

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

Comparative analysis for the evaluation of frameworks used in web development applications

Raquel Espinosa-Hurtado^{1,*}

¹ Carrera de Ingeniería en Sistemas/Computación, Universidad Nacional de Loja, Loja, Ecuador

* Autor para correspondencia: raquel.espinosa@unl.edu.ec

Fecha de recepción del manuscrito: 17/11/2021

Fecha de aceptación del manuscrito: 11/12/2021

Fecha de publicación: 24/12/2021

Resumen—El desarrollo de aplicaciones web ha tenido un crecimiento exponencial a lo largo de los últimos años, esto se debe al hecho de que cada día son más las exigencias requeridas para la web que proporcionen a los usuarios mejores experiencias al momento de hacer uso de ellas. Por tal razón, surge la necesidad de utilizar herramientas que permitan implementar estas aplicaciones de forma rápida y eficiente, además de que puedan adaptarse a las necesidades de los desarrolladores. El objetivo del presente trabajo es realizar un análisis comparativo para la evaluación de frameworks usados en el desarrollo de aplicaciones web; para esto, inicialmente se determinaron los frameworks Laravel y Django. Para conocer cuál de estos dos entornos era el mejor, se determinaron los criterios que permitieron realizar la evaluación de los mismos; estos criterios fueron establecidos en base al modelo de evaluación de la norma ISO/IEC 25000. Para realizar el análisis de los frameworks se generó un entorno de experimentación que consistió en el desarrollo de una aplicación de prueba, con la cual se pudo efectuar la evaluación de los mismos. La aplicación fue desarrollada haciendo uso de la metodología XP, ya que esta es considerada como una metodología ágil que se adapta de forma adecuada a las necesidades del proyecto. Finalmente, se realizó el análisis comparativo, mediante el cual se pudo determinar las fortalezas y debilidades que presentan los frameworks, logrando determinar que Django es el entorno más adecuado para el desarrollo de aplicaciones web.

Palabras clave—Aplicaciones web, Frameworks, ISO/IEC 25000, Modelo de evaluación

Abstract—The development of web applications has had exponential growth over the last few years, this is due to the fact that every day there are more and more demands required for the web to provide users with better experiences when using them. For this reason, there is a need to use tools that allow the implementation of these applications quickly and efficiently, and that can be adapted to the needs of developers. The objective of this work is to perform comparative analysis for the evaluation of frameworks used in the development of web applications; for this, the Laravel and Django frameworks were initially determined. In order to know which of these two frameworks was the best, the criteria that allowed the evaluation of the same were determined; these criteria were established based on the evaluation model of the ISO/IEC 25000 standard. In order to perform the analysis of the frameworks, an experimental environment was generated, which consisted of the development of a test application, with which the evaluation of the frameworks could be carried out. The application was developed using the XP methodology since this is considered an agile methodology that adapts adequately to the needs of the project. Finally, the comparative analysis was performed, through which it was possible to determine the strengths and weaknesses of the frameworks, being able to determine that Django is the most suitable environment for the development of web applications.

Keywords—Web applications, Frameworks, ISO/IEC 25000, Evaluation model

INTRODUCCIÓN

A lo largo de los años se ha podido evidenciar que el avance de la tecnología ha tenido un crecimiento exponencial en todos sus campos de aplicación, por lo tanto, el desarrollo web no es la excepción, ya que cada vez son más las exigencias requeridas, necesitando de nuevas y útiles características para los usuarios de la web. El avance de la tec-

nología en estos tiempos ha sido crucial, ya que ha permitido que los individuos puedan interactuar de mejor forma y desenvolverse en sus tareas sin mayores inconvenientes. En este contexto, el desarrollo de aplicaciones web es vital, pues facilita en gran medida la vida de las personas, por lo tanto, es necesario que se sigan desarrollando nuevas herramientas que brinden a los usuarios experiencias significativas que sean intuitivas y fáciles de usar. Sin embargo, todas

estas nuevas exigencias y requerimientos para la web, representan nuevos retos para quienes llevan a cabo la tarea del desarrollo, teniendo en cuenta que, al satisfacer las exigencias de los usuarios, también se requiere de estructuras de desarrollo que proporcionen el soporte necesario para crear aplicaciones que complazcan a los internautas, y que además de potentes sean fáciles de usar para los programadores.

Es allí donde nace la necesidad del uso de los frameworks de desarrollo web, ya que estos incorporan funcionalidades ya desarrolladas y probadas, implementadas en un determinado lenguaje de programación que permite a los desarrolladores crear aplicaciones web de forma fácil y rápida, así como también mantener el código generado y realizar configuraciones sobre el mismo, proporcionando patrones de diseño que permiten crear un código mucho más robusto, ordenado y libre de redundancias, debido a que al tener muchas funcionalidades y especificaciones, es posible que los programadores adopten malas prácticas de programación, mezclando el código generado y redundando en el desarrollo (Molina Ríos, 2016). Los frameworks son diseñados precisamente para evitar estos problemas y lograr que se generen aplicaciones de calidad. Sin embargo, surge la interrogante sobre ¿qué framework es mejor para mi aplicación?

La presente investigación nace del análisis de la problemática encontrada en el ámbito del desarrollo de aplicaciones web, con base en los inconvenientes mencionados anteriormente, se ha considerado pertinente realizar un análisis comparativo para la evaluación de frameworks usados en el desarrollo de dichas aplicaciones. Para el desarrollo del proyecto se determinaron los frameworks de desarrollo de aplicaciones web más utilizados en el año 2020, con lo cual se obtuvo una lista de frameworks, determinando que los entornos Laravel y Django fueron los más usados en dicho año, con los cuales se llevó a cabo la evaluación. Luego se seleccionó el modelo perteneciente a la Norma ISO/IEC 25000 para el establecimiento de los criterios que sirvieron para la evaluación de los frameworks anteriormente mencionados. Posterior a ello, se generó un escenario de experimentación, en el cual se realizó la implementación de una aplicación que sirvió para la evaluación de los frameworks en base a los criterios establecidos por medio del modelo seleccionado; y, finalmente se realizó el análisis comparativo en base a los resultados obtenidos en la evaluación y se establecieron las fortalezas y debilidades de cada uno de ellos.

A continuación, se describe de manera general las secciones que contiene el presente trabajo, en el cual consta todo el proceso realizado para dar cumplimiento al mismo. En la sección de Fundamentos teóricos se muestra toda la información necesaria que sirvió como base fundamental para adquirir los conocimientos requeridos para realizar la investigación. La sección de Materiales y métodos presenta los recursos que fueron utilizados y los participantes que intervinieron en el desarrollo del proyecto. Posterior a ello, se encuentra la sección de Resultados, en la cual se presenta la evidencia obtenida a lo largo del desarrollo del trabajo. En la sección de Discusión se analiza los resultados obtenidos, y de igual forma, se describe cómo se cumple con el trabajo realizado. Finalmente se encuentran las conclusiones emitidas luego de haber culminado el desarrollo del presente trabajo.

MATERIALES Y MÉTODOS

Para la presente investigación fue necesaria la utilización de ciertos recursos, los cuales se detallan a continuación:

Recursos científicos

Método analítico

Se lo empleó porque permite analizar de manera minuciosa hechos en particular, en este caso se realizó el análisis tomando en consideración las características de la métrica de calidad según el estándar internacional ISO/IEC 25000 con el objetivo de poder determinar cuál es el mejor framework que se puede utilizar para el desarrollo de aplicaciones web.

Método descriptivo

Se lo empleó para realizar la descripción de las características y prestaciones que cada uno de los frameworks posee, tomando en cuenta los criterios establecidos en la norma ISO/IEC 25000, para realizar el análisis de los resultados, y de esta manera poder emitir criterios que sean de ayuda y enriquezcan el conocimiento y ayuden a los desarrolladores a poder seleccionar la mejor herramienta que pueda servir para llevar a cabo el desarrollo de aplicaciones web.

Revisión bibliográfica

Gracias a la utilización de esta técnica se sustentó la base teórica de la investigación, mediante consultas a fuentes bibliográficas confiables como: libros, artículos científicos, revistas indexadas, bases de datos científicas, entre otros.

Reuniones

Esta técnica fue necesaria para llevar a cabo el proceso de experimentación, con lo cual se pudo dar a conocer cada una de las indicaciones y especificaciones del experimento desarrollado a los alumnos seleccionados para este proceso. De igual manera, fue útil para realizar el seguimiento de las tareas y recolección de la información.

Metodología XP

Esta metodología fue de vital importancia en el desarrollo de la aplicación de prueba realizada en la experimentación de la presente investigación, debido a que tal y como se menciona en (Arcos Chalán Chicaiza Inguillay, 2016) y (Builes, 2019), la metodología de desarrollo XP (Programación extrema) es un proceso ágil de desarrollo de software, que se enfoca en realizar buenas prácticas de codificación, una clara comunicación y está orientada al trabajo en equipo.

Recursos éticos

Consentimiento informado

Este recurso fue muy importante para realizar la experimentación, a través del cual se pudo garantizar que los alumnos seleccionados para llevar a cabo dicha tarea expresaran de forma voluntaria su intención de poder participar en la presente investigación, para ello, previamente se les dio a conocer las especificaciones del experimento a realizar.

Tabla 1: Frameworks más utilizados en el año 2020

N°	Nombre del Framework
1	Laravel
2	Django
3	VueJS
4	React
5	Ruby on Rails
6	Node.js
7	Angular
8	Express
9	jQuery
10	Electron
11	ExtJS
12	Jenkins
13	EmberJS
14	Knockout
15	A-Frame
16	Mithril
17	Mybatis
18	Codeigniter
19	BackboneJS
20	CakePHP
21	Spring
22	SSM framework
23	VAADIN Framework
24	Aurelia
25	ZK

de decir que la interfaz gráfica es sólo una página web en el navegador (Arcos Chalán Chicaiza Inguillay, 2016). En la Figura 1 se puede evidenciar la arquitectura MVC en la cual está basada el framework Laravel.

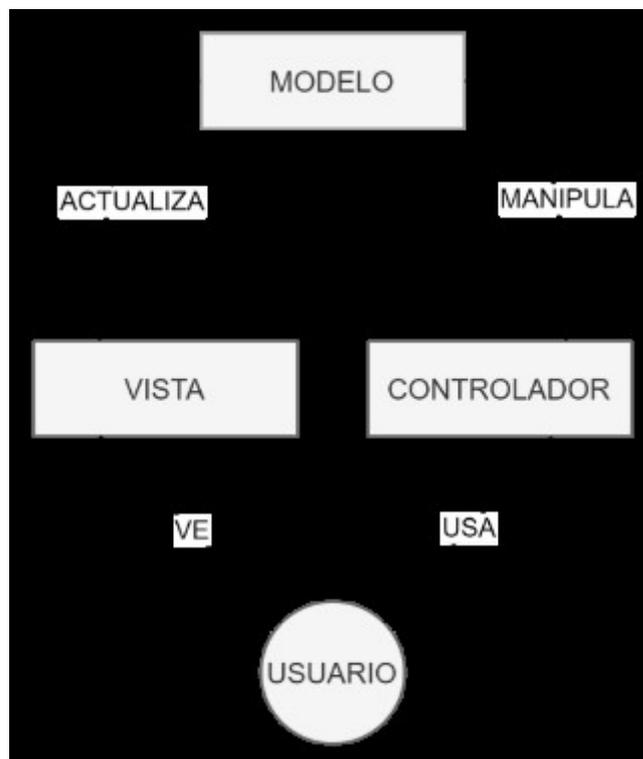


Fig. 1: Arquitectura de Laravel (Ovando Ortega, 2019).

Elección de los frameworks

De acuerdo a una RSL (Revisión Sistemática de Literatura) realizada en el año 2020 sobre los frameworks de desarrollo de aplicaciones web más utilizados a lo largo de dicho año, se obtuvieron los resultados que se presentan en la Tabla 1.

Para el presente estudio se seleccionaron los frameworks Laravel y Django para llevar a cabo la comparación de los mismo, siendo que estos fueron los dos más utilizados durante dicho año para el desarrollo de aplicaciones web.

Framework Laravel

Laravel es un framework de código abierto basado en el lenguaje PHP, este posee una filosofía muy clara, que está enfocada para que el código desarrollado sea lo más expresivo y elegante posible, para desarrollar aplicaciones y servicios web. Laravel propone una forma de desarrollar aplicaciones web de un modo mucho más ágil, simplificando el trabajo con tareas comunes como la autenticación, el enrutamiento, gestión sesiones, etc. (Cabrera León, 2019).

Arquitectura de Laravel

Laravel sigue el patrón de arquitectura MVC tradicional, mismo que especifica una separación entre la lógica de negocios, de la lógica de entradas y presentación asociada con una interfaz gráfica de usuario (GUI). De esta manera, en las aplicaciones web creadas por Laravel, la lógica de negocios básicamente consiste en los modelos de datos, como por ejemplo: usuarios, publicaciones en blogs, e incluso se pue-

Características

Modular y extensible: Laravel es un framework que permite agregar todo lo que se requiera, ya que cuenta con un directorio llamado Packalyst, el cual contiene más de 5,500 paquetes donde se puede encontrar lo que se necesite ya implementado.

Tareas automatizadas: con el API Elixir de Laravel se pueden crear tareas de Gulp con las que es posible definir el uso pre-procesadores para comprimir CSS (Hojas de Estilo en Cascada, del inglés Cascading Style Sheets) y JavaScript.

HTTP routing: Laravel cuenta con un sistema de enrutamiento rápido y eficiente, similar al que se usa en Ruby on Rails. Este sistema permite relacionar las partes de la aplicación con las rutas que ingresa el usuario en el navegador.

Caché: Posee un sistema robusto de caché configurable.

Integra un sistema ORM de mapeado de datos relacional llamado Eloquent, y a su vez, también permite la construcción de consultas directas a base de datos a través de su Query Builder.

Autenticación: Es realizada mediante un login y password que permiten restringir el acceso y el tipo de permisos de los diferentes usuarios (Ovando Ortega, 2019).

Framework Django

Django es un framework de desarrollo web de alto nivel y de código abierto, que está escrito en el lenguaje de programación Python. Este framework se caracteriza porque permite construir aplicaciones de forma rápida, segura y con menos código (en comparación con otros frameworks). Además, cuenta con una comunidad de colaboradores muy grande a nivel mundial que se encarga de realizar las debidas actualizaciones tanto del framework como de su documentación de forma diaria (Gómez García, 2018).

Arquitectura de Django

Django se basa en una arquitectura de tipo MVC (Model – View – Controller); sin embargo, en Django esta tendencia se aplica resumiendo una nueva estructura denominada MVT (Model – View – Template). En la capa “Model” se encuentra toda la información que sirve para estructurar y manipular los datos de la aplicación web. Esta capa también es conocida como la capa de acceso a la base de datos. Por otro lado, la capa “Template/Plantilla” es la capa de presentación, en ella se encuentra todas las decisiones que están relacionadas con la presentación. Finalmente, en la capa “View” se encuentra toda la lógica que accede al modelo y la delega a la plantilla apropiada para presentarle al usuario la respuesta a su petición. En la Figura 2 se muestra la arquitectura del framework Django (Caldera Vergara, 2017).

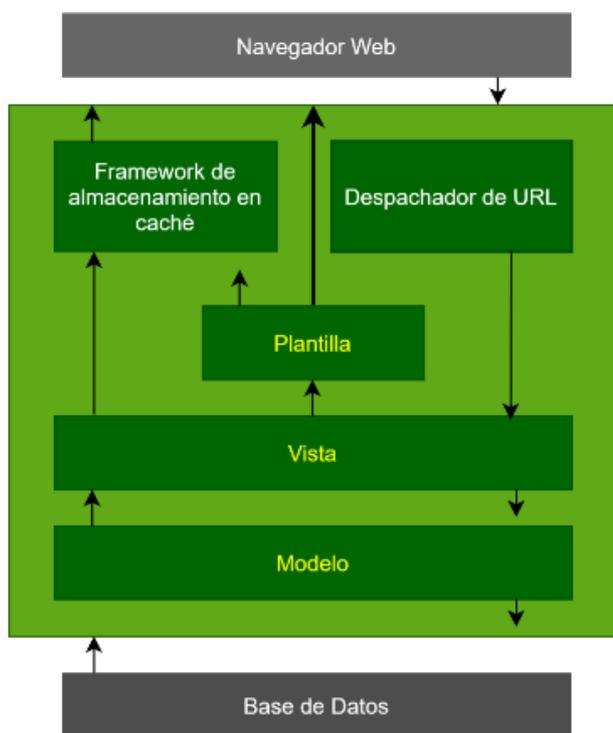


Fig. 2: Arquitectura de Django (Caldera Vergara, 2017).

Características

Complejidad: Django es un framework muy completo, provee diseños consistentes para que los desarrolladores puedan hacer lo que requieren en sus proyectos.

Versátil: permite construir casi cualquier tipo de sitio web, tales como los sistemas manejadores de contenidos y wikis, hasta el desarrollo de redes sociales y sitios nuevos.

Portabilidad: debido a que está escrito en Python, este framework puede ejecutarse en muchas plataformas, como por ejemplo: Windows, Linux y Mac.

Mantenibilidad: Django es un framework que está escrito usando principios y patrones de diseño para fomentar la creación de código que pueda ser mantenible y reutilizable. Utiliza el principio Don't Repeat Yourself (No te repitas) para evitar que exista una duplicación innecesaria en el código.

Seguridad: brinda mucha seguridad en el desarrollo. Proporciona una forma segura de administrar cuentas de usuarios y contraseñas (Gómez García, 2018).

Calidad de Software

Según García León (2016): “La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad”, por tal razón se puede decir que la calidad trata sobre los diferentes conceptos, métodos, procedimientos, estándares y técnicas que se requiere para la producción de productos y procesos de software que sean considerados de alta calidad.

La norma ISO/IEC 25000

De acuerdo a (Standardization, 2021), la norma ISO/IEC 25000, también conocida como SQuaRE, por sus siglas en inglés (System and Software Quality Requirements and Evaluation), “es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software.” La Norma ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores a ella, principalmente de las normas ISO/IEC 9126, la cual describe las particularidades de un modelo de calidad del producto software, y también de la norma ISO/IEC 14598, que abordaba el proceso para la evaluación de productos software. La norma ISO/IEC 25000 se encuentra compuesta por cinco divisiones, las cuales son:

- ISO/IEC 2500n – División de Gestión de Calidad.
- ISO/IEC 2501n – División de Modelo de Calidad.
- ISO/IEC 2502n – División de Medición de Calidad
- ISO/IEC 2503n – División de Requisitos de Calidad.
- ISO/IEC 2504n – División de Evaluación de Calidad.

Criterios de Calidad de Software

En la Tabla 2 se presenta los aspectos que fueron considerados para llevar a cabo la tarea de especificación de los criterios, mismos que fueron seleccionados en base a las características del modelo de calidad de la ISO/IEC 25010.

Tabla 2: Criterios seleccionados

Característica	Sub Característica	Criterio	Significado del Criterio
Rendimiento	Tiempo de ejecución de operaciones CRUD.	Tiempo de ejecución de operaciones Create (Crear).	Tiempo que el framework emplea en la ejecución de la operación CRUD – CREATE.
		Tiempo de ejecución de operaciones Read (Leer).	Tiempo que el framework emplea en la ejecución de la operación CRUD – READ.
		Tiempo de ejecución de operaciones Update (Actualizar).	Tiempo que el framework emplea en la ejecución de la operación CRUD – UPDATE.
		Tiempo de ejecución de operaciones Delete (Eliminar).	Tiempo que el framework emplea en la ejecución de la operación CRUD – DELETE.
Utilización de recursos.		Líneas de código.	Cantidad de líneas de código existentes por cada función implementada.
		Utilización de CPU.	Porcentaje de uso del procesador por el framework.
		Utilización de la memoria RAM.	Cantidad de memoria que es utilizado por el framework.
Usabilidad	Entendibilidad.	Efectividad de la documentación del usuario o ayuda del sistema.	Capacidad de encontrar una documentación clara que sirva para la investigación, es decir, la información disponible acerca del framework web.
	Protección contra errores del usuario.	Verificación de entradas válidas.	Cantidad de elementos de entrada que son validados.
	Curva de aprendizaje.	Tiempo requerido para aprender a utilizar el framework.	Tiempo necesario para obtener un nivel de conocimiento aceptable del uso del framework web.
Portabilidad	Capacidad de ser Instalado.	Eficiencia en el tiempo de instalación.	Tiempo que se requiere para llevar a cabo el proceso de instalación del framework.
		Facilidad de instalación.	Facilidad que posee el framework para su instalación.
Seguridad	Confidencialidad.	Capacidad de control de acceso.	Qué tan controlables son los accesos al sistema.
		Encriptación de datos.	Capacidad del framework para realizar encriptación de datos.
	Autenticidad.	Métodos de autenticación.	Qué tan bien el framework autentica la identidad de un sujeto o recurso.

RESULTADOS

En esta sección se detallan los resultados obtenidos a lo largo de la presente investigación.

Generación del entorno de experimentación para la comparación de los Frameworks

Para el desarrollo del experimento, se pusieron a prueba los frameworks seleccionados, para lo cual se implementó una aplicación web, la cual sirvió como escenario real y permitió evaluar las características establecidas en la Tabla 2, de tal forma que se pudo comparar estos marcos de desarrollo, logrando determinar cuál es el mejor de ellos para determinada aplicación web.

Selección del contexto

El objeto experimental definido en esta sección consiste en la implementación de una aplicación web sobre un catálogo de productos en línea, para ello, la aplicación fue implemen-

tada en los dos frameworks seleccionados. Dicha aplicación fue realizada haciendo uso de la metodología de desarrollo XP “eXtreme Programming” o “Programación Extrema”.

Formulación de Hipótesis

A continuación, se presenta la hipótesis planteada en la presente investigación:

¿El framework Django brinda las mejores prestaciones para el desarrollo de aplicaciones web en comparación con el framework Laravel?

Selección de sujetos

Para llevar a cabo la experimentación, se contó con la colaboración de los estudiantes del noveno ciclo de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, los cuales desarrollaron la aplicación, y través de este proceso se logró obtener resultados concretos para evaluar qué framework es mejor para el desarrollo de aplicaciones web. Cabe recalcar que, en el momento de la ejecución del ex-

perimento, estos alumnos estaban cursando la asignatura de Ingeniería del Software II y tenían conocimientos sólidos sobre lenguajes de programación como PHP y Python, lo cual era necesario para trabajar con los entornos de desarrollo Laravel y Django.

Aplicación de prueba

El desarrollo de la aplicación de prueba es considerado como una parte fundamental para la evaluación de los frameworks en base a los criterios seleccionados de la normativa ISO/IEC 25000. La aplicación permitió comprobar algunas prestaciones que ofrece cada uno de los marcos de desarrollo web seleccionados. A continuación, en la Figura 3 se presenta una vista panorámica de cómo quedó armado el entorno experimental definido en el presente trabajo, partiendo por la selección de los estudiantes que realizaron el desarrollo de la aplicación haciendo uso de los dos frameworks. El proceso de instalación de dichos frameworks fue realizado en los sistemas operativos (SO) Linux y Windows, mientras que para el desarrollo únicamente se realizó en el SO Windows. Se trabajó en un servidor local y se utilizó la base de datos de MySQL. Finalmente, para la realización de las pruebas, se utilizó la herramienta CLOC para la medición de las líneas de código y las herramientas de desarrollador proporcionadas por el navegador de Google Chrome para evaluar el rendimiento de la aplicación desarrollada a través de los dos frameworks.

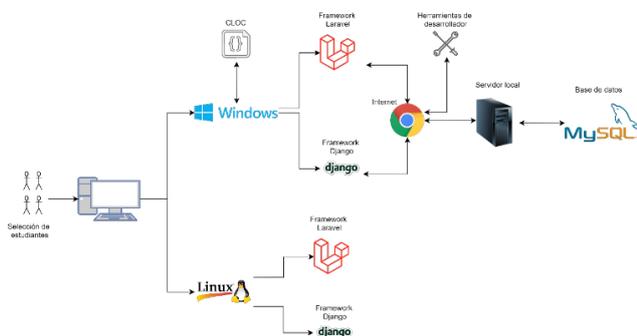


Fig. 3: Escenario de experimentación.

Evaluación de los Frameworks

En esta fase se realizó la evaluación de los frameworks en base a los criterios seleccionados de la Norma ISO/IEC 25000, los resultados obtenidos se presentan a continuación:

Análisis de los resultados obtenidos

Rendimiento

En lo referente al rendimiento, en la Tabla 3 se puede observar que Django obtuvo una mejor calificación frente a Laravel, debido a que cumple con mayor satisfacción cada uno de los criterios establecidos, obteniendo un 83.1 % del total, frente al 61.2% obtenido por Laravel. Django fue el framework con mayor velocidad en la ejecución de aplicaciones en la creación, lectura, modificación y eliminación, además, ya que el indicador fue establecido en unidades de tiempo,

esto significa que los rangos de valores son inversos, por lo tanto, a menor tiempo, mayor rendimiento. Además, la cantidad de líneas de código requeridas para una determinada operación fue menor que las que se requirió en Laravel, esto se debe a que Django cuenta con la parte de administración, en la cual ya se implementan algunas funcionalidades para que el desarrollo sea mucho más eficiente, y al estar basado en el lenguaje de programación Python, su sintaxis es mucho más fácil y el código requerido es más reducido; finalmente, se logró determinar que los recursos utilizados por Django tanto en la utilización de CPU como en memoria RAM son menores que los que se requiere por Laravel para la ejecución de la aplicación de prueba que fue implementada.

Usabilidad

En lo referente a la característica de usabilidad, Laravel obtuvo una mejor calificación que Django, aunque los resultados fueron bastante aproximados (ver Tabla 4), aun así, Laravel se llevó el primer lugar en este parámetro, obteniendo un 73.2% del total frente a Django, el cual obtuvo un 72.7%. En el desarrollo de la aplicación se pudo observar que la documentación proporcionada por los dos frameworks es muy buena, amplia y eficiente, ya que proveen guías de instalación y en sus páginas oficiales cuentan con la credibilidad necesaria, pero sobre todo, se debe recalcar que esta información se encuentra muy bien estructurada, tanto que ahorra el tiempo requerido al momento de realizar consultas, y además, sus comunidades de desarrollo mantienen la documentación oportuna ante las necesidades y exigencias de los desarrolladores. Por otro lado, en lo referente a la validación de entradas se pudo verificar que los dos frameworks permiten implementar y validar de forma adecuada las entradas de datos. Finalmente, se determinó que el desarrollo de aplicaciones es muy fácil con cualquiera de los dos frameworks, sin embargo se debe recalcar que el tiempo en el cual se logre llevar a cabo un determinado proyecto dependerá mucho de la experiencia de los desarrolladores, de tal forma que en el presente trabajo el desarrollo del catálogo de productos fue más rápido con el framework Laravel, esto se debió a que los desarrolladores tenían más experiencia en la programación y conocían de mejor manera la herramienta.

Portabilidad

Con respecto a la característica de portabilidad, como se muestra en la Tabla 5, el framework Django obtuvo una mejor calificación que Laravel, obteniendo un 88.52 % del total, frente a Laravel, el cual obtuvo un 66.3%. De esta forma se pudo determinar que el tiempo requerido por Laravel en la instalación del mismo fue menor que en Django. Así mismo, se pudo evidenciar que ambos frameworks son muy fáciles de instalar en Windows, sin embargo, para la instalación en Linux, fue Django el que obtuvo una mejor puntuación.

Seguridad

Como se puede observar en la Tabla 6, en la característica de seguridad los dos frameworks obtuvieron un valor del 88.9%, debido a que ambos cumplieron con los criterios es-

Tabla 3: Datos obtenidos de la característica Rendimiento

Sub Característica	Criterio	Valor Obtenido		Ponderación (/10)	
		Laravel	Django	Laravel	Django
Tiempo de ejecución de operaciones CRUD	Create	657 ms	227 ms	6.72	8.87
	Read	628 ms	164 ms	6.86	9.18
	Update	752 ms	142 ms	6.24	9.29
	Delete	796 ms	199 ms	6.02	9.01
Utilización de recursos	Líneas de código	2666	751	4.67	8.50
	Utilización de la CPU	3.8%	2.7%	9.24	9.46
	Utilización de RAM	3545	3153	3.08	3.84
Valor parcial	Laravel	6.12	Valor final	Laravel	2.14
	Django	8.31		Django	2.91

Tabla 4: Datos obtenidos de la característica Usabilidad

Sub Característica	Criterio	Valor Obtenido		Ponderación (/10)	
		Laravel	Django	Laravel	Django
Entendibilidad	Efectividad de la documentación del usuario o ayuda del sistema	7	7	10	10
Protección contra errores del usuario	Verificación de entradas válidas	3	3	3.33	3.33
Curva de aprendizaje	Tiempo requerido para aprender a utilizar el framework	41 horas	46 horas	8.63	8.47
Valor parcial	Laravel	7.32	Valor final	Laravel	1.83
	Django	7.27		Django	1.82

Tabla 5: Datos obtenidos de la característica Portabilidad

Sub Característica	Criterio	Valor Obtenido		Ponderación (/10)	
		Laravel	Django	Laravel	Django
Capacidad de ser Instalado.	Eficiencia en el tiempo de instalación en Windows	18 min y 15 seg	23 min y 7 seg	6.96	6.15
	Eficiencia en el tiempo de instalación en Linux	16 min y 21 seg	23 min y 4 seg	7.28	6.16
	Facilidad de instalación en Windows	1	1	10	10
	Facilidad de instalación en Linux	2	1	5	10
Valor parcial	Laravel	7.31	Valor final	Laravel	1.10
	Django	8.08		Django	1.21

Tabla 6: Datos obtenidos de la característica Seguridad

Sub Característica	Criterio	Valor Obtenido		Ponderación (/10)	
		Laravel	Django	Laravel	Django
Confidencialidad	Capacidad de control de acceso	2	2	6.67	6.67
	Encriptación de datos	1	1	10	10
Autenticidad	Métodos de autenticación	1	1	10	10
Valor parcial	Laravel	8.89	Valor final	Laravel	2.22
	Django	8.89		Django	2.22

tablecidos para la medición de este parámetro

works en cada una de las características.

Resultado final de la evaluación

A continuación, en la Tabla 7, se presenta las puntuaciones finales que obtuvieron los frameworks Laravel y Django posterior a su evaluación. Estos valores son el resultado de la sumatoria de los valores finales obtenidos por los frame-

Prueba de Hipótesis

En base a los resultados obtenidos a partir de la evaluación de los frameworks, se procedió a realizar la comprobación de la hipótesis que fue previamente establecida, obteniendo lo siguiente: En la presente investigación se comprueba

Tabla 7: Resultados finales de la evaluación de los frameworks

Resultado Final de la Evaluación			
Nombre del Framework	Calidad del Framework	Nivel de Puntuación	Grado de Satisfacción
Laravel	7,29	Aceptable	Satisfactorio
Django	8,16	Aceptable	Satisfactorio

que “el análisis comparativo para la evaluación de los frameworks Laravel y Django permitió determinar el framework más adecuado para el desarrollo de aplicaciones web”. Esta afirmación es verdadera puesto que el análisis comparativo permitió establecer a Django como el framework óptimo para el desarrollo de aplicaciones web, esta información se puede corroborar en la Tabla 7, donde se muestra que Django obtuvo un valor de 8.16, lo cual representa el 81.6% frente al 7.29 que simboliza el 72.9% obtenido por Laravel en la evaluación de las características establecidas para la comparación de los dos entornos de desarrollo web.

DISCUSIÓN

A lo largo de los últimos años se ha podido evidenciar que el uso de los frameworks para el desarrollo de aplicaciones web es cada vez más frecuente, esto debido a las crecientes exigencias requeridas por la web y los usuarios de la misma, necesitando de nuevas características y funcionalidades que posibiliten llevar el desarrollo a nuevos niveles en los cuales las aplicaciones sean cada vez más accesibles para los internautas y que sean fáciles de realizar para los programadores de dichas aplicaciones y sitios web. El desarrollo del presente trabajo permitió verificar que existen muchos frameworks creados para facilitar la tarea del desarrollo, sin embargo se debe tener en cuenta que cada uno de ellos es distinto el uno del otro y que las características que los diferencian, muchas de las veces dificultan la elección de alguno, mayormente cuando se desea desarrollar aplicaciones que cuenten con estándares de calidad.

El trabajo desarrollado en la presente investigación ha sido enfocado en la evaluación de los frameworks Laravel y Django, con base en los criterios establecidos por el modelo de calidad de la Norma ISO/IEC 25010. Por tal razón, se planteó un escenario de experimentación a través del desarrollo de una aplicación de prueba para determinar cuál de los dos frameworks es el mejor en el desarrollo de aplicaciones web, con lo cual se pudo determinar que Django es el mejor entorno de desarrollo, ya que obtuvo una mejor calificación frente a Laravel en cuanto a los criterios que fueron evaluados a lo largo del desarrollo de la aplicación implementada.

Django demostró ser mejor en cuanto a rendimiento en el tiempo de ejecución de operaciones CRUD, asimismo, la utilización de recursos fue menor y las líneas de código requeridas para la implementación de determinadas operaciones fue considerablemente menor en comparación a las que fueron requeridas por Laravel para la implementación de las mismas operaciones. Por otro lado, aunque Laravel obtuvo una mejor calificación en cuanto a la característica de usabilidad; Django resultó ser mejor en cuanto a portabilidad y obtuvo la misma calificación que Laravel con respecto a la seguridad.

Finalmente, aunque el desarrollo de la aplicación realizada en la experimentación fue más rápido con la utilización del

framework Laravel, sin embargo, al realizar la evaluación de los dos entornos, fue Django el que obtuvo una mejor calificación en la valoración de los criterios establecidos, quedando como la herramienta más viable para el desarrollo de aplicaciones web, por lo cual se acepta la hipótesis planteada para la presente investigación.

CONCLUSIONES

El modelo de calidad de la norma ISO/IEC 25000 se adecúa de manera correcta a la evaluación de frameworks, garantizando que dicha evaluación se realice de forma correcta bajo estándares de calidad, ya que provee un modelo definido y flexible, el cual permite elegir las características que se requiera, así como quitar o aumentar ciertos criterios que se crea pertinente en base a las necesidades del proyecto.

Con la finalidad de escoger el framework adecuado para la implementación de aplicaciones web, se establecieron criterios para la comparación de los frameworks, sobre: tiempo de ejecución de operaciones CRUD, líneas de código, utilización de la CPU, utilización de la memoria RAM, efectividad de la documentación del usuario o ayuda del sistema, verificación de entradas válidas, curva de aprendizaje, eficiencia en el tiempo de instalación, facilidad de instalación, capacidad de control de acceso, encriptación de datos y métodos de autenticación, que servirán como base para que los programadores puedan seleccionar la mejor herramienta para desarrollar sus aplicaciones web.

Para realizar el análisis comparativo definido en la presente investigación, se creyó pertinente poner a prueba los frameworks seleccionados, para lo cual se escogió a un grupo de estudiantes para desarrollar una aplicación de prueba, misma que permitió llevar a cabo la evaluación y comparación de Laravel y Django; durante este proceso se pudo determinar que cada framework maneja archivos de configuración diferentes, y que ambos entornos ofrecen interfaces amigables y una buena documentación para facilitar al programador la tarea del desarrollo.

En el desarrollo del experimento se pudo determinar que aunque Django es un entorno muy fácil de aprender, debido a que está basado en el lenguaje de programación Python, no obstante, el desarrollo de la aplicación de prueba para la evaluación de los frameworks fue terminada antes con el uso del framework Laravel; esto se debió a que los estudiantes que trabajaron con Laravel tenían más conocimiento y experiencia en el desarrollo de este tipo de aplicaciones y estaban más familiarizados con la utilización de la herramienta, de tal forma que dicha aplicación fue terminada en 41 horas, mientras que la aplicación desarrollada en Django fue terminada en un tiempo de 46 horas.

El desarrollo del presente trabajo permitió determinar que el Framework Django es el más adecuado para el desarrollo de aplicaciones web, de acuerdo a los criterios seleccionados de las características de Rendimiento, Usabilidad, Portabilidad y Seguridad, puesto que, de acuerdo a la evaluación realizada, este obtuvo un valor del 81.6% frente a Laravel, el cual obtuvo un valor del 72.9%.

AGRADECIMIENTOS

A la Universidad Nacional de Loja, a los docentes de la Carrera de Ingeniería en Sistemas y de manera especial al Ing. Edison Leonardo Coronel Romero, Mg. Sc., por su excelente asesoría y apoyo para llevar a cabo el desarrollo del presente trabajo.

Ovando Ortega, D. J. (2019). Bootstrap y Laravel, herramientas para el desarrollo de aplicaciones web. , 8(5), 55.

Standardization, I. O. f. (2021). Normas ISO/IEC 25000. <https://iso25000.com/index.php/normas-iso-25000>

CONTRIBUCIONES DE LOS AUTORES

Conceptualización, REH; metodología, REH; análisis formal, REH; investigación, REH; recursos, REH; curación de datos, REH; redacción — preparación del borrador original, REH; redacción — revisión y edición, REH. Todos los autores han leído y aceptado la versión publicada del manuscrito. Raquel Espinosa-Hurtado: REH.

FINANCIAMIENTO

El presente estudio fue financiado por el autor del mismo.

REFERENCIAS

Arcos Chalán, J. S., y Chicaiza Inguillay, D. E. (2016). Análisis comparativo de los Frameworks Laravel y Codeigniter para la implementación del sistema de gestión de concursos de méritos y oposición en la Universidad Nacional de Chimborazo. 190. <http://dSPACE.unach.edu.ec/handle/51000/2943>

Builes, J. A. J., Bedoya, D. L. R., y Bedoya, J. W. B. (2019). Metodología de desarrollo de software para plataformas educativas robóticas usando ROS-XP. *Revista Politécnica*, 15(30), 55–69. <https://doi.org/10.33571/RPOLITEC.V15N30A6>

Cabrera León, P. A. (2019). Desarrollo de una aplicación web utilizando el Framework Laravel para el control de existencias de los activos fijos de la Federación Deportiva de Morona Santiago. <http://dSPACE.esPOCH.edu.ec/handle/123456789/13635>

Caldera Vergara, R. (2017). Estudio del framework de desarrollo web Django. <https://ebuah.uah.es/dSPACE/handle/10017/32018>

García León, D., Beltrán Benavides, A., y Fernández Carrasco, O. (2016). Un enfoque actual sobre la calidad del software. *ACIMED*, 3(3), 40–42. <http://scielo.sld.cu/scielo.php?script=sciarttextpid=S1024-94351995000300005&lng=es&nrm=isot&lng=es>

Gómez García, D. E. (2018). Desarrollo del sistema de requisiciones para la empresa hidroeléctrica Abanico S.A. Aplicando el entorno de programación Node.js. <http://dSPACE.esPOCH.edu.ec/handle/123456789/9110>

Molina Ríos, J. R., Loja Mora, N. M., Zea Ordóñez, M. P., y Loaiza Sojos, E. L. (2016). Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python. *Revista Latinoamericana de Ingeniería de Software*, 4(4), 201. <https://doi.org/10.18294/relais.2016.201-207>