

# Frameworks backend en la plataforma tecnológica de gestión de comunidades académicas y recreativas de la Universidad Técnica de Babahoyo

*Backend frameworks in a technological platform for the management of academic and recreational communities of the Universidad Técnica de Babahoyo*

<https://doi.org/10.5281/zenodo.15243051>

**AUTORES:** José Teodoro Mejía Viteri<sup>1\*</sup>

Oscar Adonis Almeida Santillán<sup>2</sup>

Yefer Eduardo Moran Vera<sup>3</sup>

Rene Alejandro Herrera Barcos<sup>4</sup>

**DIRECCIÓN PARA CORRESPONDENCIA:** [jmejia@utb.edu.ec](mailto:jmejia@utb.edu.ec)

**Fecha de recepción:** 09 / 01 / 2025

**Fecha de aceptación:** 27 / 03 / 2025

## RESUMEN

En la actualidad los Frameworks backend han surgido como una solución popular a la desarrollar proyectos robustos y escalables, proporcionando una estructura y un conjunto de herramientas que agilizan el proceso de desarrollo de aplicaciones. En este contexto, ante las complicaciones presentes en el proceso de desarrollo del proyecto semillero “Plataforma tecnológica para la gestión de comunidades académicas y recreativas de la Universidad Técnica de Babahoyo” se ha buscado solución a través de una propuesta de Framework backend. A la vez que la plataforma crezca en usuarios y comunidades, la seguridad de sus

---

<sup>1\*</sup> Universidad técnica de Babahoyo, [jmejia@utb.edu.ec](mailto:jmejia@utb.edu.ec)

<sup>2</sup> Universidad técnica de Babahoyo, [olamedias@utb.edu.ec](mailto:olamedias@utb.edu.ec)

<sup>3</sup> Universidad técnica de Babahoyo, [ymoran@utb.edu.ec](mailto:ymoran@utb.edu.ec)

<sup>4</sup> Universidad técnica de Babahoyo, [rherrera@utb.edu.ec](mailto:rherrera@utb.edu.ec)

datos y eficiencia de la plataforma es una prioridad fundamental. Este estudio no solo aborda aspectos teóricos de las tecnologías usadas en la plataforma y de los frameworks backend ya que también propone una solución a implementar. El objetivo del presente artículo mejorará la plataforma tecnológica con el fin de que sea más eficiente, segura y escalable buscando satisfacer las necesidades que tenga la comunidad universitaria en el futuro.

**Palabras clave:** *Frameworks backend, plataforma tecnológica, proyecto semillero, seguridad, escalabilidad y rendimiento*

## ABSTRACT

Currently, backend frameworks have emerged as a popular solution to develop robust and scalable projects, providing a structure and a set of tools that streamline the application development process. In this context, given the complications present in the development process of the seed project “Technological platform for the management of academic and recreational communities of the Technical University of Babahoyo”, a solution has been sought through a proposed backend framework. While the platform grows in users and communities, the security of its data and efficiency of the platform is a fundamental priority. This study not only addresses theoretical aspects of the technologies used in the platform and backend frameworks but also proposes a solution to implement. The objective of this article is to improve the technological platform in order to make it more efficient, secure and scalable to meet the needs of the university community in the future.

**Keywords:** *Backend frameworks, technological platform, seed project, security, scalability and performance*

## INTRODUCCIÓN

En estos últimos años se ha tenido un incremento notable correspondiente al desarrollo de aplicaciones web en el mundo. Cabe recalcar que las plataformas tecnológicas son infraestructuras digitales que posibilitan la interacción entre dos o más grupos, ubicándose como intermediarios entre diferentes usuarios, clientes, anunciantes, prestaciones de servicios, etc. (Hernández, 2021). Por esta razón, las herramientas y tecnologías que facilitan la creación de backends robustos y escalables han tenido cada vez más demanda.

La razón es que los marcos de Backend brindan herramientas y estructuras que facilitan el proceso de desarrollo, convirtiéndose en una solución bastante popular. En el Ecuador, el desarrollo de aplicaciones web ha tenido un incremento significativo, la adopción de frameworks backends por parte de desarrolladores y empresas ecuatorianas ha ido en aumento puesto que, se busca dar soluciones a partir de la tecnología en varios sectores dando como resultado el crecimiento de un país digitalizado (Salazar Chacón, 2021).

La “Plataforma tecnológica para la gestión de comunidades académicas y recreativas de la Universidad Técnica de Babahoyo” se encuentra en etapa de desarrollo en el proyecto semillero. La elección de la tecnología encargada del backend fue basada en los requerimientos y factibilidades del equipo, sin contemplar a los funcionarios del departamento de sistemas de la Universidad, mismos que serán los encargados de la continuidad del proyecto en caso de tener aceptación. A medida que el equipo avanzaba en el proceso de desarrollo y pruebas, se hicieron evidente algunas complicaciones al momento de realizar peticiones a la base de datos y enviarlas al lado del cliente, esto sumado a los bajos parámetros de seguridad que tenía proyecto y el tiempo excesivo que consumía el desarrollo del mismo los llevó a cuestionarse si la tecnología seleccionada es la mejor opción para garantizar la seguridad, rendimiento y escalabilidad de la aplicación a futuro. Llegando a plantear el siguiente problema: ¿Qué Framework backend es el mejor para integrar en la “Plataforma tecnológica de gestión de comunidades académicas y recreativas de la Universidad Técnica de Babahoyo”?; por tanto el objetivo general es generar el backend de la “Plataforma tecnológica de gestión de comunidades académicas y recreativas de la Universidad Técnica de Babahoyo” con la finalidad de mejorar la seguridad, escalabilidad y rendimiento a través un framework backend, del anterior se derivan los siguientes objetivos específicos.

1. Analizar los frameworks backend para la “Plataforma tecnológica de gestión de comunidades académicas y recreativas de la Universidad Técnica de Babahoyo” basándose en información obtenida de estudios y sitios oficiales para identificar sus características, ventajas, desventajas y casos de uso.
2. Determinar el framework backend más adecuado por medio de indicadores de seguridad, escalabilidad y rendimiento para evaluar las debilidades y fortalezas de cada uno.

3. Desarrollar el backend para la “Plataforma tecnológica de gestión de comunidades académicas y recreativas de la Universidad Técnica de Babahoyo” a través los resultados obtenidos de la evaluación para su implementación en el proyecto.

## **METODOLOGÍA**

La metodología utilizada en esta investigación se fundamenta en diversos enfoques para garantizar que el estudio sea riguroso y orientado a resultados. Según el nivel de análisis, se clasifica como una investigación aplicada, enfocada en crear soluciones que sean de utilidad.

Lo que conlleva al desarrollo del Backend de la plataforma empleando el framework apropiado, elegido después de un análisis exhaustivo, este último es también conocido como un marco de trabajo usado por los desarrolladores para elaborar programas por medio buenas prácticas, de esta manera facilita y acelera de las fases de desarrollo y evitando los duplicados en el código (Bravo Rivera, 2021).

Con respecto al lugar, es tratada desde un enfoque de campo la cual se basa en la recopilación directa de datos por medio de entrevistas de preguntas cerradas hacia los desarrolladores implicados en el proyecto semillero, permitiendo contextualizar las habilidades y capacidades del equipo en base a las tecnologías que implementaron en el proyecto. Asimismo, es complementada con una investigación del tipo bibliográfica realizando un análisis datos obtenidos de fuentes escritas y visuales proveídas por instituciones y expertos, con la finalidad de identificar características, ventajas, desventajas y casos de uso de los frameworks Backend tomados en cuenta (Florez, 2024).

Para conseguir este objetivo, se pondrá en práctica el método de comparación, permitiendo la evaluación varios frameworks Backend dependiendo de criterios como escalabilidad, rendimiento y seguridad. individualmente se le asignará un rango de calificación para determinar con carácter objetivo sus fortalezas y debilidades (Caiza Samaniego, 2023).

Por medio del empleo del enfoque cuantitativo, que se basa en la medición y evaluación de datos cuantificables, se logrará la comprobación de la hipótesis mediante el análisis de los datos recolectados en las entrevistas y las evaluaciones de los marcos seleccionados (Abad-Segura et al., 2022).

El proceso para gestionar los datos contemplará varias actividades programadas, tales como: interacción con el tema a los involucrados en el proyecto inicial, análisis de las tecnologías empleadas, análisis de los frameworks Backend más famosos en el mercado, evaluación para establecer cuál es el más adecuado mediante indicadores, selección del framework más adecuado y desarrollo de la propuesta (Sánchez Molina et al., 2021).

Después, esta propuesta será presentada a los desarrolladores donde, para evaluar su nivel de satisfacción se realizarán entrevistas organizando los datos recopilados en una hoja de cálculo por medio de tablas etiquetadas y filtros aplicados para su correcto procesamiento y demostración gráfica presentando los resultados que se obtuvieron con claridad y de forma organizada (Damian et al., 2021).

Con respecto a técnicas e instrumentos de medición estarán presentes: Revisión documental, evaluación comparativa y entrevistas emitas al personal encargado del desarrollo del proyecto. Todo esto con la finalidad de reunir datos esenciales acerca de los Frameworks backend para consiguiente evaluar la satisfacción de la propuesta ante los desarrollares.

Se utilizará un cuestionario de preguntas cerradas para medir el grado de satisfacción, tablas comparativas para organizar y visualizar las características de cada framework, y gráficos que representen visualmente los resultados de la evaluación bajo diferentes criterios.

En última instancia, se garantizará la ética en cada etapa del proyecto. Se citarán correctamente las fuentes bibliográficas, asegurando que la información sea reciente y concediendo crédito a los autores e instituciones. Durante la recopilación de datos, se pedirá el consentimiento informado de los participantes, garantizando que los datos obtenidos se empleen únicamente para propósitos académicos. No se recolectará datos personales de los entrevistados, y cualquier información que pudiera identificar a una persona será suprimida para asegurar la privacidad y confidencialidad.

## **RESULTADOS**

### **Tabla con análisis comparativo**

Se eligieron estos frameworks en base al conocimiento y experiencia de las personas que desarrollaron el proyecto.

Framework	Lenguaje	Ventajas	Desventajas	Referencia
Express js	JavaScript	- <b>Alto rendimiento:</b> Manejo eficiente de conexiones concurrentes gracias a su modelo asíncrono.	- <b>Callback hell:</b> Complejidad en el manejo de múltiples callbacks anidados.	( <i>Top Node.js Backend Frameworks for 2024</i> , 2024)
		- <b>Amplia comunidad:</b> Gran cantidad de paquetes y módulos disponibles.	- <b>No apto para tareas CPU-intensivas:</b> Rendimiento reducido en operaciones que requieren alto procesamiento.	
Django	Python	- <b>Desarrollo rápido:</b> Filosofía "batteries included" que proporciona herramientas listas para usar.	- <b>Monolítico:</b> Puede ser menos flexible debido a su estructura opinada.	N/A
		- <b>Seguridad:</b> Protecciones integradas contra vulnerabilidades comunes.	- <b>Curva de aprendizaje:</b> Requiere familiarización con sus convenciones y estructura.	N/A
Spring Boot	Java	- <b>Robustez:</b> Ideal para aplicaciones empresariales de gran escala.	- <b>Curva de aprendizaje:</b> Complejidad inicial debido a la amplitud del framework.	(«(PDF) An Analysis of the Significance of Spring Boot in The Market», 2024)
		- <b>Ecosistema amplio:</b> Integración con diversos proyectos de Spring.	- <b>Configuración:</b> Aunque simplificada, puede ser abrumadora para principiantes.	(«(PDF) Building Scalable Java Applications», 2024)

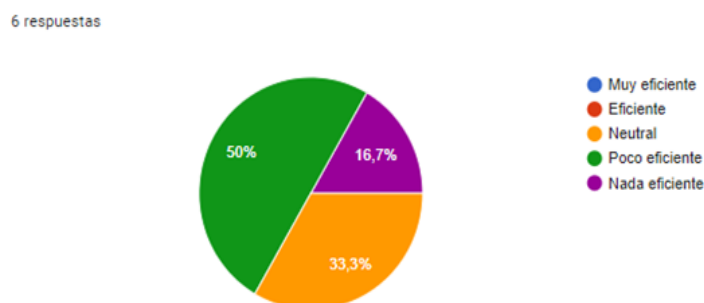
### Cuadro Comparativo: Desarrollo en Java con y sin Frameworks

Aspecto	Desarrollo con Spring Boot	Desarrollo sin Frameworks
<b>Productividad</b>	- <b>Aumento significativo:</b> Spring Boot simplifica la configuración y despliegue de aplicaciones Java, reduciendo el tiempo de desarrollo. (Casero, 2024)	- <b>Menor velocidad:</b> Requiere desarrollar manualmente funcionalidades que los frameworks ya ofrecen, incrementando el tiempo de desarrollo.
<b>Curva de Aprendizaje</b>	- <b>Pronunciada:</b> Es necesario invertir tiempo en comprender la estructura y funcionalidades de Spring Boot (Richardson, 2018).	- <b>Variable:</b> Depende de la complejidad del proyecto y de las tecnologías utilizadas.
<b>Flexibilidad y Personalización</b>	- <b>Moderada:</b> Spring Boot ofrece una estructura predefinida que puede limitar la personalización completa, aunque permite configuraciones específicas.	- <b>Alta:</b> Permite una personalización total, adaptando cada componente según las necesidades específicas del proyecto.
<b>Mantenimiento y Consistencia</b>	- <b>Mejorados:</b> Fomenta buenas prácticas y una estructura coherente, facilitando la colaboración y el mantenimiento a largo plazo.	- <b>Desafiantes:</b> Sin una estructura estándar, el código puede volverse inconsistente, dificultando el mantenimiento futuro (Freeman et al., 2021).
<b>Dependencia Externa</b>	- <b>Elevada:</b> Se depende de las actualizaciones y soporte de la comunidad de Spring Boot, lo que puede generar problemas si el framework queda obsoleto o cambia drásticamente.	- <b>Nula:</b> No hay dependencia de terceros, otorgando total control sobre el desarrollo y evolución de la aplicación.

## **Análisis de los resultados de las preguntas realizadas en la entrevista de preguntas cerradas**

Se entrevistó a 6 integrantes del proyecto mediante una serie de preguntas cerradas donde se identificaron patrones y tendencias, evaluando aspectos clave que ayudaron a la decisión de optar por un Framework backend.

### **1. ¿Qué tan eficiente considera la tecnología actual que se usa para desarrollar en equipo el backend del proyecto semillero?**

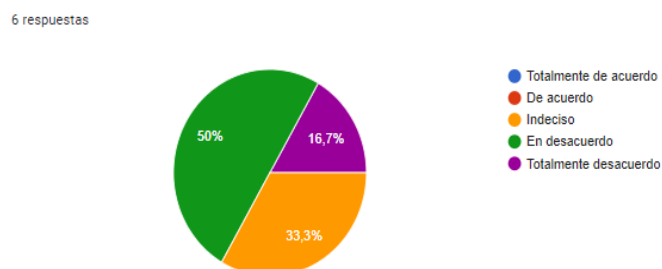


**Figura 1.** Grafica de resultados de la pregunta uno

**Nota:** Adaptado de los resultados obtenidos de la entrevista

En esta pregunta el 50% de los entrevistados consideran poco eficiente la tecnología actual que se usa para desarrollar en equipo el backend del proyecto semillero, el 33,3% indicó estar neutral ante la pregunta mientras que el 16,7% indicó que la tecnología desarrollada actualmente en el semillero es nada eficiente.

### **2. ¿Está de acuerdo que el backend actual cumple con todos los requisitos de seguridad establecidos para el proyecto semillero?**

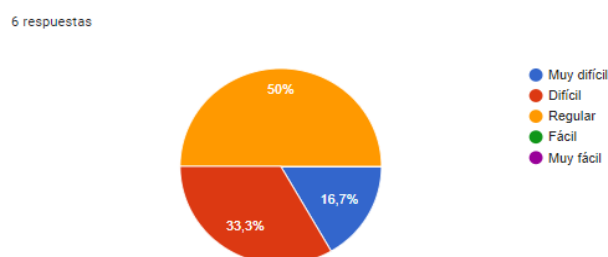


**Figura 2.** Grafica de resultados de la pregunta dos

**Nota:** Adaptado de los resultados obtenidos de la entrevista

De las respuestas se puede observar que el 50% de los entrevistados están en desacuerdo que el backend actual cumple con todos los requisitos de seguridad establecidos para el proyecto semillero, seguido del 33,3% que se mostró indeciso, solo un 16,7% declaró estar totalmente en desacuerdo. En base a estos resultados se concluyó que existe un desacuerdo en el cumplimiento de requisitos del proyecto desarrollado para el backend.

### 3. ¿Qué tan difícil es integrar nuevas funcionalidades en el backend del proyecto semillero con la tecnología actual?



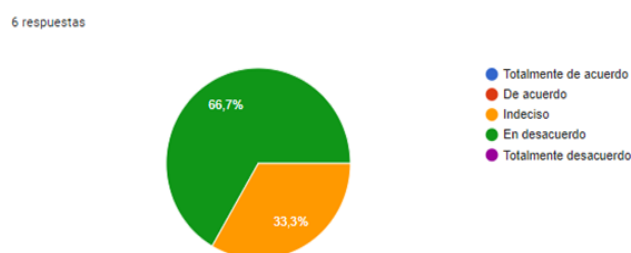
**Figura 3.** Grafica de resultados de pregunta tres

**Nota:** Adaptado de los resultados obtenidos de la entrevista

#### Análisis:

Al realizar un análisis de las respuestas se demostró que el 50% de entrevistados expresaron que integrar nuevas funcionalidades en el backend del proyecto semillero con la tecnología actual tiene una dificultad “Regular” sin embargo, el 33,3% respondió que es “Difícil” tomando en cuenta el contratiempo que estaban teniendo en base al cronograma de actividades siendo confirmado por el 16,7% de ellos que respondieron “Muy difícil”.

### 4. ¿Está de acuerdo que la solución del backend actual es fácilmente escalable y puede adaptarse a futuros crecimientos del proyecto semillero?



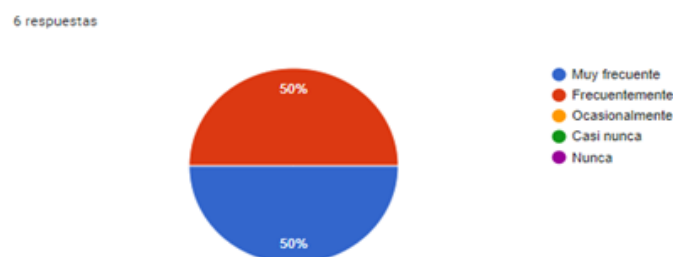
**Figura 4.** Grafica de resultados de la pregunta cuatro

**Nota:** Adaptado de los resultados obtenidos de la entrevista



**Análisis:**

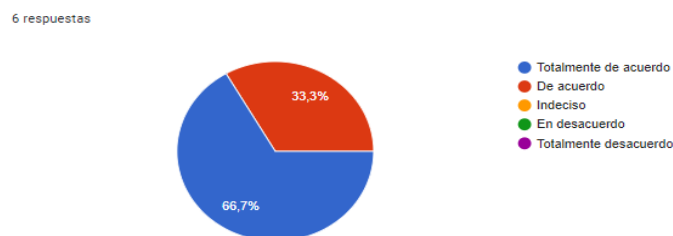
En esta pregunta el 66,7% de los entrevistados respondieron estar en desacuerdo en que el backend actual es fácilmente escalable y puede adaptarse a futuros crecimientos del proyecto semillero mientras que el 33,3% expresaron estar totalmente en desacuerdo, esto es debido a la dificultad que presentaban al momento de agregar funciones a medida que avanzaban en la etapa de desarrollo del proyecto semillero como lo expresan en la pregunta anterior.

**5. ¿Con qué frecuencia a utilizado frameworks backend?****Figura 5.** Grafica de resultados de pregunta cinco

**Nota:** Adaptado de los resultados obtenidos de la entrevista

**Análisis:**

Las respuestas de esta pregunta revelaron que todos los entrevistados frecuentan usar frameworks backend ya sea para tareas, trabajos o incluso proyectos personales, gracias a esto es posible que un framework sea la solución que ellos necesitan.

**6. ¿Estaría de acuerdo a considerar el uso de un framework backend en el proyecto semillero si este demostrara beneficios en términos de escalabilidad, seguridad y rendimiento?****Figura 6.** Grafica de resultados de pregunta seis

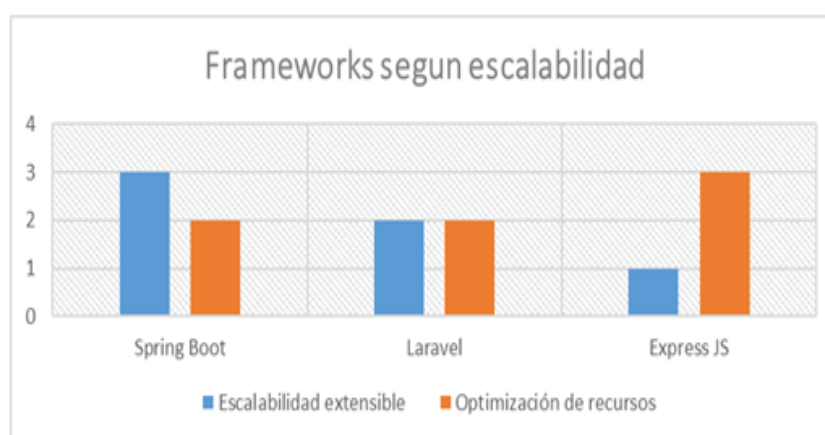
**Nota:** Adaptado de los resultados obtenidos de la entrevista

Al realizar un análisis de las respuestas obtenidas de los entrevistados se muestra que la mayoría está a favor de utilizar un framework siempre que se demuestre los beneficios en términos de escalabilidad, seguridad y rendimiento que este puede aportar en el proyecto semillero.

### **Análisis de resultados del marco comparativo de Spring Boot, Laravel y Express JS**

Se hizo una revisión documental en base a los resultados obtenidos de la encuesta realizada por el Stack Overflow en el año 2024.

#### **Escalabilidad**



**Figura 7.** Grafica de resultados de frameworks según escalabilidad

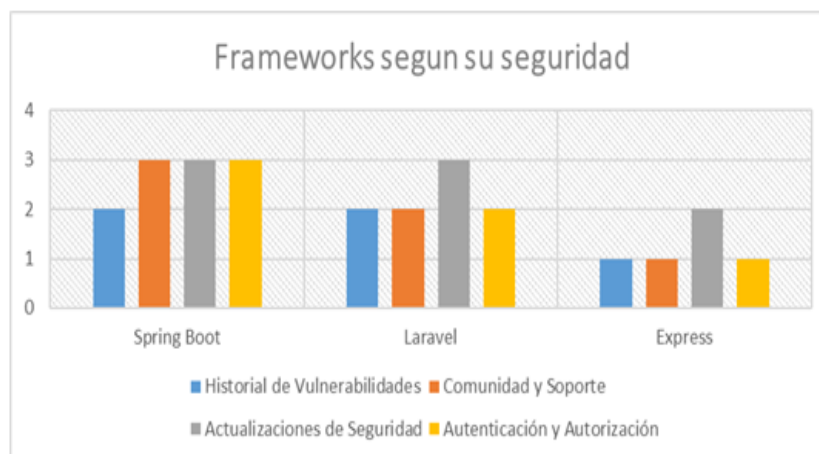
**Nota:** La figura muestra los resultados obtenidos de la tabla de comparativa de Framework según su escalabilidad

De la calificación Spring Boot obtuvo una calificación de 3 en escalabilidad extensible por tener ser fácil de integrar, y poseer una arquitectura modular y una puntuación de 2 en optimización de recursos por ser eficiente sin embargo posee un consumo mayor de memoria.

Laravel obtuvo una calificación de 2 en cuanto a escalabilidad y optimización de recursos por tener desafíos ante proyectos muy grandes y por ser menos eficiente en alta concurrencia al estar basado en PHP.

Express obtuvo una calificación de 1 en cuanto a escalabilidad extensible por no tener una estructura predefinida lo que dificulta mucho en proyectos grandes y un 3 en optimización de recurso por ser minimalista y eficiente en gestión de solicitudes concurrentes. Dando como resultado a Spring Boot el ganador en cuanto a escalabilidad por tener más puntaje.

## Seguridad



**Figura 8.** Grafica de resultados de frameworks según su seguridad

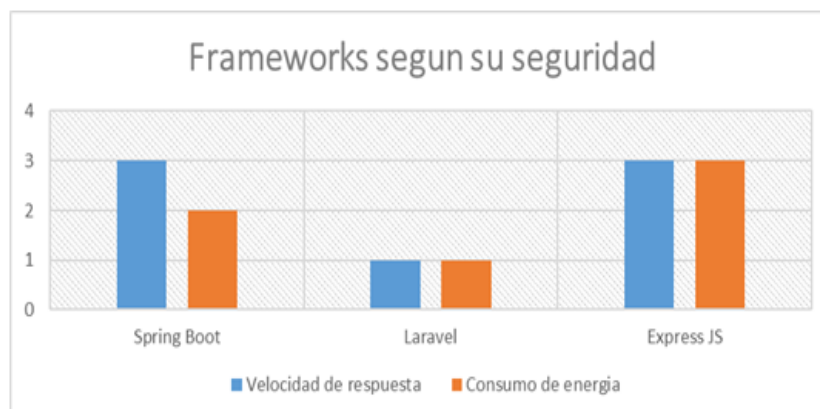
**Nota:** La figura muestra los resultados obtenidos de la tabla de comparativa de Framework según su seguridad

Spring Boot obtuvo una calificación de 2 por tener un buen historial, pero con algunas vulnerabilidades significativas las cuales no eran recurrentes, pero sí graves. Tanto en actualización de seguridad, comunidad y autenticación obtuvo una calificación de 3 por tener actualizaciones rápidas y frecuentes además de tener una comunidad activa y gran soporte corporativo así mismo por tener su componente de Spring Security el cual es robusto y flexible.

Laravel destacó por ser una comunidad vibrante y buen soporte a su vez obtuvo una calificación intermedia por tener una cantidad de vulnerabilidad más extensa en comparación que Spring Boot poseer actualizaciones semestrales y tener buenas capacidades sin embargo se necesitan de algunos componentes externos para la mayoría de ellas.

En este caso Express JS obtuvo una calificación de 1 en su mayoría por tener un mayor riesgo con dependencias externas, actualizaciones menos frecuentes y de requerimientos de configuraciones manuales. Dando como resultado nuevamente a con el puntaje más alto Spring Boot

## Rendimiento

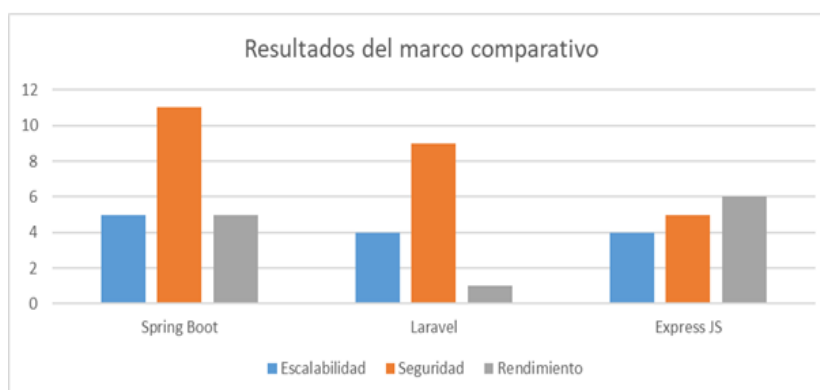


**Figura 9.** Grafica de resultados de frameworks según su rendimiento

**Nota:** La figura muestra los resultados obtenidos de la tabla de comparativa de Framework según su rendimiento

Express JS en rendimiento demostró ser mejor que los demás Frameworks comparados siendo ser el más rápido en la mayoría de los escenarios de prueba con un consumo de energía moderado, aun así Spring Boot se mantuvo en velocidades de respuestas superando a Express en algunos casos pero no fue el más eficiente en cuanto consumo de energía en algunas pruebas, por el contrario Laravel a pesar un Framework consistente tuvo velocidades de respuesta algo lentas en comparación con los demás Frameworks sumado a su alto consumo de energía le dieron una calificación baja.

## Resultados finales del marco comparativo



**Figura 10.** Resultados del marco comparativo

**Nota:** La figura muestra los resultados obtenidos de la tabla de resultados de marco comparativo

Sumado los valores obtenidos de las comparaciones anteriores por cada característica se llegó a determinar que el mejor Framework backend para la plataforma tecnológica del proyecto semillero sería Spring Boot.

## Comparación del backend actual en Java con el backend en Spring Boot

### Crear una comunidad

#### Versión en Java

```
@WebServlet(name = "SvCrearComunidad", urlPatterns = {"/SvCrearComunidad"})
public class SvCrearComunidad extends HttpServlet {
    private ComunidadService comunidadService = new ComunidadService();

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String nombre = req.getParameter("nombre");
        String descripciongrupo = req.getParameter("descripciongrupo");
        String imagenportadaBase64 = req.getParameter("imagenportada"); // assuming it's base64 encoded
        byte[] imagenportada = Base64.getDecoder().decode(string: imagenportadaBase64);

        Comunidad comunidad = new Comunidad(comid:0, nombre, descripciongrupo, imagenportada);
        boolean isAdded = comunidadService.addComunidad(comunidad);

        resp.setContentType("application/json");
        PrintWriter out = resp.getWriter();
        if (isAdded) {
            out.print(string: "{\"status\": \"success\", \"message\": \"Comunidad creada exitosamente\"}");
        } else {
            out.print(string: "{\"status\": \"error\", \"message\": \"No se pudo crear la comunidad\"}");
        }
        out.flush();
    }
}
```

**Figura 11.** Creación de una comunidad en Java  
**Nota:** Creación de comunidad por Java usando Servlet

```
public boolean addComunidad(Comunidad comunidad) {
    boolean isAdded = false;
    Connection connection = null;
    try {
        connection = conexionDB.getConexion();
        String sql = "INSERT INTO comunidades (com_var_nombre, com_var_descripciongrupo, com_byt_imagenportada) VALUES (?, ?, ?)";
        try (PreparedStatement ps = connection.prepareStatement(string: sql)) {
            ps.setString(1, comunidad.getNombre());
            ps.setString(2, comunidad.getDescripciongrupo());
            ps.setBytes(3, comunidad.getImagenportada());
            int rowsAffected = ps.executeUpdate();
            isAdded = rowsAffected > 0;
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        if (connection != null) {
            conexionDB.close();
        }
    }
    return isAdded;
}
```

**Figura 12.** Servicio de creación de comunidad en Java  
**Nota:** Creación de comunidad directa a la base de datos

## Versión en el framework

```
@PostMapping(value = @RequestMapping("comunidad/nueva", consumes = "multipart/form-data") new *  
public ResponseEntity<?> createComunidad(@RequestPart("archivo") MultipartFile file, ComunidadDTO comunidadDTO) {  
    return ResponseEntity.ok(comunidadService.createComunidad(comunidadDTO, file));  
}
```

**Figura 13.** Creación de comunidad en Spring Boot

**Nota:** Creación de comunidad por medio de un controlador

```
@Override @usage new *  
public Boolean createComunidad(ComunidadDTO comunidadDTO, MultipartFile file) {  
    Comunidad comunidad = Comunidad.builder()  
        .nombre(comunidadDTO.getNombre())  
        .descripciongrupo(comunidadDTO.getDescripciongrupo())  
        .imagenportada(imagen.compressImageFromFile(file))  
        .build();  
    comunidadDAO.save(comunidad);  
    return true;  
}
```

**Figura 14.** Servicio de creación de comunidad en Spring Boot

**Nota:** Creación de comunidad por medio de mapping y validación del token

La creación de una comunidad en Java además de ser tediosa por la configuración manual del servlet que se necesita para la solicitud http es insegura por interactuar directamente con la base de datos ocasionando posibles ataques a futuro, sin embargo, en Spring Boot basta con pocas líneas para la creación de un controlador que solicite los datos requeridos para creación de una comunidad no sin antes solicitar la autenticación por token para luego pasar a ser procesados a través del mapping.

## Actualizar datos de usuario

### Versión en Java

```
@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    String identificacion = req.getParameter("identificacion");
    String nombrespellidos = req.getParameter("nombrespellidos");
    String nombresusuario = req.getParameter("nombresusuario");
    String imagenperfilBase64 = req.getParameter("imagenperfil");
    byte[] imagenperfil = Base64.getDecoder().decode(string:imagenperfilBase64);
    int genid = Integer.parseInt(string:req.getParameter("genid"));
    String telefono = req.getParameter("telefono");
    String celular = req.getParameter("celular");
    String correo = req.getParameter("correo");
    int proid = Integer.parseInt(string:req.getParameter("proid"));
    int canid = Integer.parseInt(string:req.getParameter("canid"));
    String direccion = req.getParameter("direccion");
    String clave = req.getParameter("clave");
    int treid = Integer.parseInt(string:req.getParameter("treid"));
    int tusid = Integer.parseInt(string:req.getParameter("tusid"));
    int stsid = Integer.parseInt(string:req.getParameter("stsid"));
    Usuario updatedUsuario = new Usuario(identificacion, nombrespellidos, nombresusuario, imagenperfil, genid, telefono, celular, correo, proid, canid, direccion, clave, treid, tusid, stsid);
    boolean isUpdated = usuarioService.updateUsuario(updatedUsuario);
    resp.setContentType("application/json");
    PrintWriter out = resp.getWriter();
    if (isUpdated) {
        out.print(string:"{\n\"status\": \"success\", \"message\": \"Usuario actualizado exitosamente\"}");
    } else {
        out.print(string:"{\n\"status\": \"error\", \"message\": \"No se pudo actualizar el usuario\"}");
    }
    out.flush();
}
```

**Figura 15.** Actualizar datos de usuario en Java

**Nota:** Actualización de datos de usuario por Java usando Servlet

```
public boolean updateUsuario(Usuario updatedUsuario) {
    boolean isUpdated = false;
    Connection connection = null;
    try {
        connection = conexionDB.getConnection();
        String sql = "UPDATE usuarios SET usu_var_identificacion = ?, usu_var_nombrespellidos = ?, usu_var_nombresusuario = ?, usu_byt_imagenperfil = ? WHERE usu_genid = ?";
        try (PreparedStatement ps = connection.prepareStatement(string:sql)) {
            ps.setString(1, string:updatedUsuario.getIdentificacion());
            ps.setString(2, string:updatedUsuario.getNombrespellidos());
            ps.setString(3, string:updatedUsuario.getNombresusuario());
            ps.setBytes(4, bytes:updatedUsuario.getImagenperfil());
            ps.setInt(5, int:updatedUsuario.getGenid());
            ps.setString(6, string:updatedUsuario.getTelefono());
            ps.setString(7, string:updatedUsuario.getCelular());
            ps.setString(8, string:updatedUsuario.getCorreo());
            ps.setInt(9, int:updatedUsuario.getProid());
            ps.setInt(10, int:updatedUsuario.getCanid());
            ps.setString(11, string:updatedUsuario.getDireccion());
            ps.setString(12, string:updatedUsuario.getClave());
            ps.setInt(13, int:updatedUsuario.getTreid());
            ps.setInt(14, int:updatedUsuario.getTusid());
            ps.setInt(15, int:updatedUsuario.getStsid());
            ps.setString(16, string:updatedUsuario.getNombresusuario());
            int rowsAffected = ps.executeUpdate();
            isUpdated = rowsAffected > 0;
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return isUpdated;
}
```

**Figura 16.** Servicio de actualización de datos de usuario en Java

**Nota:** Actualización de datos de usuario directa a la base de datos

```
@PostMapping(path="/actualizarusuario") new *
public ResponseEntity<Boolean> actualizarusuario(@RequestBody UsuarioDTO usuarioDTO){
    return ResponseEntity.ok(usuarioService.actualizarUsuario(usuarioDTO));
}
```

**Figura 17.** Actualización de datos de usuario en Spring Boot

**Nota:** Actualización de datos de usuario por medio de un controlador

### Versión en el framework

```
public boolean actualizarUsuario(Integer id, UsuarioDTO usuarioDTO) {
    if (usuarioDAO.existsById(id)) {
        Usuario usuario = Usuario.builder()
            .identificacion(usuarioDTO.getIdentificacion())
            .nombresapellidos(usuarioDTO.getNombresapellidos())
            .nombreusuario(usuarioDTO.getNombreusuario())
            .genid(Genero.builder().genid(usuarioDTO.getGenid()).build())
            .telefono(usuarioDTO.getTelefono())
            .celular(usuarioDTO.getCelular())
            .correo(usuarioDTO.getCorreo())
            .proid(Provincia.builder().proid(usuarioDTO.getProid()).build())
            .canid(Canton.builder().canid(usuarioDTO.getCanid()).build())
            .direccion(usuarioDTO.getDireccion())
            .clave(passwordEncoder.encode(usuarioDTO.getClave()))
            .treid(TipoRegistro.builder().treid(usuarioDTO.getTreid()).build())
            .tusid(TipoUsuario.builder().tusid(usuarioDTO.getTusid()).build())
            .stsid(Estado.builder().stsid(1).build())
            .build();
        usuarioDAO.save(usuario);
        return true;
    } else {
        return false;
    }
}
```

**Figura 18.** Servicio de actualización de datos de usuario en Spring Boot

**Nota.** Actualización de datos de usuario por medio de mapping y validación del token

## DISCUSIÓN

Cuando se realizan comparaciones entre las perspectivas de distintos autores en base a la utilización de un Framework para el proceso de elaboración de aplicaciones basadas en Java, se logra apreciar cada uno de sus enfoques y recomendaciones presentes en sus investigaciones realizadas.

Comenzando con Choma et al. (2023) quienes realizaron un estudio comparando los marcos de Express, Django y Spring Boot, realizaron una evaluación de la eficiencia y fiabilidad de cada tecnología. Donde llegaron a la conclusión que Spring Boot destaca en términos de tiempo de procesamiento de solicitudes y alta fiabilidad, pero su porcentaje del procesamiento incorrecto de solicitudes es mayor que Express en ambientes de carga extrema.



En cambio, Vardhan (2024) desarrolló una prueba en base al rendimiento entre Django y Spring Boot, utilizando como herramienta de medición Apache Bench donde valora una API simple en cada framework. Los resultados obtenidos dieron como superior a Spring Boot en cuanto a tiempo de respuestas gracias a su mayor número de manejo de solicitudes por segundo y los menores tiempos que necesitaba para su conexión y procesamiento de datos.

Comparando estas perspectivas es evidente que el framework más adecuado para el desarrollo de aplicaciones basadas en Java es Spring Boot por lo que debe tratarse de forma cuidadosa las necesidades específicas del proyecto tales como experiencia de desarrollo y los objetivos a largo plazo, este último brinda beneficios en cuanto a consistencia y productividad, cabe recalcar que se debe tener presente las limitaciones y desafíos que se pueden presentar a futuro.

Estando comprobados los resultados de este estudio, se demostró un interés profundo por parte de los desarrolladores involucrados en el proyecto semillero por optar por un framework backend para la “Plataforma tecnológica de gestión de comunidades académicas y recreativas de la Universidad Técnica de Babahoyo”, sin embargo se debe estar conforme a las necesidades de escalabilidad, seguridad y rendimiento deseadas, respondiendo a la problemática inicial dando resultado al framework Spring Boot como el mejor en comparación con otros por sus características antes mencionadas, más aún porque los desarrolladores al poseer conocimientos previos en la tecnología en la que está basado pueden empezar a desarrollar sin una formación extensa, de esta manera el equipo se motivó en su desarrollo con los tiempos estimados sin contratiempos.

## **CONCLUSIONES**

En conclusión, mediante los datos obtenidos se determinó que los desarrolladores de la “Plataforma tecnológica de gestión de comunidades académicas y recreativas de la Universidad Técnica de Babahoyo” necesitaban del uso de un framework backend para agilizar su trabajo y cumplir con los requerimientos de escalabilidad, seguridad y rendimiento a tiempo.

Tras un análisis de los datos del marco comparativo se concluyó que el mejor framework backend para el proyecto semillero era Spring Boot por su gran capacidad en cargas de

trabajo teniendo en cuenta el uso a futuro de la plataforma por toda la comunidad universitaria.

Se concluye que el framework mejoró el backend permitiendo el uso de tokens de autenticación en el inicio de sesión, el mapeo de la base de datos para evitar consultas directas que provoquen posibles ataques de SQLInjection, gracias a su estructura ahora se puede agregar nuevos módulos o funciones sin que esto afecte al desempeño de la plataforma.

## **REFERENCIAS BIBLIOGRÁFICAS**

- Abad-Segura, E., González-Zamar, M. D., & López-Meneses, E. (2022). El proceso de toma de decisiones basado en métodos cuantitativos: Análisis de tendencias en el ámbito corporativo. *Revista de Métodos Cuantitativos para la Economía y la Empresa*, 34, 118-136. <https://doi.org/10.46661/revmetodoscuanteconempresa.5135>
- Bravo Rivera, R. I. (2021). Análisis comparativo entre frameworks, para el desarrollo de aplicaciones móviles multiplataformas [bachelorThesis, BABAHOYO: UTB, 2021]. <http://dspace.utb.edu.ec/handle/49000/9473>
- Caiza Samaniego, D. A. (2023). Desarrollo de una aplicación web para la difusión y venta de productos de la Organización MiKUY aplicando prácticas Devops. <http://dspace.esPOCH.edu.ec/handle/123456789/20893>
- Casero, A. (2024, septiembre 16). Spring Framework vs Spring Boot: ¿En qué se diferencian? <https://keepcoding.io/blog/spring-framework-vs-spring-boot-diferencias/>
- Choma, D., Chwaleba, K., & Dzieńkowski, M. (2023). THE EFFICIENCY AND RELIABILITY OF BACKEND TECHNOLOGIES: EXPRESS, DJANGO, AND SPRING BOOT. *Informatyka, Automatyka, Pomiar w Gospodarce i Ochronie Środowiska*, 13(4), Article 4. <https://doi.org/10.35784/iapgos.4279>
- Damian, D., Linåker, J., Johnson, D., Clear, T., & Blincoe, K. (2021). Challenges and Strategies for Managing Requirements Selection in Software Ecosystems. *IEEE Software*, 38(6), 76-87. *IEEE Software*. <https://doi.org/10.1109/MS.2021.3105044>
- Florez, E. D. M. (2024). Tecnología e Innovación en el Preescolar: Estrategias Ludo Pedagógicas Para Promover la.

- Freeman, E., Freeman, E., Sierra, K., Bates, B., Freeman, E., Robson, E., & man, F. li. (2021). Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software. O'Reilly Media, Incorporated.
- Hernández, S. J. (2021). El turismo en la era digital: Las aplicaciones móviles como herramienta de innovación. <http://ri.unsam.edu.ar/handle/123456789/1497>
- (PDF) An Analysis of the Significance of Spring Boot in The Market. (2024, diciembre 7). ResearchGate. <https://doi.org/10.1109/ICICT54344.2022.9850910>
- (PDF) Building Scalable Java Applications: An In-Depth Exploration of Spring Framework and Its Ecosystem. (2024). ResearchGate. <https://doi.org/10.61463/ijset.vol.4.issue3.103>
- Richardson, C. (2018). Microservices Patterns. <https://www.manning.com/books/microservices-patterns>
- Salazar Chacón, G. D. (2021). Hybrid Networking SDN y SD-WAN: Interoperabilidad de arquitecturas de redes tradicionales y redes definidas por software en la era de la digitalización [Tesis, Universidad Nacional de La Plata]. <https://doi.org/10.35537/10915/129910>
- Sánchez Molina, A. A., Murillo Garza, A., Sánchez Molina, A. A., & Murillo Garza, A. (2021). Enfoques metodológicos en la investigación histórica: Cuantitativa, cualitativa y comparativa. Debates por la historia, 9(2), 147-181. <https://doi.org/10.54167/debates-por-la-historia.v9i2.792>
- Top Node.js Backend Frameworks for 2024. (2024, junio 26). DEV Community. <https://dev.to/3a5abi/top-nodejs-backend-frameworks-for-2024-4f57>
- vardhan, S. (2024, julio 19). Benchmarking Django vs Spring Boot: A comparative study. CodeX. <https://medium.com/codex/benchmarking-django-vs-spring-boot-a-comparative-study-e233dcb047c2>