés:

Los autores declaran no tener conflictos de interés.

Contribución de los autores:

Tonyse de la Rosa-Martín, Jair Castillo: Conceptualización, curación de datos, análisis formal, investigación, metodología, supervisión, validación, visualización, redacción del borrador original y redacción, revisión y edición.

Declaración ética:

El estudio se basó en el análisis de fuentes documentales y datos de acceso público, por lo que no implicó la participación directa de seres humanos. No se manejó información personal identificable.