



Ciencia Latina
Internacional

Ciencia Latina Revista Científica Multidisciplinaria, Ciudad de México, México.

ISSN 2707-2207 / ISSN 2707-2215 (en línea), mayo-junio 2024,

Volumen 8, Número 3.

https://doi.org/10.37811/cl_rcm.v8i3

ANÁLISIS DE FRAMEWORKS FRONTEND PARA APLICAR UX/IU EN EL DESARROLLO WEB: UNA REVISIÓN SISTEMÁTICA

**ANALYSIS OF FRONTEND FRAMEWORKS FOR APPLYING UX/UI
IN WEB DEVELOPMENT: A SYSTEMATIC REVIEW**

Jessica Patricia Tituaña Sangucho

Universidad Técnica de Ambato, Ecuador

José Luis Ojeda Carrasco

Universidad Técnica de Ambato, Ecuador

Jean Pierre Girón Rodríguez

Universidad Técnica de Ambato, Ecuador

Dennis Steven Bonilla Granja

Universidad Técnica de Ambato, Ecuador

Santiago Jara Moya

Universidad Técnica de Ambato, Ecuador

DOI: https://doi.org/10.37811/cl_rcm.v8i3.11290

Análisis de Frameworks Frontend para Aplicar UX/IU en el Desarrollo Web: Una Revisión Sistemática

Jessica Patricia Tituaña Sangucho¹

jesifv1999@gmail.com

<https://orcid.org/0009-0000-7688-797X>

Universidad Técnica de Ambato
Ecuador

Jean Pierre Girón Rodríguez

jean_0720@hotmail.com

<https://orcid.org/0009-0005-6836-389X>

Universidad Técnica de Ambato
Ecuador

Santiago Jara Moya

sd.jara@uta.edu.ec

<https://orcid.org/0000-0002-4360-6008>

Universidad Técnica de Ambato
Ambato-Ecuador

José Luis Ojeda Carrasco

jojeda5171@gmail.com

<https://orcid.org/0009-0004-0114-5241>

Universidad Técnica de Ambato
Ecuador

Dennis Steven Bonilla Granja

dennissbonilla@hotmail.com

<https://orcid.org/0009-0003-3030-1496>

Universidad Técnica de Ambato
Ecuador

RESUMEN

Basado en una revisión sistemática de frameworks frontend para aplicaciones web, este artículo presenta un análisis exhaustivo de varios frameworks populares, evaluando sus características, ventajas y desventajas en términos de experiencia del usuario (UX) y la interfaz de usuario (UI). La metodología utilizada incluyó una búsqueda exhaustiva en bases de datos académicas utilizando criterios específicos, seleccionando estudios que compararon al menos dos frameworks frontend prominentes. Se examinaron diversas métricas, como funcionalidad, confiabilidad, usabilidad, eficiencia y portabilidad, empleando normas ISO y metodologías como QSOS e IQMC. Los hallazgos presentados resaltan aspectos críticos sobre la calidad y rendimiento de frameworks como Spring, Django, Angular, React, Vue.js, Laravel, entre otros. Se proporciona una visión comparativa detallada, ofreciendo orientación clave para desarrolladores y diseñadores en la selección de herramientas adecuadas para optimizar la experiencia del usuario y la interfaz en proyectos web. Este análisis contribuye significativamente al campo del desarrollo web al identificar los puntos fuertes y débiles de los frameworks frontend más utilizados, ayudando a los profesionales a tomar decisiones informadas sobre la implementación de tecnologías que mejoren la calidad de la UX/UI en aplicaciones web.

Palabras clave: framework, frontend, UX, UI, desarrollo web

¹ Autor principal

Correspondencia: jesifv1999@gmail.com



Analysis of Frontend Frameworks for Applying UX/UI in Web Development: A Systematic Review

ABSTRACT

This article presents a comprehensive analysis of various popular frontend frameworks based on a systematic review, evaluating their features, advantages, and disadvantages in terms of user experience (UX) and user interface (UI). The methodology involved an exhaustive search across academic databases using specific criteria, selecting studies that compared at least two prominent frontend frameworks. Various metrics such as functionality, reliability, usability, efficiency, and portability were examined, employing ISO standards and methodologies such as QSOS and IQMC. The findings highlighted critical aspects regarding the quality and performance of frameworks like Spring, Django, Angular, React, Vue.js, Laravel, among others. A detailed comparative overview is provided, offering key guidance for developers and designers in selecting suitable tools to optimize user experience and interface in web projects. This analysis significantly contributes to the field of web development by identifying strengths and weaknesses of the most widely used frontend frameworks, aiding professionals in making informed decisions on implementing technologies that enhance the quality of UX/UI in web applications.

Keywords: framework, frontend, UX, UI, web development

Artículo recibido 10 abril 2024

Aceptado para publicación: 15 mayo 2024



INTRODUCCIÓN

En el contexto del desarrollo web, la atención hacia la experiencia del usuario (UX) y la interfaz de usuario (UI) ha alcanzado un nivel crítico. Las interfaces web no solo deben ser funcionales y eficientes, sino también atractivas, intuitivas y altamente satisfactorias para el usuario. En este escenario, los frameworks frontend juegan un papel crucial al proporcionar a los desarrolladores las herramientas necesarias para construir interfaces web de alta calidad de manera eficiente.

Los frameworks frontend ofrecen una estructura sólida y conjuntos de herramientas predefinidas que permiten a los desarrolladores agilizar el proceso de desarrollo, manteniendo estándares de diseño y usabilidad. Sin embargo, la abundancia de opciones disponibles presenta un desafío significativo: la selección del framework más adecuado para un proyecto específico.

Esta revisión sistemática se propone analizar los frameworks frontend más populares utilizados para mejorar la UX/UI en el desarrollo web. Se ha llevado a cabo una búsqueda exhaustiva en bases de datos académicas y motores de búsqueda utilizando palabras clave pertinentes, seleccionando artículos publicados en revistas científicas y repositorios académicos de renombre. Los criterios de selección se centran en la calidad metodológica y la relevancia de los estudios para el tema de investigación.

Por tanto, en este artículo exploraremos a fondo las características, ventajas y desventajas de los principales frameworks frontend, con el objetivo de proporcionar una guía fundamentada para los desarrolladores al momento de elegir y aplicar estas herramientas en sus proyectos web. Este análisis crítico servirá como recurso invaluable para optimizar la experiencia del usuario y la interfaz en el desarrollo web.

METODOLOGÍA

Para esta investigación, se realizó una búsqueda de información en bases de datos científicas como IEEE Xplore Digital Library, Google Scholar y Science Direct. Para estructurar esta búsqueda, se definieron términos clave como "Análisis", "Comparación", "Framework", "Frontend", "UI", y "UX" lo que permitió crear cadenas de búsqueda específicas como Análisis AND (UI OR UX) AND "framework frontend", Comparativa AND "framework frontend" y (Comparativa OR Análisis) AND "frameworks frontend" para discretizar los contenidos dentro de estas bases de datos. Además, se implementaron criterios de inclusión y exclusión que requerían artículos publicados desde 2015 en adelante, que



compararan al menos dos frameworks de frontend y que incluyan información relacionada con la experiencia de usuario (UX) o la interfaz de usuario (UI). Esto garantizó que los artículos recuperados fueran realmente útiles para la investigación. Finalmente, se realizó un análisis de relevancia de los artículos para determinar cuáles serían utilizados en el estudio. Para obtener más detalles sobre el enfoque definido, consulte la Figura 1.

Con los artículos seleccionados, se realizó una extracción de información relevante, detallada a continuación:

Mejía Heber en su publicación se basó en el uso de herramientas como el Modelo de Construcción de Calidad Individual (IQMC) y las características de calidad propuestas en la norma ISO 25010n. Se aplicaron metodologías como el QSOS para definir características comparativas del software de código libre. Las técnicas de adquisición de datos incluyeron la revisión de documentación, la evaluación práctica y la aplicación de modelos de calidad. En la tabla 1 se presenta un análisis detallado de cuatro frameworks frontend. Los frameworks evaluados son Spring, Struts, JSF y Angular JS. Los resultados muestran que, en términos de funcionalidad, Struts y JSF tienen un desempeño similar, superando ligeramente a Spring y Angular JS. En cuanto a confiabilidad, todos los frameworks obtienen puntajes muy bajos, lo que sugiere áreas de mejora significativas en este aspecto. En usabilidad, Spring se destaca ligeramente sobre los demás, mientras que Angular JS muestra el menor puntaje. En eficiencia, mantenimiento y portabilidad, los frameworks tienen un rendimiento similar, con diferencias marginales entre ellos. Este análisis general revela que Spring fue el mejor framework evaluado en términos de calidad en el contexto del análisis de UX/UI para el desarrollo web. Estos resultados proporcionan una visión comparativa importante para desarrolladores y diseñadores en la elección de un framework adecuado para aplicaciones web con enfoque en UX/UI. (Mejia et al., 2015)

Espinosa Hurtado mediante su investigación empleó herramientas como el modelo de evaluación de la norma ISO/IEC 25000 para establecer criterios de evaluación de los frameworks Laravel y Django. Además, se utilizaron metodologías analíticas y descriptivas para comparar las características y prestaciones de ambos frameworks. En la tabla 2 se presenta un análisis comparativo de estos frameworks, haciendo uso de la metodología XP y la herramienta Google Page Speed Insights. Los resultados revelan que Django sobresale en varios aspectos clave en comparación con Laravel. En



términos de usabilidad, Django obtiene un puntaje ligeramente superior al de Laravel. En portabilidad, Django muestra un rendimiento significativamente mejor que Laravel, con un puntaje del 88,52% frente al 66,30% de Laravel. En cuanto a rendimiento, nuevamente Django supera a Laravel con un puntaje del 83,10% en comparación con el 61,20% de Laravel. Curiosamente, en seguridad, ambos frameworks obtienen el mismo puntaje del 88,90%. Este análisis general indica que Django resultó ser el mejor framework evaluado en términos de calidad y rendimiento, destacando su capacidad para abordar aspectos críticos como usabilidad, portabilidad y rendimiento en el desarrollo web. Estos hallazgos ofrecen valiosas perspectivas para los profesionales de desarrollo en la selección de frameworks adecuados para proyectos web con un enfoque en la experiencia del usuario y la interfaz de usuario. (Espinosa Hurtado, 2021)

En la publicación “Research and Analysis of the Front-end Frameworks and Libraries in E-Business Development” los autores hicieron uso de una combinación de herramientas y metodologías para analizar y comparar React, Angular y Vue. Las técnicas de adquisición de datos incluyeron la recopilación de datos de uso de GitHub para medir las preferencias de los desarrolladores frontend globales. En la tabla 3 se presenta un análisis comparativo de los frameworks, se consideraron aspectos como el tipo de procesamiento de datos (unidireccional o bidireccional) y el rendimiento en términos de Ready Memory y Run Memory en segundos. Los resultados muestran que Angular 2 se destaca como el mejor framework en términos generales. En cuanto al procesamiento de datos, Angular 2 y Vue JS ofrecen capacidades tanto unidireccionales como bidireccionales, lo que les otorga una versatilidad en comparación con React (unidireccional) y Angular 1 (bidireccional). En cuanto al rendimiento, Angular 2 muestra un mejor desempeño en ambas métricas de memoria (Ready Memory y Run Memory) en comparación con React y Vue JS, con valores más bajos que indican una mejor eficiencia. Análisis general resalta la superioridad de Angular 2 como el framework más completo y eficiente en términos de procesamiento de datos y rendimiento de memoria. Estos hallazgos son relevantes para desarrolladores y diseñadores web que buscan seleccionar el framework más adecuado para proyectos que requieren un procesamiento de datos flexible y un rendimiento eficiente. (Xing et al., 2019)



En la investigación de Chávez se empleó la metodología cuantitativa y experimental permitió comparar el diseño web adaptativo y responsivo, mientras que el método Delphi se utilizó para consultar a expertos en el campo. En la tabla 4 se presenta un análisis comparativo entre dos marcos de diseño, responsivo y adaptativo con una población de 50 docentes. Los resultados muestran que el marco de diseño responsivo fue preferido por la mayoría de los docentes en términos de simplicidad y rapidez, adaptabilidad de las interfaces, reducción de código, originalidad de las interfaces y mejora del rendimiento. Específicamente, en cuanto a simplicidad y rapidez, 40 docentes escogieron el marco de diseño responsivo mientras que 10 prefirieron el adaptativo. Para adaptabilidad de las interfaces, 30 docentes optaron por el marco de diseño responsivo y 20 por el adaptativo. En términos de reducción de código, originalidad de las interfaces y mejora del rendimiento, la preferencia por el marco de diseño responsivo fue aún más marcada, con 43, 34 y 41 docentes respectivamente escogiéndolo, en comparación con 7, 16 y 9 docentes que prefirieron el marco adaptativo. Estos resultados sugieren que el marco de diseño responsivo es ampliamente preferido por la población de docentes evaluados, destacándose en múltiples aspectos clave para el desarrollo de interfaces web eficientes y efectivas. Estos hallazgos ofrecen información valiosa para profesionales del diseño web y desarrollo de interfaces, respaldando la adopción y uso continuo del marco de diseño responsivo en proyectos y aplicaciones web. (Chavez Calderón et al., 2022)

Kaluža y Vukelić en su investigación explican cómo se compararon los frameworks frontend Vue.js, React.js y Angular, considerando varios factores. En la tabla 5 se presenta un análisis comparativo estos frameworks utilizando el método de Evaluación de Componentes con Preguntas Ponderadas y la herramienta GitHub. Los resultados muestran que en la categoría de Multi Pages, Vue JS obtiene la puntuación más alta con 8 puntos, seguido de cerca por Angular con 2 puntos y React JS con 5,5 puntos. En la categoría de Single Page, Angular lidera con 7 puntos, seguido por Vue JS con 6 puntos y React JS con 4,5 puntos. Este análisis revela que Vue JS sobresale en la capacidad de manejar múltiples páginas, mientras que Angular muestra fortaleza en la creación de páginas únicas. Por otro lado, React JS obtiene puntajes intermedios en ambas categorías. Estos hallazgos son significativos para desarrolladores y diseñadores web, ya que proporcionan información clave sobre las capacidades y



limitaciones de cada framework en diferentes contextos de desarrollo de aplicaciones web.(Kaluža & Vukelić, 2018)

Luzuriaga Mendoza en su investigación analiza los factores que influyen en la selección de un lenguaje y framework de programación para el desarrollo de software web. Se destaca la importancia de elegir las herramientas adecuadas para garantizar un desarrollo eficiente y satisfacer las expectativas de los usuarios. Se realizó una investigación en la ciudad de Machala, donde se encuestaron a empresas de desarrollo de software y desarrolladores independientes para identificar los factores relevantes en la selección de tecnologías. En la tabla 6 comparan los frameworks mencionados en el documento, se observaron diferencias significativas en términos de rendimiento, flexibilidad, robustez y curva de aprendizaje. Por ejemplo, en el caso de los frameworks Java Spring y Java Struts, ambos ofrecen un alto rendimiento y son flexibles y robustos, pero pueden tener diferencias en la curva de aprendizaje. En el caso de los frameworks PHP Laravel y PHP Symfony, se destacó la facilidad de uso de Laravel y la modularidad de Symfony, lo que los hace más adecuados para diferentes tipos de proyectos. (Luzuriaga Mendoza, 2020)

En (Ollila et al., 2022) se explica el estudio de rendimiento de renderizado de los frameworks frontend, comparando las estrategias de renderizado utilizadas por cada uno. Se encontró que las diferencias en estas estrategias pueden implicar que algunos frameworks realicen varias órdenes de magnitud más trabajo que otros en ciertas circunstancias, afectando significativamente al rendimiento. Blazor, siendo el único framework basado en WebAssembly revisado, mostró un rendimiento significativamente peor que sus competidores basados en JavaScript, incluido React, que tiene un modelo de renderizado idéntico. También se destaca que las APIs del DOM son propensas a errores, contribuyendo hasta en un 80% de los errores de alto impacto en las aplicaciones web debido a errores en la manipulación del DOM, lo que puede afectar la experiencia del usuario. En la tabla 7, se enfoca en comparar los diferentes enfoques de renderizado utilizados por Angular, React, Vue, Svelte y Blazor, destacando la importancia de factores como el tamaño y la forma del árbol de componentes, así como la proporción de contenido estático a dinámico en los componentes, observando así que la valoración de frameworks depende fielmente de los requerimientos solicitados. (Ollila et al., 2022)



En (Vyas, 2022) se centra en la comparación de tres frameworks de front-end para aplicaciones web: Vue, React y Angular. Se discuten las ventajas y desventajas de cada uno, así como sus características distintivas. Se destaca la importancia de elegir el framework adecuado para el desarrollo de aplicaciones web, considerando factores como la arquitectura, la popularidad, el rendimiento y la sintaxis del código. En la tabla 8 muestra la arquitectura involucrada, las características de cada uno y el rendimiento en términos de manipulación del DOM y enlace de datos. Por lo que se puede considerar que dentro de la Arquitectura: Angular sigue el patrón MVC (Modelo-Vista-Controlador), lo que lo hace robusto y estructurado. React se basa en componentes, lo que permite una mayor flexibilidad y reutilización. Vue es un framework progresivo que ofrece un equilibrio entre Angular y React. Considerando también las características: Angular es conocido por su amplio conjunto de herramientas y funcionalidades, lo que lo hace ideal para aplicaciones complejas. React destaca por su eficiencia en el renderizado y su flexibilidad. Vue es el más ligero de los tres y es fácil de aprender, lo que lo hace atractivo para proyectos más pequeños. Dentro del rendimiento: Angular utiliza el Real DOM, lo que puede afectar el rendimiento al actualizar toda la estructura del árbol DOM. React y Vue utilizan Virtual DOM, lo que permite actualizaciones selectivas y mejora el rendimiento en comparación con Angular. Fijándonos también en la Popularidad: React es el framework más popular de los tres, seguido de Angular. Vue está experimentando un crecimiento en popularidad, especialmente por su facilidad de uso y buen rendimiento. En base a esto podemos enfatizar la importancia de elegir el framework adecuado para el desarrollo de aplicaciones web y se identifican criterios para la selección de un framework específico. (Vyas, 2022)

Chalan Jonathan se enfoca en el análisis comparativo de los frameworks PHP: Laravel y CodeIgniter para la implementación del Sistema de Gestión de Concursos de Méritos y Oposición en la Universidad Nacional de Chimborazo. Se realizan pruebas y mediciones de productividad en ambos frameworks. En la tabla 9 se observa que Laravel: Ofrece una filosofía clara que promueve un código expresivo y elegante. Utiliza Eloquent ORM para operaciones en la base de datos y es modular y extensible, permitiendo la incorporación de paquetes adicionales. Su sistema de enrutamiento HTTP es eficiente y cuenta con HTTP Middleware para filtrar las llamadas HTTP. Por otro lado, CodeIgniter: Destaca por su versatilidad, facilidad de instalación y flexibilidad. Es conocido por su ligereza y capacidad de



trabajar en diversos entornos. Ofrece una arquitectura flexible, lo que facilita la curva de aprendizaje. Su acoplamiento y singularidad de componentes contribuyen a la autonomía y reutilización del código. Sin embargo, dentro de la arquitectura: Ambos frameworks siguen el patrón Modelo-Vista-Controlador (MVC), lo que facilita la separación de la lógica de negocio de la presentación. Laravel y CodeIgniter ofrecen una estructura organizada y modular que mejora la mantenibilidad y escalabilidad de las aplicaciones desarrolladas con ellos. Los resultados indican que Laravel supera a CodeIgniter en términos de productividad, cumpliendo con el 100% de los parámetros establecidos en el análisis, en comparación con el 68.86% de cumplimiento de CodeIgniter. La elección entre ambos frameworks dependerá de las necesidades y objetivos específicos del proyecto. (Chalan Jonathan, 2016)

Mosquera Coronel analiza de manera comparativa los frameworks ZK y Laravel para el desarrollo de aplicaciones web y móviles. Se destaca la importancia de elegir el marco de desarrollo adecuado para satisfacer las necesidades de los desarrolladores y del proyecto en cuestión. Se menciona que se utilizó una metodología de investigación cualitativa, con recursos como documentación, recopilación bibliográfica y libros, siguiendo los lineamientos de la línea de investigación de Sistemas de Información

y Comunicación, Emprendimiento e Innovación. En la tabla 10 muestra el estudio comparativo entre los frameworks ZK y Laravel, se observa que ZK es un framework de interfaz de usuario que permite el desarrollo de aplicaciones web y móviles sin la necesidad de aprender JavaScript o AJAX. Por otro lado, Laravel es un framework de código abierto para programadores de aplicaciones web en PHP 5, con una sintaxis simple y elegante. Se destaca que ZK ofrece un desarrollo ágil sin la necesidad de JavaScript, mientras que Laravel permite una alta velocidad de codificación en la vista mediante el uso del motor de plantillas Blade. Además, se menciona que ZK se basa en una arquitectura monolítica con componentes del lado del servidor, mientras que Laravel ofrece la posibilidad de desarrollar aplicaciones basadas en microservicios. En cuanto a la seguridad de las aplicaciones, se señala que ZK proporciona una protección y seguridad adecuadas, al igual que Laravel. Ambos frameworks permiten la creación de PWAs (Aplicaciones Web Progresivas) y ofrecen una experiencia de usuario satisfactoria. (Mosquera Coronel, 2022)



En la investigación de Ilievská & Gramatikov se comparan tres de los frameworks más populares para el desarrollo de aplicaciones web de front-end: Angular, React.js y Vue.js, evaluando el rendimiento, la modularidad y la usabilidad de una aplicación de página única (SPA) creada con cada uno de ellos mediante pruebas exhaustivas utilizando la herramienta Lighthouse y el proceso de jerarquía analítica (AHP) para obtener una calificación general. En la tabla 11 los resultados muestran que React.js sobresalió en rendimiento, obteniendo los mejores puntajes en métricas clave como First Contentful Paint, Speed Index y Largest Contentful Paint, y también destacó en modularidad gracias a su robusto ecosistema de paquetes NPM y su flexibilidad, mientras que Vue.js obtuvo la puntuación más alta en usabilidad con una excelente documentación y una curva de aprendizaje más sencilla. En contraste, Angular tuvo un desempeño más deficiente en términos de rendimiento y modularidad, aunque se destacó en la manipulación del DOM. La investigación concluyó que, para el desarrollo de una aplicación SPA simple, React.js es la opción más adecuada en términos generales, seguido de cerca por Vue.js, quedando Angular rezagado en esta comparación específica, aunque los resultados pueden variar dependiendo de la complejidad de la aplicación y los requisitos del proyecto. (Ilievská & Gramatikov, 2022)

En la investigación de Aggarwal & Verma, se realiza una comparación crítica entre los frameworks MEAN y MERN, destacando las características de cada uno de los componentes que los conforman. Mientras que MEAN utiliza AngularJS para el desarrollo del front-end, MERN emplea ReactJS, el cual ofrece un mejor rendimiento y una arquitectura más flexible. Sin embargo, AngularJS proporciona una estructura más sólida y un modelo más robusto. Express.js y Node.js son compartidos por ambos frameworks, brindando una plataforma eficiente para el back-end. MongoDB, la base de datos NoSQL, también es utilizada en ambos casos, ofreciendo escalabilidad y flexibilidad en el manejo de datos. Aunque ambos frameworks son poderosos y ampliamente utilizados, en la tabla 12 se concluye que MERN podría ser la opción más adecuada para el desarrollo de aplicaciones web, gracias a la superioridad de ReactJS en términos de rendimiento, abstracción y empaquetado, lo que facilita el desarrollo y el mantenimiento del código. (Aggarwal & Verma, 2018)

Ghimire en su investigación compara entre los frameworks de Python para desarrollo web, como Flask, Django, Pyramid, Bottle, CherryPy y Tornado, revela una diversidad de opciones con características



distintivas. En la tabla 13 se detalla que Flask destaca por su enfoque ligero y flexible, ideal para proyectos pequeños que requieren una rápida implementación y una alta personalización. Por otro lado, Django emerge como la elección preferida para aplicaciones web complejas, gracias a su extensa gama de funcionalidades integradas y su enfoque en la escalabilidad. Sin embargo, se debe tener en cuenta que Pyramid, Bottle, CherryPy y Tornado también ofrecen ventajas específicas según las necesidades del proyecto. La decisión final entre estos frameworks depende de factores como el tamaño y la complejidad del proyecto, las preferencias de desarrollo del equipo y los plazos establecidos. En este contexto, mientras Flask destaca por su simplicidad y control detallado, Django se posiciona como una solución integral para proyectos de mayor envergadura, proporcionando un equilibrio entre facilidad de uso y potencia. (Ghimire, 2020)

Alomari en su investigación presenta un análisis exhaustivo de varios frameworks de pruebas de software, evaluando características clave como descripción, tipo, lenguaje de programación compatible, facilidad de instalación, calidad de documentación y una puntuación general. En la tabla 14 se muestra una evaluación minuciosa donde destaca a JUnit como el framework ganador, sobresaliendo en múltiples criterios. JUnit se distingue por su robustez en pruebas unitarias para Java, ofreciendo una instalación sencilla y una documentación excelente, lo que facilita su adopción y uso. Su amplia aceptación en la industria del desarrollo de software respalda su fiabilidad y preferencia. En contraste, otros frameworks presentan deficiencias en áreas como la facilidad de instalación, la compatibilidad con diferentes lenguajes o la calidad de la documentación. En resumen, JUnit emerge como la opción más completa y confiable para realizar pruebas unitarias en proyectos de desarrollo de software en Java, respaldado por su robustez, facilidad de uso y aceptación en la industria. (Alomari et al., 2020)

En la investigación de Barbotó Álvarez se hizo un análisis crítico de los frameworks Vue.js y Angular revela diferencias significativas en sus enfoques y características. Vue.js destaca por su naturaleza progresiva y adaptable, lo que lo hace altamente integrable en proyectos existentes y con una curva de aprendizaje más suave, mientras que Angular se posiciona como un framework más completo y robusto, especialmente adecuado para aplicaciones empresariales a gran escala debido a su arquitectura MVC y soporte de tipado estático en TypeScript. Aunque Vue.js ofrece simplicidad y ligereza, carece de algunas características avanzadas nativas presentes en Angular, lo que puede limitar su idoneidad para proyectos



más complejos. Por otro lado, Angular, a pesar de su mayor complejidad y curva de aprendizaje, proporciona una solución más completa y potente para aplicaciones que requieren características avanzadas y un alto grado de flexibilidad. En última instancia, la elección entre ambos frameworks dependerá de las necesidades específicas del proyecto y las habilidades del equipo de desarrollo, pero en la tabla 15 muestra que Angular emerge como la opción preferida para aplicaciones empresariales robustas y escalables. (Barbotó Álvarez, 2022)

RESULTADOS Y DISCUSIÓN

En este estudio se compararon diversos frameworks frontend con el objetivo de analizar su idoneidad para aplicaciones web con enfoque en UX y UI. Los hallazgos revelaron diferencias significativas en términos de seguridad, usabilidad, rendimiento, arquitectura, mantenimiento, modularidad, documentación, manipulación del DOM y curva de aprendizaje entre los frameworks evaluados.

Análisis de Seguridad

En términos de seguridad, se observó que la mayoría de los frameworks evaluados obtuvieron puntajes similares. En particular, en el estudio de (Mejia et al., 2015), se menciona que todos los frameworks evaluados, incluidos Spring, Struts, JSF, Angular JS, obtuvieron puntajes bajos en confiabilidad, lo que indica la necesidad de mejoras significativas en este aspecto en todos los frameworks analizados. Esto sugiere que la seguridad podría no ser un factor decisivo en la elección entre estos frameworks, ya que todos muestran áreas de mejora en este aspecto.

Análisis de Usabilidad

En términos de usabilidad, se observaron diferencias significativas entre los frameworks evaluados. Por ejemplo, en el estudio de (Espinosa Hurtado, 2021), se destaca que Django sobresale en usabilidad con un puntaje ligeramente superior al de Laravel, mientras que Angular JS mostró el menor puntaje en funcionalidad según el estudio de (Mejia et al., 2015). Esto sugiere que para proyectos donde la usabilidad es crucial, como en aplicaciones web enfocadas en la experiencia del usuario (UX/UI), la elección del framework podría depender de su capacidad para ofrecer una experiencia de usuario intuitiva y eficiente.



Análisis de Rendimiento

En cuanto al rendimiento, varios estudios destacaron diferencias significativas entre los frameworks evaluados. Por ejemplo, en el estudio de (Xing et al., 2019), se menciona que Angular 2 sobresale como el mejor framework en términos generales, mientras que en el estudio de (Kaluža & Vukelić, 2018) se observa que Vue JS y Angular lideran en diferentes aspectos de rendimiento, como la capacidad de manejar múltiples páginas y la creación de páginas simples respectivamente. Estos hallazgos sugieren que el rendimiento es un factor crucial para considerar al elegir un framework para el desarrollo web, especialmente en aplicaciones que requieren un procesamiento eficiente de datos y una respuesta rápida.

Análisis de Arquitectura y Flexibilidad

En términos de arquitectura y flexibilidad, se observaron diferencias entre los frameworks evaluados. Por ejemplo, en el estudio de (Vyas, 2022) se destaca que Angular sigue el patrón MVC, mientras que React se basa en componentes y Vue es un framework progresivo que ofrece un equilibrio entre Angular y React. Además, en el estudio de (Ghimire, 2020) se menciona que Flask destaca por su enfoque ligero y flexible, mientras que Django ofrece una solución integral para proyectos más complejos. Estos hallazgos indican que la arquitectura y la flexibilidad son aspectos importantes para considerar según las necesidades y objetivos específicos del proyecto.

Análisis de Mantenimiento y Actualización

En términos de mantenimiento y actualización, se observa que algunos frameworks sobresalen en proporcionar herramientas y estructuras que facilitan la actualización y el mantenimiento del código. Por ejemplo, en el estudio de (Chalan Jonathan, 2016) se destaca que Laravel ofrece una filosofía clara que promueve un código expresivo y elegante, lo que podría facilitar el mantenimiento a largo plazo. Además, en el estudio de (Luzuriaga Mendoza, 2020) se menciona que frameworks como Java Spring y Java Struts son flexibles y robustos, lo que podría facilitar la adaptabilidad y el mantenimiento de las aplicaciones desarrolladas con estos frameworks.

Análisis de Modularidad y Reutilización de Componentes

La modularidad y la reutilización de componentes son aspectos cruciales en el desarrollo web. Algunos frameworks se destacan en proporcionar estructuras y herramientas que favorecen la modularidad y la reutilización. Por ejemplo, en el estudio de (Ghimire, 2020) se menciona que Flask destaca por su



enfoque ligero y flexible, lo que podría facilitar la modularidad y la reutilización de componentes en proyectos pequeños. Además, en el estudio de (Barbotó Álvarez, 2022) se destaca que Angular ofrece una arquitectura MVC y soporte de tipado estático en TypeScript, lo que podría favorecer la estructuración modular de aplicaciones empresariales a gran escala.

Análisis de Documentación y Soporte Comunitario

La calidad de la documentación y el soporte comunitario son aspectos esenciales para los desarrolladores al trabajar con un framework. Algunos frameworks se destacan en proporcionar documentación detallada y un sólido soporte por parte de la comunidad. Por ejemplo, en el estudio de (Alomari et al., 2020) se destaca que JUnit ofrece una excelente documentación y una amplia aceptación en la industria del desarrollo de software, lo que podría facilitar su adopción y uso por parte de los desarrolladores. Asimismo, en el estudio de (Espinosa Hurtado, 2021) se menciona que Django ofrece una excelente documentación y soporte comunitario, lo que podría contribuir a su popularidad y uso en proyectos web.

Análisis de Rendimiento de Renderizado y Manipulación del DOM

El rendimiento de renderizado y la manipulación eficiente del DOM son aspectos críticos para la experiencia del usuario en aplicaciones web. Algunos frameworks se destacan en ofrecer un rendimiento eficiente en términos de renderizado y manipulación del DOM. Por ejemplo, en el estudio de (Ollila et al., 2022) se menciona que React.js sobresale en rendimiento de renderizado, obteniendo los mejores puntajes en métricas clave como First Contentful Paint, Speed Index y Largest Contentful Paint, lo que podría contribuir a una experiencia de usuario más fluida y rápida en aplicaciones web desarrolladas con este framework.

Análisis de Curva de Aprendizaje y Facilidad de Uso

La curva de aprendizaje y la facilidad de uso son factores importantes para considerar, especialmente para desarrolladores y equipos que buscan herramientas que les permitan comenzar a trabajar rápidamente y sin complicaciones. Algunos frameworks se destacan en ofrecer una curva de aprendizaje suave y una facilidad de uso intuitiva. Por ejemplo, en el estudio de (Mosquera Coronel, 2022) se menciona que Laravel permite una alta velocidad de codificación en la vista mediante el uso del motor de plantillas Blade, lo que podría facilitar la creación de interfaces de usuario de manera eficiente y



rápida para los desarrolladores. Asimismo, en el estudio de (Ilievska & Gramatikov, 2022) se menciona que Vue.js ofrece una curva de aprendizaje más suave y adaptable, lo que podría facilitar su adopción por parte de nuevos desarrolladores y equipos.

Ilustraciones, Tablas, Figuras

Tabla 1 - Análisis comparativo de frameworks para el desarrollo de aplicaciones web en Java

Método de Análisis	Modelo de Evaluación	Norma	Frameworks	Funcionalidad (%)	Confiabilidad (%)	Usabilidad (%)	Eficiencia (%)	Mantenimiento (%)	Portabilidad (%)
Calidad de software de código abierto (QSOS)	Modelo de Construcción de calidad individual (IQMC)	ISO 25010n	Spring	17,14% 39%	1,30% 7%	12,71% 27%	1,02% 3%	5,09% 17%	4,11% 7%
			Struts	27,99% 39%	1,30% 7%	12,19% 27%	1,02% 3%	5,07% 17%	3,90% 7%
			JSF	29,06% 39%	1,27% 7%	12,69% 27%	1,02% 3%	4,24% 17%	3,60% 7%
			Angular JS	28,36% 39%	1,27% 7%	10,23% 27%	1,02% 3%	4,08% 17%	3,45% 7%

Tabla 2 Análisis comparativo para la evaluación de frameworks usados en el desarrollo de aplicaciones web

Norma	Metodología	Herramientas	Frameworks	Usabilidad (%)	Portabilidad (%)	Rendimiento (%)	Seguridad (%)
ISO/IEC 25000	XP	Google Page Speed Insights	Laravel Django	73,20% 72,20%	66,30% 88,52%	61,20% 83,10%	88,90% 88,90%

Tabla 3 Research and Analysis of the Front-end Frameworks and Libraries in E-Business Development

Herramientas	Frameworks	Procesamiento de Datos	Ready Memory	Run Memory
			(seg)	(seg)
GitHub	React	Unidireccional	4,5 ± 0,1 (1,3)	9,7 ± 0,1 (2,6)
	Angular 1	Bidireccional	-	-
	Angular 2	Unidireccional y Bidireccional	4,8 ± 0,0 (1,4)	10,9 ± 0,1 (2,7)
	Vue JS	Unidireccional y Bidireccional	3,8 ± 0,0 (1,1)	7,5 ± 0,1 (1,9)



Tabla 4 El Frontend: Diseño web adaptativo y diseño web responsivo para el desarrollo de aplicaciones web.

Marcos de Diseño	Método de Evaluación	Población	Simplicidad y Rapidez (ptos.)	Adaptabilidad de las interfaces (ptos.)	Reducción de código (ptos.)	Originalidad de las Interfaces (ptos.)	Mejora del Rendimiento (ptos.)
Responsivo		50 docentes.	40	30	43	34	41
Adaptativo	Delphy		10	20	7	16	9

Tabla 5 Comparison of front-end frameworks for web applications development

Método de Evaluación	Herramientas	Frameworks	Multi Pages (ptos.)	Single Pages (ptos.)
Evaluación de componentes con preguntas ponderadas.	GitHub	Angular	2	7
		Vue JS	8	6
		React JS	5,5	4,5

Tabla 6 Análisis de factores que inciden en la selección de un lenguaje y framework de programación para desarrollo de software web

Framework	Rendimiento	Flexibilidad	Robustez	Curva de aprendizaje
Java Spring	Alto	Flexible	Robusto	Alta
Java Struts	Alto	Flexible	Robusto	Alta
PHP Laravel	Bueno	Menos Flexible	Limitado	Medio
PHP Symfony	Bueno	Modular	Completo	Medio
Angular	Menor	-	-	-
Vue	Más Rápido	-	-	-
React	Ligeramente más Rápido	-	-	-

Tabla 7 Modern Web Frameworks: A Comparison of Rendering Performance

Estructura	Componentes procesados	Elementos procesados	DOM Virtual
Angular	Todo	Solo fijaciones	No
React	Subarbol	Todo	Si
Vue	Solo componentes sucios	Solo fijaciones	Si
Svelte	Solo componentes	Solo fijaciones	No
Blazor	Subarbol del componente actualizado	Todo	Si

Tabla 8 Comparative Analysis on Front-End Frameworks for Web Applications

Framework	Arquitectura	Características	Popularidad	Rendimiento
Angular	MVC	Completo, soporte para múltiples funcionalidades	Alta	Real DOM, actualización completa
React	Componentes	Biblioteca flexible, renderizado eficiente	Muy Alta	Virtual DOM, actualización selectiva
Vue	Progresivo	Ligero, fácil de aprender, buen rendimiento	En Crecimiento	Virtual DOM, actualización selectiva



Tabla 9 Análisis comparativo de los frameworks Laravel y codeigiter para la implementación del sistema de gestión de recursos de méritos y oposición en la universidad nacional de Chimborazo

Framework	Definición	Características	Arquitectura	Productividad (%)
Laravel	Framework Open-Source para desarrollar en PHP, enfocado en elegancia y expresividad	Filosofía clara, Eloquent ORM, modular y extensible, HTTP routing eficiente, HTTP Middleware	MVC	100%
CodeIgniter	Framework PHP para desarrollo de aplicaciones web, con enfoque en simplicidad y rendimiento	Versatilidad, facilidad de instalación, flexibilidad, ligereza, acoplamiento y singularidad de componentes	Modelo-Vista-Controlador	68,86%

Tabla 10 Estudio Comparativo de frameworks (java) zk y (php) laravel para desarrollo de aplicaciones web

Criterios	ZK	LARAVEL
Desarrollo Agil sin el uso de javascript	Alto	Poco
Codifica en la vista	Mediante el id del componente	Desde el motor de pantilla
Implementacion de un controlador	Extendemos zk's composer	Es un pojo
Acceso a la informacion de la ui	Directo	No
Acceso a la ifnормacion desde el backend	Manipulacion directa	Si
Rendimiento	Alto	Normal
Monolitico	Si	Si
Microservicios	No	Si
Generador de XML	Si	Si
IDE para desarrollo	Netbeans	PHPStorm
MVC	Si	Si
Ajax y Json	Si	Si
WebSocket	Si	Si
Seguridad	Si	Si
Motor de Plantilla	Zuml	Blade
Creacion de PWa	Si	Si
Experiencia de usuario	Buena	Ninguan



Tabla 11 Analysis and comparative evaluation of front-end technologies for web application development

Criterio	Angular	React	Vue.js
Performance (Lighthouse)	86	92	91
DOM Manipulación	4	3	2
Memory Allocation	3	3	3
Build Size	4	4	4
Startup Time	3	5	3
Rich Package Ecosystem	3	4	3
Flexibility	3	4	4
Reusability	4	4	4
Documentación	3	4	4
Learning Curve	3	3	4
Puntuación promedio	3.33	3.79	3.44

Tabla 12 Comparative analysis of MEAN stack and MERN stack

Característica	MEAN (Angular)	MERN (React)
Rendimiento	Bueno	Mejor
Arquitectura	MVC impuesto por el framework	Vista solamente, mayor flexibilidad
Librerías de terceros	Pocas	Muchas requeridas
Tendencias	Más antiguo, perdiendo popularidad	Más moderno, ganando popularidad
Data Binding	Bidireccional	Unidireccional
Plantillas	HTML	JSX
Modelo de datos	Fuerte	Mediano
Renderizado	Del lado del cliente	Del lado del servidor

Tabla 13 Comparative study on Python web frameworks: Flask and Django

Características	Flask	Django
Tipo de Framework	Microframework	Fullstack
Diseño y Arquitectura	Sigue el patrón MVC de forma flexible	Sigue estrictamente el patrón MVC
Ruteo y Manejo de Solicitudes	Decoradores para definir rutas, manejo de solicitudes automático	Enrutamiento configurable, solicitudes requieren explícitamente el objeto request
Blueprints (Módulos)	Soporte para blueprints para organizar la aplicación	No tiene soporte para blueprints, se utiliza la estructura de apps
Base de Datos y ORM	Base de datos y ORM de terceros (SQLAlchemy, Peewee, etc.)	ORM integrado y soporte para múltiples bases de datos
Seguridad	Extensiones de terceros (Flask-WTF, Flask-Login, etc.)	Mecanismos de seguridad integrados (protección CSRF, XSS, inyección SQL,



		etc.)
Configuración	Configuración flexible a través de objetos de configuración	Configuración centralizada en el archivo settings.py
Despliegue	Despliegue sencillo (Heroku, etc.)	Despliegue configurable pero más complejo
Documentación	Documentación extensa y ejemplos claros	Documentación exhaustiva y tutoriales detallados
Curva de Aprendizaje	Curva de aprendizaje más rápida	Curva de aprendizaje más pronunciada
Comunidad y Soporte	Comunidad activa pero más pequeña que Django	Comunidad muy grande y amplio soporte

Tabla 14 A User Interface (UI) and User eXperience (UX) evaluation framework for cyberlearning environments in computer science and software engineering education

Framework	Descripción	Tipo	Lenguaje	Instalación	Documentación	Puntuación
JUnit	Framework de pruebas unitarias	Pruebas unitarias	Java	Sencilla	Excelente	5
JDepend	Herramienta de métricas de código	Calidad de código	Java	Moderada	Buena	4
EMMA	Herramienta de cobertura de código	Cobertura de código	Java	Sencilla	Buena	4
CPPU	Framework de pruebas unitarias	Pruebas unitarias	C++	Compleja	Regular	3
Cobertura	Herramienta de cobertura de código	Cobertura de código	Java	Moderada	Regular	3

Tabla 15 Análisis comparativo de los frameworks móviles entre Vue y Angular año 2022

Punto de Discusión	Angular	Vue
Complejidad	Alta	Media
Integración Fácil	Fácil de integrar, pero con componentes de terceros	fácil de integrar con otras bibliotecas front-end
Enlace de datos	Utiliza el enlace bidireccional entre ámbitos.	Utiliza el flujo de datos en el camino entre los componentes
Flexibilidad	Si	Si
Curva de aprendizaje	Es más pronunciada debido a su extensa documentación de API.	No es tan pronunciada, permitiendo a los usuarios desarrollar aplicaciones en menos tiempo
Mecanografiado	Si	No utiliza



Figura 1 Análisis de bases de datos científicas



CONCLUSIONES

Tras analizar detenidamente los datos obtenidos y considerando la argumentación teórica respaldada por la evidencia fáctica, se sostiene que la elección del mejor framework depende en gran medida de las necesidades y objetivos específicos de cada proyecto. Aunque se han identificado puntos fuertes y áreas de mejora en diversos frameworks frontend, no se puede afirmar categóricamente que uno sea superior al resto en todos los aspectos evaluados.

En seguridad, si bien no hubo un claro ganador, se observó que todos los frameworks presentaron medidas de seguridad mejorables, lo que sugiere la importancia de implementar prácticas adicionales de seguridad en el desarrollo de aplicaciones web.

En usabilidad, se destaca la facilidad de uso y adaptabilidad de Django, así como la curva de aprendizaje suave y la experiencia intuitiva de Vue JS, aspectos que los posicionan como opciones atractivas para desarrolladores y equipos que valoran la usabilidad y la experiencia del usuario.

En rendimiento, Angular 2 mostró eficiencia en términos de procesamiento de datos y manipulación del DOM, lo que lo convierte en una opción a considerar para proyectos que requieren un alto rendimiento.



En arquitectura y flexibilidad, Angular y React ofrecieron soluciones robustas y completas, especialmente adecuadas para proyectos a gran escala con necesidades avanzadas de arquitectura y flexibilidad.

En mantenimiento y actualización, frameworks como Laravel y Django se destacaron por su filosofía clara, facilidad de mantenimiento y documentación detallada, lo que facilita la actualización y el mantenimiento a largo plazo del código.

Por último, cabe destacar que la elección del framework más adecuado debe basarse en una evaluación cuidadosa de sus características en relación con los requisitos del proyecto y las habilidades del equipo de desarrollo. Se plantea como tarea pendiente la investigación continua en este campo para abordar interrogantes no resueltos y colaborar en la ampliación del conocimiento sobre el uso de frameworks frontend en el desarrollo de aplicaciones web.

REFERENCIAS BIBLIOGRAFICAS

- Aggarwal, S., & Verma, J. (2018). Comparative analysis of MEAN stack and MERN stack. In *International Journal of Recent Research Aspects* (Vol. 5).
- Alomari, H. W., Ramasamy, V., Kiper, J. D., & Potvin, G. (2020). A User Interface (UI) and User eXperience (UX) evaluation framework for cyberlearning environments in computer science and software engineering education. *Heliyon*, 6(5). <https://doi.org/10.1016/j.heliyon.2020.e03917>
- Barbotó Álvarez, T. J. (2022). Análisis comparativo de los frameworks móviles entre Vue y Angular año 2022. *Universidad Técnica de Babahoyo*. <http://dspace.utb.edu.ec/handle/49000/11699>
- Chalan Jonathan. (2016). Análisis comparativo de los frameworks Laravel y codeigiter para la implementación del sistema de gestión de recursos de méritos y oposición en la universidad nacional de Chimborazo. *Universidad Nacional de Chimborazo*.
- Chavez Calderón, J. X., Zambrano Romero, W. D., Cedeño Palma, E. A., Zambrano Zambrano, D. M., & Cotera Ramírez, G. A. (2022). El Frontend. Diseño web adaptativo y diseño web responsive para el desarrollo de aplicaciones web. *Informática y Sistemas: Revista de Tecnologías de La Informática y Las Comunicaciones*, 6(1), 79–95. <https://doi.org/10.33936/isrtic.v6i1.4625>
- Espinosa Hurtado, R. (2021). Análisis comparativo para la evaluación de frameworks usados en el desarrollo de aplicaciones web. *CEDAMAZ*, 11(2), 133–141.



<https://doi.org/10.54753/cedamaz.v11i2.1182>

Ghimire, D. (2020). Comparative study on Python web frameworks: Flask and Django.

<https://www.theses.fi/handle/10024/339796>

Ilievska, F., & Gramatikov, S. (2022). Analysis and comparative evaluation of front-end technologies for web application development. <http://hdl.handle.net/20.500.12188/25342>

Kaluža, M., & Vukelić, B. (2018). Comparison of front-end frameworks for web applications development. *Zbornik Veleučilišta u Rijeci*, 6(1), 261–282. <https://doi.org/10.31784/zvr.6.1.19>

Luzuriaga Mendoza, A. M. (2020). Análisis de factores que inciden en la selección de un lenguaje y framework de programación para desarrollo de software. *Universidad Tecnica de Machala*, 1–52.

Mejia, H., Tuesta Monteza, V., & Sánchez Acosta, C. (2015). Análisis comparativo de frameworks para el desarrollo de aplicaciones web en java. <https://www.researchgate.net/publication/343614953>

Mosquera Coronel, K. J. (2022). Estudio comparativo de frameworks de java ZK y (PHP) Laravel de aplicaciones web y móviles. <http://dspace.utb.edu.ec/handle/49000/13056>

Ollila, R., Mäkitalo, N., & Mikkonen, T. (2022). Modern Web Frameworks: A Comparison of Rendering Performance. *Journal of Web Engineering*, 21(3), 789–813.

<https://doi.org/10.13052/jwe1540-9589.21311>

Vyas, R. (2022). Comparative Analysis on Front-End Frameworks for Web Applications. *International Journal for Research in Applied Science and Engineering Technology*, 10(7), 298–307. <https://doi.org/10.22214/ijraset.2022.45260>

Xing, Y. K., Huang, J. P., & Lai, Y. Y. (2019). Research and Analysis of the Front-end Frameworks and Libraries in E-Business Development. *ACM International Conference Proceeding Series*, 68–72. <https://doi.org/10.1145/3313991.3314021>

