

## Herramientas eficientes de desarrollo de software más utilizados en Ecuador: caso de estudio aplicado al Club de Robótica “BOT’S UTC”

Efficient tools for software development more used in Ecuador: case of study applied to the Robotics Club: “BOT’S UTC”

Estefania Mishell Guanoluisa Tonato<sup>1</sup>[0009-0005-0476-0941], Jhomara Ibeth Párraga Barriga<sup>2</sup>[0009-0007-0548-0911], José Augusto Cadena Moreano<sup>3</sup>[0000-0002-1775-252X]

<sup>1,2,3</sup>Universidad Técnica de Cotopaxi – Ciencias de la Ingeniería y aplicadas– Sistemas de Información, 050102, Latacunga, Cotopaxi– Ecuador

<sup>1</sup>estefania.guanoluisa2880@utc.edu.ec, <sup>2</sup>jhomara.parraga1308@utc.edu.ec, <sup>3</sup>jose.cadena@utc.edu.ec

### CITA EN APA:

Guanoluisa Tonato, E., Jhomara, P., & Cadena, J. (2024). Herramientas eficientes de desarrollo de software más utilizados en Ecuador: caso de estudio aplicado al Club de Robótica “BOT’S UTC”. Technology Rain Journal, 3(1).  
<https://doi.org/10.55204/trj.v3i1.e24>

**Recibido:** 26 de noviembre de 2023

**Aceptado:** 9 de enero de 2024

**Publicado:** 16 de enero de 2024

Technology Rain Journal  
ISSN: 2953-464X

**Resumen.** El presente trabajo analiza e identifica las metodologías, IDEs, arquitecturas, gestores de BDD y lenguajes de programación predominantes en el desarrollo de software en Ecuador. A través de una investigación realizada de forma autónoma basándose en artículos vigentes relacionados con el tema de la investigación y los resultados de encuestas aplicadas a desarrolladores en el territorio nacional para obtener una visión más clara y realista de los aspectos que interesan conocer. En conjunto con un prototipo desarrollado, haciendo uso de las prácticas y herramientas que resultaron como las más empleadas por los desarrolladores siendo estas, la metodología SCRUM, arquitectura MVC, el IDE Visual Studio Code, el gestor de base de datos MySQL y el lenguaje de programación PHP, se demuestra también la fiabilidad de estas en el desarrollo de software, aplicándolas al caso de estudio del Club de Robótica de la Universidad Técnica de Cotopaxi “BOT’S UTC”.

**Palabras Clave:** Metodologías, Arquitectura, IDE, MVC, BDD.

**Abstract:** This paper analyses and identifies the methodologies, IDEs, architectures, DB engines and programming languages predominant in software development in Ecuador. Through an autonomous research based on current articles related to the research topic and the results of surveys applied to developers in the national territory to obtain a clearer and more realistic view of the aspects that interest us to know. In conjunction with a prototype developed, making use of the practices and tools that were found to be the most used by developers being these, the SCRUM methodology, MVC architecture, the IDE Visual Studio Code, the MySQL database manager and PHP programming language, also demonstrates their reliability in software development, applying them to the case study of the Robotics Club of the Technical University of Cotopaxi “BOT’S UTC”.

**Keywords:** Methodologies, Architecture, IDE, MVC, DB.



Los contenidos de este artículo están bajo una licencia de Creative Commons Attribution 4.0 International (CC BY 4.0)

Los autores conservan los derechos morales y patrimoniales de sus obras.

## 1. INTRODUCCIÓN

Todos los proyectos de software requieren de diferentes aspectos para su desarrollo, incluye las metodologías, arquitecturas, lenguajes de programación, IDEs, gestores de bases de datos, la gran pregunta a esta interrogante planteada es: ¿Cuáles son las herramientas más utilizadas por diferentes empresas en el Ecuador?

El desarrollo de software es un proceso complejo que requiere la participación de diferentes disciplinas y tecnologías. Para que un proyecto de desarrollo de software sea exitoso, es necesario tener en cuenta una serie de aspectos, entre los que se encuentran:

Metodologías de desarrollo de software: estas metodologías proporcionan un marco para planificar, diseñar, desarrollar, probar y desplegar software. Algunas de las metodologías de desarrollo de software más populares son las metodologías ágiles, como Scrum y Kanban, y las metodologías tradicionales, como el modelo en cascada.

Según un estudio de la Universidad de los Andes (2022), las metodologías ágiles son cada vez más populares en el desarrollo de software (García, 2022). Estas metodologías se caracterizan por su flexibilidad y su capacidad de adaptarse a los cambios.

Arquitecturas de software: definen la estructura y la organización de un sistema de software. Existen diferentes tipos de arquitecturas, como las arquitecturas monolíticas, las arquitecturas distribuidas y las arquitecturas microservicios.

Las arquitecturas distribuidas son cada vez más comunes en los sistemas de software modernos (Larrondo, 2022). Estas arquitecturas permiten que los sistemas de software sean más escalables y tolerantes a fallos.

Lenguajes de programación: son las herramientas que utilizan los desarrolladores para escribir código. Existen diferentes tipos de lenguajes de programación, cada uno con sus propias ventajas y desventajas. Los lenguajes de programación más populares en el desarrollo de software son Java, Python y JavaScript (Stack Overflow, 2022). Estos lenguajes son versátiles y se pueden utilizar para desarrollar una amplia gama de aplicaciones.

Los gestores de bases de datos son sistemas que almacenan y administran datos. Existen diferentes tipos de gestores de bases de datos, como relacionales y no relacionales. Los relacionales son los más populares en el desarrollo de software, por otro lado, las bases de datos no relacionales o NoSQL (Not Only SQL) han ganado popularidad en aplicaciones web, redes sociales, análisis de big data y otros casos de uso donde la escalabilidad y la flexibilidad son fundamentales (DB-Engines, 2023).

En cuanto a los entornos de desarrollo, son herramientas que ayudan a los desarrolladores a crear y mantener software (JetBrains, 2023). Existen diferentes tipos de entornos de desarrollo, como los entornos de desarrollo integrados (IDE) y los entornos de desarrollo web.

En este artículo, se conocerán las posibles metodologías utilizadas por las empresas ecuatorianas dedicadas al desarrollo de software mediante la extracción de datos adquiridos a partir de una encuesta realizada a dichas empresas. Los datos fueron adquiridos mediante una encuesta realizada a diferentes empresas del país. Con el resultado, se pretende evaluar la utilidad las herramientas más utilizadas en Ecuador.

Nuestro propósito es identificar la combinación óptima de estos aspectos para optimizar el tiempo y recursos en el desarrollo de software en el Ecuador.

## **2. METODOLOGÍA**

### **2.1 Revisión Bibliográfica**

En el Ecuador, existe una amplia variedad de metodologías de desarrollo, lenguajes de programación y arquitecturas de información que se utilizan en el desarrollo de software. Sin embargo, no existe una combinación única que sea óptima para todos los proyectos.

Hace varios años se han propuesto métodos para el desarrollo de software y se han realizado múltiples estudios sobre cómo agilizar el desarrollo de software en todo el mundo, ya que, esta tarea lleva demasiado tiempo y recursos para optimizar su desarrollo. Un estudio realizado por la Universidad de los Andes en 2022 encontró que las empresas desarrolladoras de software en Colombia están adoptando cada vez más metodologías ágiles (García, 2022).

Existen diferentes puntos de vista de las metodologías y herramientas para el desarrollo de software una de ellas es la dificultad para decidir cuál de ellas es perfecta para completar con éxito y beneficiarse de las características que aporta dichas metodologías, arquitecturas, herramientas y demás, gracias al estudio desarrollado por la Pontificia Universidad Católica del Ecuador Ambato indica que Kanban es una metodología que cumple con los criterios de usabilidad de acuerdo con (Garcés Freire, Pailiacho Mena, & Balseca Manzano, 2018).

Para ello, hemos realizado una revisión bibliográfica, para definir cuál de las metodologías, lenguajes de programación, motores de bases de datos, IDEs y arquitecturas de software son los más utilizados do con las teorías de varios autores.

A continuación, se presenta cada aspecto con una tabla comparativa referenciada.

### **2.1.1 Metodologías para el desarrollo de software**

Scrum es un marco de trabajo ágil que se centra en la entrega de productos de alta calidad de forma rápida y flexible. El trabajo se divide en ciclos cortos de tiempo (Sprint) de dos a cuatro semanas, y al final de cada sprint se entrega un producto funcional. Los equipos de Scrum están formados por personas de diferentes disciplinas que trabajan juntas para lograr un objetivo común. Scrum está diseñado para adaptarse a los cambios en los requisitos o las prioridades (Schwaber & Sutherland, 2022).

Para (Rivas, Corona, Gutiérrez, & Hernández, 2021), las metodologías de software son importantes ya que gracias a ello los requisitos de los usuarios pueden ser resueltos. Permitiendo seguir un flujo de trabajo de manera organizada y precisa.

Según (Yasvi & Yadav, 2019), las metodologías ágiles son las más factibles para el desarrollo rápido, reduciendo costos, adaptándose a los cambios que puedan presentarse a medida que se desarrolla el proyecto. También menciona que dentro de estas metodologías ágiles están SCRUM, XP, Kanban.

XP (Programación Extrema), es un marco del modelo de desarrollo ágil de software que se centra en la producción de software de mayor calidad a través de la planificación adaptativa, el desarrollo evolutivo, la mejora continua y la respuesta rápida y flexible a los cambios. Destaca la satisfacción del cliente, la simplicidad y la atención continua tanto de los desarrolladores como de los clientes. XP engloba todas estas prácticas para lograr sus objetivos (Barros & Pereira Oliveira, 2021).

Según (Cockburn, 2020), la metodología Crystal es un marco de trabajo ágil que se centra en las personas, los procesos y los productos. Está diseñado para ser utilizado en proyectos pequeños y medianos, y se adapta a las necesidades específicas de cada proyecto. Cada nivel de Crystal proporciona un conjunto de prácticas y recomendaciones que se adaptan al tamaño y la complejidad del proyecto.

Para (Schwaber & Sutherland, 2022), la metodología Kanban es un marco de trabajo ágil que se centra en la mejora continua del flujo de trabajo. Se basa en la visualización de las tareas en un tablero Kanban, que ayuda a los equipos a ver lo que se está trabajando, lo que está pendiente y lo que se ha

completado. La metodología Kanban puede ser utilizada en cualquier tipo de proyecto, independientemente de su tamaño o complejidad, ver Tabla 1.

**Tabla 1. Metodologías para el desarrollo de software.**

| Metodología | Ventajas                                                                                                                                                                                                                                                                                                                                                           | Desventajas                                                                                                                                                                                                                                                                |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SCRUM       | <ul style="list-style-type: none"> <li>-Mejora la comunicación y la colaboración entre los equipos.</li> <li>-Aumenta la flexibilidad y la capacidad de adaptación a los cambios.</li> <li>-Mejora la calidad del producto.</li> <li>-Agile Software Development.</li> <li>-Reduce el riesgo de fracaso del proyecto (Schwaber &amp; Sutherland, 2022).</li> </ul> | <ul style="list-style-type: none"> <li>-Scrum también requiere un alto nivel de compromiso y puede ser difícil de implementar en entornos que no están acostumbrados a los métodos ágiles (Schwaber &amp; Sutherland, 2022).</li> </ul>                                    |
| XP          | <ul style="list-style-type: none"> <li>- Entrega de software de alta calidad.</li> <li>- Entrega rápida y eficiente.</li> <li>- Adaptable al cambio.</li> <li>- Intercambio de la comunicación y la colaboración (Barros &amp; Pereira Oliveira, 2021).</li> </ul>                                                                                                 | <ul style="list-style-type: none"> <li>-Implementación y gestión complicada.</li> <li>-Requiere varios recursos y puede resultar costoso.</li> <li>-Difícil de adaptarse a proyectos grandes (Barros &amp; Pereira Oliveira, 2021).</li> </ul>                             |
| CRYSTAL     | <ul style="list-style-type: none"> <li>-Es flexible y se adapta a las necesidades específicas de cada proyecto.</li> <li>-Fomenta la colaboración en equipo.</li> <li>-Permite adaptarse a los cambios durante el desarrollo (Cockburn, 2020).</li> </ul>                                                                                                          | <ul style="list-style-type: none"> <li>-Puede ser compleja de implementar y gestionar.</li> <li>-Requiere un alto nivel de compromiso de los miembros del equipo.</li> <li>-Puede no ser adecuada para proyectos de gran tamaño o complejidad (Cockburn, 2020).</li> </ul> |
| KANBAN      | <ul style="list-style-type: none"> <li>-Se centra en la mejora continua del flujo de trabajo.</li> <li>-Es visual y fácil de entender.</li> <li>-Es flexible y se puede adaptar a cualquier tipo de proyecto (Schwaber &amp; Sutherland, 2022).</li> </ul>                                                                                                         | <ul style="list-style-type: none"> <li>-Puede ser difícil de implementar en proyectos con requisitos muy estrictos.</li> <li>-Requiere un alto nivel de disciplina y compromiso de los miembros del equipo (Schwaber &amp; Sutherland, 2022).</li> </ul>                   |

### 2.1.2 Arquitecturas de software

La arquitectura de software es el diseño estructural de un sistema de software que define los componentes o módulos del sistema, sus relaciones, principios y pautas que guían su organización y evolución. Incluye decisiones clave sobre la división de responsabilidades, la distribución de tareas, la gestión de la complejidad y la interoperabilidad entre los diferentes módulos o capas del sistema.

De acuerdo con (Santiago Gonzáles, y otros, 2021), el patrón de arquitectura MVC se destaca por su capacidad para la escalabilidad además de una inherente facilidad para la implementación de módulos dentro de un aplicativo sin perder la facilidad para encontrar y corregir errores, lo cual es una ventaja relevante en el desarrollo de software.

SOA (Arquitectura Orientada a Servicios), es un enfoque arquitectónico para el diseño de sistemas de software donde los componentes del software se organizan y comunican entre sí como servicios. En la investigación centrada en esta arquitectura se destaca la integración de servicios, concluyendo que la implementación fue adecuada, destacando la eficacia de SOA en este proyecto. Además, la evaluación del rendimiento mostró un éxito del 100% en 384 peticiones al sistema,

subrayando la eficiencia de SOA en la gestión de reservas (Bustamante Granda & Palmay Vargas, 2022).

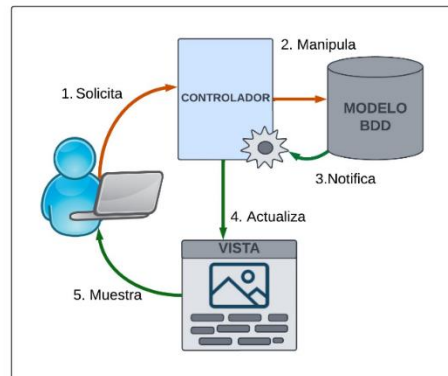


Fig 1. Modelo Vista Controlador.

El propósito del patrón MVC (Fig 1), es separar la lógica de la aplicación en tres componentes principales para facilitar la gestión y el mantenimiento del software.

**Modelo:** Maneja datos y lógica de negocios.

**Vista:** Se encarga del diseño y presentación.

**Controlador:** Enruta comandos a los modelos y vistas.

A continuación, se presentan algunas ventajas y desventajas de las arquitecturas de software, Tabla 2.

Tabla 2. Arquitectura MVC vs SOA

|     | Ventajas                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Desventajas                                                                                                                                                                                                                                                                                                                                      |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MVC | <ul style="list-style-type: none"> <li>-La lógica de negocios y la interacción con la base de datos se encuentran separadas y poseen sus funciones bien definidas.</li> <li>-La Interfaz Gráfica de Usuario (GUI) permite la interacción entre el usuario y el software, evitando que se acceda directamente a la base de datos.</li> <li>-El Controlador de acciones sirve de puente entre el usuario y la base de datos, manteniendo más seguridad y mejor abstracción para el desarrollo y funcionamiento.</li> <li>-Separación de las tareas de aplicación</li> <li>-Capacidad de prueba</li> <li>-Desarrollo controlado por pruebas</li> <li>-Marco extensible y conectable.</li> <li>-Se solicitan datos desde el apartado de controladores, para luego ser manejados por los mismos y luego enviados hacia los modelos que los conectarán con la base de datos (Santiago Gonzáles, y otros, 2021).</li> </ul> | <ul style="list-style-type: none"> <li>-Distintas vulnerabilidades explotables desde el apartado del frontend o vistas.</li> <li>-De ser el caso de no existir el conocimiento y las prácticas necesarias aplicadas al desarrollo, se generan más problemas de mantenimiento del código y depuración (Singh, 2020).</li> </ul>                   |
| SOA | <ul style="list-style-type: none"> <li>-Cada servicio es desarrollado de forma modular, lo que permite su integración e interoperabilidad dentro de un sistema.</li> <li>-De fácil escalamiento y mantenimiento.</li> <li>-Cada servicio puede ser integrado de forma individual sin afectar al funcionamiento general del sistema que lo contiene (Bustamante Granda &amp; Palmay Vargas, 2022).</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | <ul style="list-style-type: none"> <li>-Se necesita estandarizar el desarrollo y presentación de interfaces a fin de generar una integración uniforme y sin contratiempos.</li> <li>-La seguridad y rendimiento dependen completamente de la adopción de prácticas estandarizadas y administración eficiente de la interoperabilidad.</li> </ul> |

### 2.1.3 Entorno de Desarrollo Integrado

Se trata de una aplicación que reúne en un solo lugar todas las herramientas necesarias para llevar a cabo un proyecto de desarrollo de software. El IDE no es sólo un editor de código, sino que combina la funcionalidad de varios procesos de programación en un solo lugar. Como mínimo, tiene capacidades de editor, compilador, desbloqueador, completado de código o gestión general del código.

Según (Bellman, 2018), Visual Studio Code es un editor ligero y altamente personalizable que admite una variedad de lenguajes de programación y tecnologías. Es un lenguaje de programación de lanzamiento creativo que se puede utilizar para editar, depurar y compilar su código y, en última instancia, publicar su aplicación.

Para (Sacndeva & Nafim, 2022), Android Studio se define como un entorno de desarrollo integrado (IDE) creado específicamente para el desarrollo de aplicaciones para el sistema operativo Android. Proporciona un conjunto completo de herramientas para la creación de aplicaciones Android, incluyendo diseño de interfaz, codificación, depuración, prueba y empaquetado.

**Tabla 3. Entornos de Desarrollo Integrado.**

| IDEs               | Ventajas                                                                                                                                                                                                                                                                                                                                                                                                                            | Desventajas                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Visual Studio Code | <ul style="list-style-type: none"> <li>-Ligero y con buen rendimiento en ordenadores poco potentes.</li> <li>-Amplia gama de extensiones disponibles para instalarse de forma sencilla.</li> <li>-Multiplataforma y posee integración con Git (Bellman, 2018).</li> </ul>                                                                                                                                                           | <ul style="list-style-type: none"> <li>-Demasiado sencillo y simple para desarrolladores avanzados.</li> <li>-Por defecto, viene con pocas características para desarrollar software y depende mucho del usuario para ser configurado (Bellman, 2018).</li> </ul>                                                                                                                                    |
| Android Studio     | <ul style="list-style-type: none"> <li>-Entorno Especializado</li> <li>-Integración con Google Services.</li> <li>-Actualizaciones Constantes.</li> <li>-Incluye un emulador de Android para probar aplicaciones en diferentes dispositivos virtuales sin necesidad de hardware físico.</li> <li>-Ofrece potentes herramientas de desarrollo, como Editor de Diseño Gráfico, el depurador, y herramientas de desarrollo.</li> </ul> | <ul style="list-style-type: none"> <li>- Requiere de hardware potente.</li> <li>-Lleva tiempo aprender y acostumbrarse de la amplia gama de funciones que posee.</li> <li>-Puede consumir una cantidad significativa de recursos del sistema.</li> <li>-La instalación inicial puede requerir la descarga de grandes cantidades de datos y archivos.</li> </ul> <p>(Sacndeva &amp; Nafim, 2022).</p> |

### 2.1.4 Lenguajes de programación

Permiten a los programadores comunicarse con las computadoras y especificar cómo se deben realizar ciertas tareas. Cada lenguaje de programación tiene su propia sintaxis y semántica, y algunos están diseñados para propósitos específicos. En la Tabla 4, de acuerdo con el sitio (W3Techs, 2023), se presenta el top de los 6 lenguajes de programación utilizados en los últimos años.

Para (Villacres Tadazo, 2022), Java es un lenguaje de programación que ofrece muchos beneficios para sus usuarios debido a la capacidad innata que tiene para ser multiplataforma además de que los desarrolladores pueden aprenderlo e implementarlo en sus proyectos gracias a sus numerosas bibliotecas.

Según (Muqoribin & Rozaq Rais, 2022), manifiesta que PHP ofrece conveniencia y ventajas que ahorran tiempo durante el proceso de desarrollo. Esto hace que el desarrollo de aplicaciones de gran tamaño sea más fácil y rápido. Por otro lado, también ayuda con el desarrollo de aplicativos webs gracias a su buena interacción e implementación con servidores web y proveedores de hosting de sitios web.

De acuerdo con (Germain Ramírez, 2020), JavaScript es un lenguaje bastante versátil para el desarrollo de software, principalmente sitios web, por su capacidad para ejecutar código en navegadores web y gracias a los frameworks y librerías existentes se puede codificar para que se haga uso en la parte del servidor.

El lenguaje de programación Python es uno de los lenguajes más versátiles de igual forma gracias a su soporte hacia distintos tipos de paradigmas de programación, su tipado débil y la capacidad que tiene para adaptarse a proyectos de toda índole, pudiendo ser usado en aplicaciones de escritorio, web o inteligencia artificial. Para (Rawat, 2020) todas estas características lo convierten en un excelente lenguaje de programación, ver Tabla 5.

**Tabla 4.** Top 6 lenguajes de programación más utilizados para desarrollo de aplicaciones web en Ecuador.

| Lenguaje de Programación | Porcentaje |
|--------------------------|------------|
| PHP                      | 77.7%      |
| ASP.NET                  | 7.4%       |
| Ruby                     | 5.3%       |
| Java                     | 4.6%       |
| JavaScript               | 2.3%       |
| Scala                    | 2.8%       |

Nota: Datos obtenidos del sitio W3Techs.

**Tabla 5. Lenguajes de programación.**

| Lenguajes de Programación | Ventajas                                                                                                                                                                                                                                                                                                                                                                                                | Desventajas                                                                                                                                                                                                                                                                                                   |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Java                      | <ul style="list-style-type: none"> <li>-Portabilidad con máquinas virtuales de java.</li> <li>-Paradigma POO.</li> <li>-Modularidad, mantenimiento y reutilización de código.</li> <li>-Seguridad a través de la máquina virtual de java.</li> <li>-Gestiona automáticamente la asignación y liberación de memoria.</li> <li>- Amplia biblioteca de APIs disponible (Villacres Tadao, 2022).</li> </ul> | <ul style="list-style-type: none"> <li>-Rendimiento inferior en tareas de ejecución rápida.</li> <li>-Mayor consumo de recursos y memoria.</li> <li>-Aplicaciones de escritorio con arranque lento.</li> <li>-Código más extenso y sintaxis más verbosa.</li> <li>-Fallos en seguridad históricos.</li> </ul> |
| PHP                       | <ul style="list-style-type: none"> <li>- Fácil aprendizaje y uso.</li> <li>- Amplia compatibilidad con servidores web y sistemas operativos.</li> <li>- Excelente integración con base de datos.</li> <li>- Amplia documentación.</li> <li>- Frameworks basados en el lenguaje para desarrollo web rápido (Muqoribin &amp; Rozaq Rais, 2022).</li> </ul>                                                | <ul style="list-style-type: none"> <li>- Errores difíciles de detectar y depurar.</li> <li>-Vulnerabilidades de seguridad históricas.</li> <li>-Escaso soporte para programación asíncrona.</li> <li>-Poco rendimiento en aplicaciones de procesamiento rápido.</li> </ul>                                    |
| Java Script               | <ul style="list-style-type: none"> <li>-Se ejecuta nativamente en los navegadores.</li> <li>-Versatilidad para usarse tanto en el lado del cliente como en un servidor.</li> <li>-Gran variedad en frameworks y librerías para el desarrollo.</li> <li>-Puede realizar solicitudes asíncronas e interactúa con APIs (Germain Ramírez, 2020).</li> </ul>                                                 | <ul style="list-style-type: none"> <li>-Vulnerabilidades de seguridad a inyección de código.</li> <li>-Compatibilidad entre navegadores inconsistentes.</li> <li>-Mayor exigencia de recursos en dispositivos antiguos.</li> <li>-Difícil mantenimiento de código.</li> </ul>                                 |
| Python                    | <ul style="list-style-type: none"> <li>-Sintaxis clara, legible y de fácil comprensión.</li> <li>-Amplia variedad de bibliotecas y frameworks disponibles.</li> <li>-Soporta distintos paradigmas de programación (Rawat, 2020).</li> <li>-Alta versatilidad en desarrollo y aplicaciones.</li> </ul>                                                                                                   | <ul style="list-style-type: none"> <li>-Rendimiento inferior en tareas de rápida ejecución.</li> <li>-Gestión de memoria ineficiente.</li> <li>-Rendimiento bloqueado para su ejecución en múltiples hilos, dando pésimo rendimiento en tareas multinúcleo.</li> </ul>                                        |

A continuación, se muestra en la Fig 2, un indicador que demuestra que el lenguaje de programación más utilizado es PHP (W3Techs, 2023).

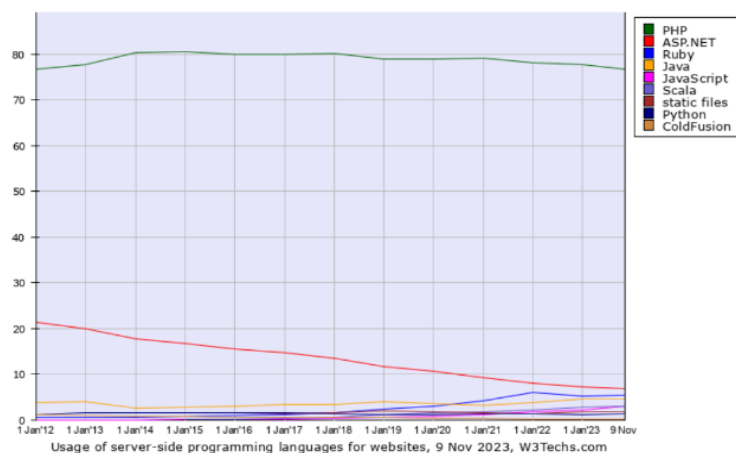


Fig 2. Indicador de lenguajes de programación para desarrollo de aplicaciones web (W3Techs, 2023).

### 2.1.5 Gestores de base de datos

Según (Robledano, 2019), manifiesta que mongoDB, es un sistema de gestión de bases de datos NoSQL de código abierto, desarrollado en C++ y centrado en documentos. En lugar de almacenar la información en tablas, utiliza estructuras de datos similares a JSON con un esquema flexible.

Según (Kinsta, 2023), PostgreSQL es un sistema de gestión de bases de datos de código abierto, altamente confiable, que ofrece compatibilidad con diversas funciones de SQL, incluyendo claves foráneas, subconsultas, disparadores, así como la capacidad de definir diferentes tipos y funciones de usuario.

Para (Illic, Kopanja, Zlatkovic, & Trajkovic, 2021), Microsoft SQL Server es un motor que está desarrollado con la implementación en un ambiente que use software propietario de la compañía Microsoft en gran medida, ayudando con las tareas de almacenamiento y gestión de datos.

Según (Wahyudi, Asbari, Sasono, Pramono, & Novitasari, 2020), MySQL es un sistema de gestión de bases de datos relacionales (RDBMS) gratuito que puede ser utilizado por cualquier persona bajo la Licencia Pública General (GPL), excepto para productos derivados comerciales. Se basa en el concepto de SQL (lenguaje de consulta estructurado), que se utiliza para operaciones de bases de datos, como seleccionar e ingresar datos. SQL permite operaciones de datos fáciles y automatizadas.

De acuerdo con (JESUITES educació Formación Profesional, 2019), Oracle es una herramienta de gestión de bases de datos, especialmente adoptada por grandes empresas para controlar y administrar extensos contenidos desde una única fuente.

Este enfoque no solo reduce costos, sino que también mitiga el riesgo de pérdida de información. Entre las funciones clave de Oracle se incluyen el soporte de transacciones, estabilidad y compatibilidad multiplataforma con Windows, Linux, Mac OS X, Unix y BDS, ver Tabla 6.

**Tabla 6. Gestores de Bases de Datos**

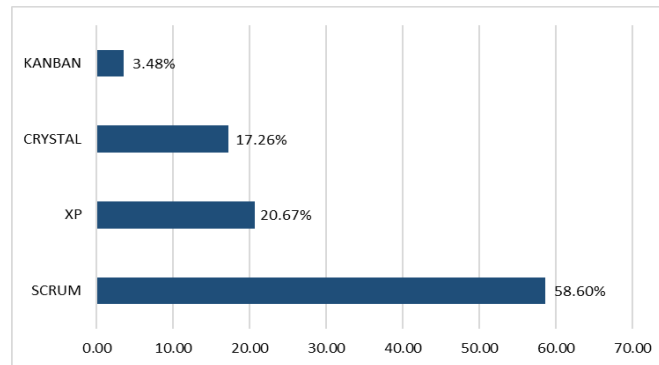
| <b>Motor de Base de Datos</b> | <b>Ventajas</b>                                                                                                                                                               | <b>Desventajas</b>                                                                                         |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| MongoDB                       | -Esquema flexible<br>-Escalabilidad horizontal<br>-Rendimiento (Robledano, 2019).                                                                                             | -Falta de soporte por defecto<br>-Consumo de recursos<br>-Complejidad en consultas complejas.              |
| PostgreSQL                    | -Funcionalidades avanzadas.<br>-Extensibilidad.<br>-Integridad de datos (Kinsta, 2023).                                                                                       | -Consumo de recursos<br>-Curva de aprendizaje<br>-Menor popularidad en algunas áreas.                      |
| Microsoft SQL Server          | -Integración con entorno Microsoft.<br>-Herramientas de administración y monitorización<br>-Servidores de informes y análisis (Illic, Kopanja, Zlatkovic, & Trajkovic, 2021). | -Costo de licencia<br>-Limitaciones en sistemas No-Windows<br>-Menor flexibilidad en la escala horizontal. |
| MySQL                         | -Velocidad y rendimiento.<br>-Escalabilidad.<br>-Amplia adopción y comunidad activa (Wahyudi, Asbari, Sasono, Pramono, & Novitasari, 2020).                                   | -Limitaciones en funciones avanzadas<br>-Gestión de transacciones<br>-Desarrollo controlado por Oracle.    |
| Oracle                        | -Alta disponibilidad y confiabilidad<br>-Seguridad Avanzada<br>-Escalabilidad vertical y horizontal (JESUITES educació Formación Profesional, 2019).                          | -Costo<br>-Complejidad de recursos<br>-Licenciamiento complejo.                                            |

## 2.2 Análisis de resultados de la encuesta aplicada

Para la encuesta, se seleccionaron empresas que desarrollan software en Ecuador. La selección se realizó de forma aleatoria, utilizando una lista de empresas, proporcionada por la Cámara de Comercio de Quito. Se contactó a las empresas seleccionadas por correo electrónico y se les invitó a participar en la encuesta.

Para ello, se seleccionaron las preguntas que se han considerado fundamentales. A continuación, se presenta cada una de ellas, con su respectivo indicador y análisis:

### 2.2.1 Pregunta ¿Qué tipo de metodología ágil sería su preferida para el desarrollo de un proyecto de software?

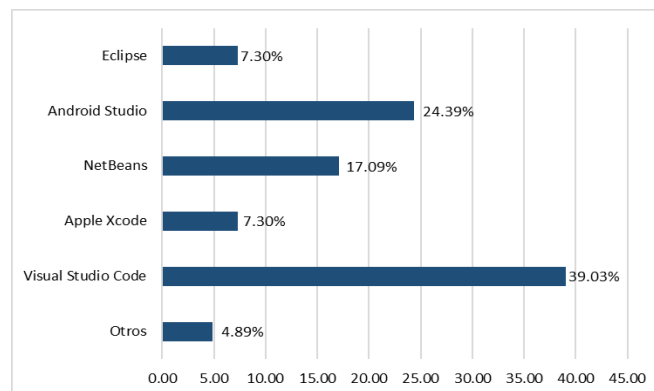


*Fig 3. Metodologías de desarrollo de software.*

Se obtuvieron los siguientes datos, como se puede apreciar en la Fig 3, la mayoría de encuestados optan por la metodología SCRUM con 58.60% de aceptación para la elaboración de sus proyectos, cabe recalcar que también existe un 20.67% de empresas que utilizan la metodología XP.

Además, el 17.26% de empresas utilizan la metodología CRYSTAL y un 3.48% de empresas eligieron la opción KANBAN. Con la recopilación de estos datos se afirma que la metodología más utilizada en Ecuador es la SCRUM.

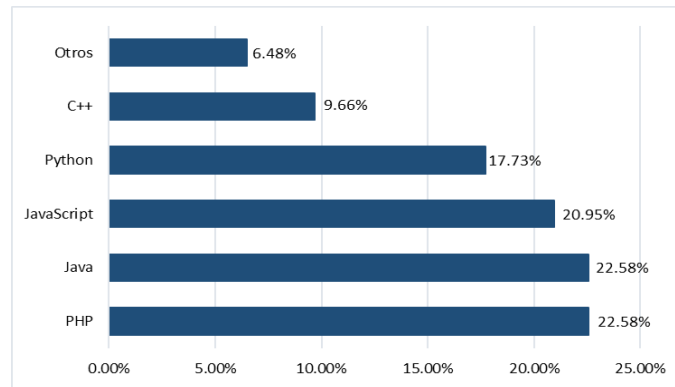
### **2.2.2 Pregunta ¿Qué entorno de desarrollo integrado utiliza en su empresa?**



*Fig 4. Entorno de Desarrollo Integrado.*

De acuerdo con la Fig. 4, los datos recopilados en la encuesta muestran que el 39.03% de empresas seleccionaron la opción Visual Studio Code, dejando en evidencia que es la más utilizada. Por otro lado, los encuestados optan por utilizar Android Studio con 24.39% de aceptación. Además, el 17.09% utiliza NetBeans mientras que el 7.30% utilizan Eclipse al igual que Apple Xcode. Por otro lado, el 4.89% manifiestan que utilizan otros IDEs para el desarrollo de proyectos de software.

### **2.2.3 Pregunta ¿Qué lenguaje de programación utiliza en su empresa?**

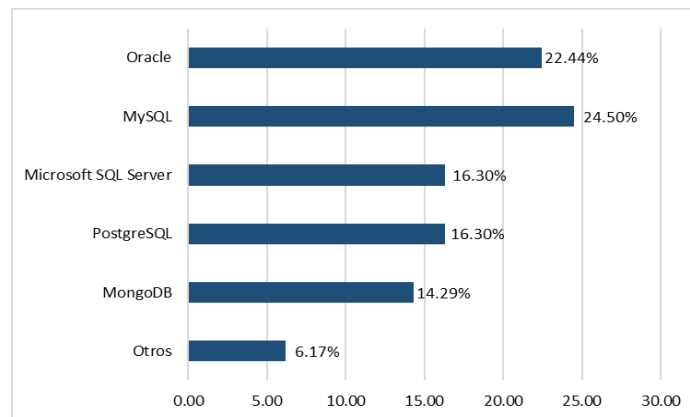


*Fig 5. Lenguajes de programación.*

De acuerdo con los resultados obtenidos en esta pregunta, los lenguajes de programación más utilizados en el desarrollo de software son PHP con un 22.58%, y Java con 22.58% del total de respuestas recibidas, como se muestra en la Fig 5.

Por otro lado, en tercer lugar, tenemos a JavaScript con 20.95%, a Python con 17.73%, a C++ con 9.66% de aceptación y el 6.48% de empresas que utilizan otros lenguajes. En general las empresas encuestadas usan una gran variedad de lenguajes de programación entre los que se destacan PHP y Java.

#### **2.2.4 Pregunta ¿Qué gestor de base de datos utiliza en la empresa?**



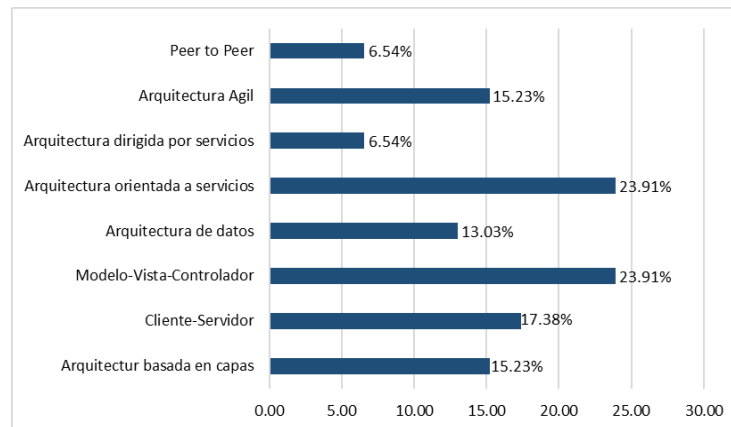
*Fig 6. Motores de base de datos.*

De acuerdo con el resultado de las encuestas Fig 6, se obtuvo que la mayoría de ellas utiliza MySQL con 24.50% de aceptación. Sin embargo, un 22.44% de empresas optan por utilizar Oracle para gestionar sus datos. Por otro lado, tenemos a Microsoft SQL Server con 16.30% al igual que PostgreSQL.

No obstante, también existe un 14.29% de empresas que utilizan MongoDB y 6.17% de empresas que utilizan otros motores de bases de datos.

De las empresas encuestadas, se obtuvo como resultado que el motor de base de datos elegido por la mayoría de los desarrolladores es MySQL.

### 2.2.5 Teniendo en cuenta la importancia de generar o implementar una arquitectura de software para la creación de un proyecto, indique, ¿Qué tipo de arquitectura es la más usada para desarrollar dicho proceso?



*Fig 7. Arquitecturas de software.*

En la Fig 7, se pudo identificar que la SOA (Arquitectura Orientada a Servicios), es la más utilizada con 23.91%, al igual que la arquitectura MVC(Modelo-Vista-Controlador). En cambio, el 17.38% opta por la arquitectura Cliente-Servidor.

Por otro lado, la Arquitectura basada en capas y Arquitectura ágil tienen un 15.23% de aceptación. Además, para la Arquitectura de datos existe un porcentaje del 13.03% de empresas que indican utilizar este tipo de arquitectura. Se evidencia índices bajos en el uso de otras arquitecturas de software a implementar como es la Arquitectura dirigida por servicios (EDA) y Peer to peer con un 6.54%.

De acuerdo con estos resultados las empresas optan por la implantación de una SOA (Arquitectura Orientada a Servicios) y MVC (Arquitectura Modelo Vista Controlador) dejando a éstas como las más utilizadas.

### 2.3 Diseño de un prototipo

Para el diseño del prototipo, se ha empleado HTML para desplegar los contenidos y CSS para mejorar la vista de cada sección, los resultados alcanzados se evidencian en el prototipo desarrollado, muestra Fig 8. Se seleccionaron los colores, de acuerdo con el logo del club de Robótica.



*Fig 8. Interfaz principal del sistema BOT'S UTC.*

Las herramientas utilizadas dependen del proyecto y de la facilidad del programador, porque existen varios programadores que se adaptan a otros lenguajes, metodologías, arquitecturas, etc.

Con la finalidad de agilizar el avance de su desarrollo y cumplir con los requisitos del usuario este sistema cuenta con dos presentaciones una destinada a la vista del administrador y otra para los usuarios normales.

La vista del Administrador es destinada para realizar el CRUD (Create, Read, Update y Delete) de cada sección antes mencionada. La vista Miembro está destinada para consultas del club de robótica.

Como resultado final del desarrollo del sistema se logró crear un prototipo capaz de satisfacer y solventar las necesidades del Club de Robótica "BOT'S UTC". Se decidió diseñar diferentes módulos dentro del aplicativo cada uno centrado en mejorar el proceso de desarrollo de varias actividades como:

### **2.3.1 Gestión de miembros**

En este apartado se facilita el proceso de gestión de miembros, pues presenta una interfaz gráfica amigable, intuitiva y fácil de utilizar mejorando el cumplimiento de esta tarea, Fig 9.

Esta sección permite realizar un CRUD (Create, Read, Update y Delete) de los datos del usuario. Además, para agilizar el tedioso proceso de búsqueda de miembros se ha implementado un panel de búsqueda en la parte superior, para que el usuario visualice los registros que requiere en el menor tiempo posible.



### MIEMBROS

Nuevo Miembro

Mostrar 10 registros

Buscar:

| ID # | NOMBRE           | APELLIDO          | CÉDULA     | CORREO                         | CELULAR    | ACCIONES |
|------|------------------|-------------------|------------|--------------------------------|------------|----------|
| 1    | STEVEN PAUL      | AREQUIPA POAQUIZA | 0503549883 | steven.arequipa9883@utc.edu.ec | 0978809799 |          |
| 2    | ROMEL STIVEN     | ANTE GUANOCHANGA  | 0504018250 | romel.ante8250@utc.edu.ec      | 0997061495 |          |
| 3    | STALIN BRANDON   | GORDON HIDALGO    | 1725818932 | brandon.gordon8932@utc.edu.ec  | 0983855666 |          |
| 4    | ADRIAN GUILLERMO | GUATO CAIZA       | 0550239131 | adrian.guato9131@utc.edu.ec    | 0984573243 |          |
| 5    | EDUARDO LUCIANO  | NOVILLO VELEZ     | 1724054141 | eduardo.novillo4141@utc.edu.ec | 0939119988 |          |
| 6    | AMILCAR GIORDANO | ORTIZ MOLINA      | 2100625124 | amilcar.ortiz5124@utc.edu.ec   | 0992072764 |          |

Fig 9. Interfaz Miembros del sistema BOT'S UTC

### 2.3.1 Gestión de archivos

Para la gestión de todos los documentos generados por la entidad se ha diseñado un apartado, el cual cuenta con un CRUD, así que, el usuario podrá crear, leer, eliminar y actualizar o editar los documentos subidos a esta sección.

Además, cuenta con un panel de búsqueda para que el usuario pueda explorar mediante la opción añadida simplemente ingresando los parámetros de búsqueda, como se muestra en la Fig 10.

### ARCHIVOS

Nuevo Archivo

Mostrar 10 registros

Buscar:

| ID # | NOMBRE                | DESCRIPCIÓN                         | DOCUMENTO | ACCIONES |
|------|-----------------------|-------------------------------------|-----------|----------|
| 1    | Inventario            | Materiales del Club                 |           |          |
| 2    | Nomina de estudiantes | Registro de información de miembros |           |          |

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior 1 Siguiente

Fig 10. Interfaz Archivos del aplicativo BOT'S UTC.

### 2.3.3. Gestión de activos

Se ha diseñado un apartado dedicado al registro de activos que posee la entidad, mediante la implementación del módulo activos el cual cuenta con lo siguiente:

Cuenta con un CRUD para gestionar los activos del Club "BOT'S UTC". Además, en la parte superior izquierda se presenta un panel de búsqueda, para facilitar la navegación del usuario, al buscar los distintos activos almacenados, Fig 11.

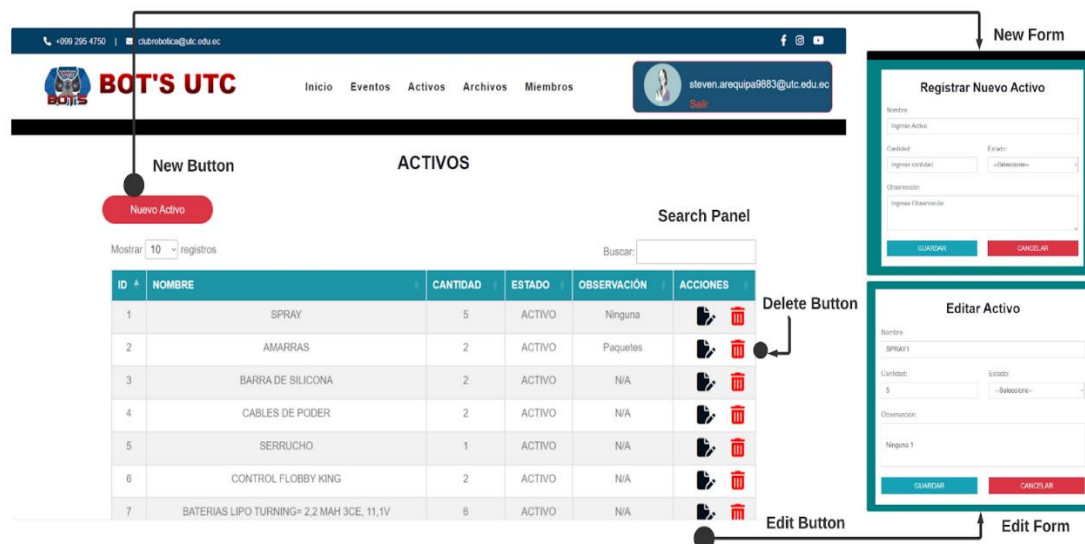


Fig 11. Interfaz Activos del aplicativo BOTS UTC.

### 2.3.4 Gestión de eventos

Cada evento creado se puede visualizar en la página principal de eventos, como se presenta en la Fig 12, cada evento está almacenado en tarjetas y está compuesto por “header-card” y “body-card”, en la parte inferior de cada tarjeta, se incorporó 2 botones (Editar y Eliminar). Cada botón redirige a una vista diferente.

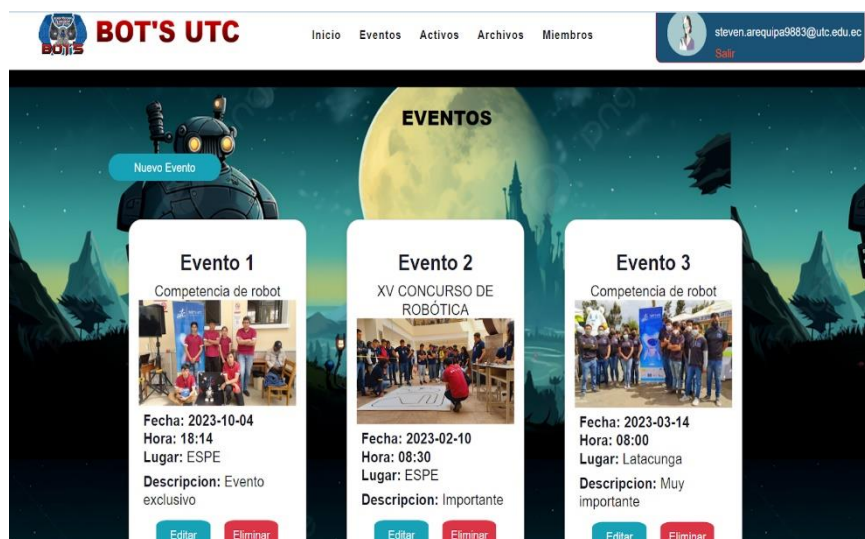


Fig 12. Interfaz eventos del aplicativo BOT'S UTC.

### 2.4 Pruebas

En la Tabla 7, se puede observar los resultados de las pruebas efectuadas de acuerdo con los CP (Casos de Prueba) mostrado en la Fig 13, que se han propuesto para revisar el cumplimiento y satisfacción de los RF (Requisitos Funcionales) que se obtuvieron basados en los requerimientos de los usuarios.

**Tabla 7. Pruebas de requisitos funcionales**

|      | RF-01 | RF-02 | RF-03 | RF-04 | RF-05 | RF-06 | RF... |
|------|-------|-------|-------|-------|-------|-------|-------|
| CP-1 | X     |       | X     |       | X     |       | X     |
| CP-2 |       | X     |       | X     |       | X     |       |
| CP-3 | X     |       | X     |       | X     |       | X     |
| CP-4 | X     | X     | X     | X     | X     | X     | X     |

| Caso de prueba de interfaz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | CP-01                  |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|----|
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | ¿Prueba de despliegue? | No |
| <b>Descripción:</b><br>Se comprueba que la interfaz no presente errores y que se visualice con normalidad.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                        |    |
| <b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• Desarrollar la interfaz del apartado de información correspondiente.</li> <li>• Tener la interfaz integrada dentro del prototipo.</li> <li>• Revisar el prototipo desde un navegador de escritorio.</li> </ul>                                                                                                                                                                                                                                                                                                               |                        |    |
| <b>Pasos:</b> <ol style="list-style-type: none"> <li>1. Abrir el prototipo y entrar en la sección inicio.</li> <li>2. Verificar que no existan errores en la presentación como fuentes, tamaños de letra, colores, maquetado de los elementos de la página o falta de elementos.</li> <li>3. En el caso de que existan formularios que requieran ingresos por teclado de parte del usuario, asegurarse de que estos no permitan valores erróneos o distintos a los solicitados.</li> <li>4. Los enlaces o hipervínculos deben llevar de manera correcta entre páginas del sitio.</li> </ol> |                        |    |
| <b>Resultado esperado:</b><br>Se espera que todos los elementos se muestren de forma correcta, que la página se vea de acuerdo a lo esperado y los botones o vínculos que lleven a otra página, lleven a la página deseada.                                                                                                                                                                                                                                                                                                                                                                 |                        |    |
| <b>Resultado obtenido:</b><br>Los botones y vínculos funcionan de forma correcta, no existen irregularidades en la presentación y todo el texto de la página mantiene coherencia con el resto del sitio y el tema usado.                                                                                                                                                                                                                                                                                                                                                                    |                        |    |

*Fig 13. Ejemplo de Caso de prueba N°1.*

### 3. RESULTADOS Y DISCUSIÓN

La investigación realizada a fin de generar un apoyo bibliográfico por parte de distintas fuentes de conocimiento y autores ha demostrado que la metodología para desarrollo de software dentro de un equipo de trabajo adecuada es SCRUM de acuerdo con (Schwaber & Sutherland, 2022).

Por otro lado, para (Santiago Gonzáles, y otros, 2021), la arquitectura para generar módulos que cumplan con las necesidades propuestas y de fácil corrección es la MVC. Según (Bellman, 2018), el IDE que mejor se adapta al desarrollo de sitios y aplicativo web es Visual Studio Code. Por lo expuesto por (Muqoribin & Rozaq Rais, 2022), el lenguaje de programación que mejor interactúa y facilita el desarrollo de software gracias a su estabilidad y variedad de soluciones mediante bibliotecas y frameworks es PHP. Finalmente, el motor de base de datos con mayor aceptación, documentación, facilidad de aprendizaje, uso e integración a diversos tipos de lenguajes de programación y servidores web es MySQL, según la opinión de (Wahyudi, Asbari, Sasono, Pramono, & Novitasari, 2020).

El análisis de esta investigación es de gran ayuda y complementa la visión que se obtiene al analizar los resultados obtenidos de la encuesta realizada.

Los resultados que las encuestas efectuadas han devuelto son otra base para determinar un resultado claro acerca de las metodologías preferidas por los desarrolladores de software siendo esta SCRUM con un 58.60%, mostradas en la Fig 3. El IDE más usado es Visual Studio Code con un 39.03%, como se observa en la Fig 4.

Además, entre los encuestados, los lenguajes más utilizados empatados con un 22.58% son Java y PHP, mostrados en la Fig 5. Por otro lado, el motor de base de datos más ocupado es MySQL de acuerdo con un 24.50% de las respuestas, como se evidencia en la Fig 6, y las arquitecturas más implementadas son el Modelo-Vista-Controlador (MVC) y la arquitectura orientada a servicios (SOA) empatadas con 23.91% cada una, de acuerdo con la Fig. 7.

Respaldada ahora, por la visión generada por las opiniones de distintos autores y los resultados de las encuestas, es mucho más fácil dar una opinión sobre el desarrollo de software en Ecuador, respecto al conjunto de metodologías, arquitecturas y herramientas usadas.

Y, con el fin de comprobar la eficacia de estos mismos recursos empleados, se ha generado un prototipo de sitio web, orientado a solventar diversas necesidades planteadas por el Club de Robótica “BOT’S UTC”, ocupando para ello la metodología SCRUM, la arquitectura MVC, el IDE Visual Studio Code, el lenguaje de programación PHP y una base de datos generada en MySQL.

Con la finalización del prototipo, se evidencia gracias al plan de pruebas ejecutado, observado en la tabla 7, que, en efecto, todo el conjunto de prácticas y herramientas usadas han probado ser eficaces al permitir un desarrollo de software que cumple con todas las necesidades solicitadas por un cliente.

#### **4. CONCLUSIONES**

La investigación generada presenta diversas opiniones de autores y artículos relacionados con el tema. Las encuestas realizadas generaron una imagen más clara respecto al proceso del desarrollo de software en el Ecuador. Mediante el uso y aplicación de lo investigado se han generado resultados que

reflejan satisfactoriamente qué herramientas y prácticas se emplean mayormente por desarrolladores en el país. La combinación de herramientas y prácticas halladas como las más usadas demuestran ser eficientes y fiables para el desarrollo de software. La eficiencia de estas herramientas se ve reflejada en la reducción de errores y tiempo de desarrollo.

## CONFLICTO DE INTERESES

Nosotros, los autores del artículo titulado “Herramientas eficientes de desarrollo de software más utilizados en Ecuador: caso de estudio aplicado al Club de Robótica “BOT’S UTC””, sobre lenguajes de programación, hacemos constar por la presente que no existe ningún conflicto de intereses en relación con la investigación y los resultados presentados en dicho trabajo. Confirmamos que no tenemos afiliaciones, intereses financieros, ni relaciones personales que puedan influir de manera indebida en la objetividad y la integridad de la investigación realizada.

Reafirmamos nuestro compromiso con la transparencia y la ética en la presentación de nuestro trabajo académico, asegurando a la comunidad científica y a los lectores en general que la investigación se llevó a cabo de manera imparcial y sin influencias externas que pudieran comprometer la validez de los resultados.

## CONTRIBUCIÓN DE AUTORÍA

Los autores declaran sus contribuciones en la siguiente matriz:

|                                                               | Guanoluisa Estefanía | Párraga Jhomara | Cadena José |
|---------------------------------------------------------------|----------------------|-----------------|-------------|
| <b>Participar activamente en:</b>                             |                      |                 |             |
| Conceptualización                                             |                      | X               |             |
| Análisis formal                                               | X                    | X               |             |
| Adquisición de fondos                                         | X                    | X               | X           |
| Investigación                                                 | X                    |                 |             |
| Metodología                                                   | X                    |                 |             |
| Administración del proyecto                                   | X                    |                 |             |
| Recursos                                                      |                      | X               |             |
| Redacción –borrador original                                  |                      | X               |             |
| Redacción –revisión y edición                                 |                      |                 | X           |
| <b>La discusión de los resultados</b>                         |                      |                 | X           |
| <b>Revisión y aprobación de la versión final del trabajo.</b> |                      |                 | X           |

## 5. REFERENCIAS

Barros, F., & Pereira Oliveira, A. (2021). Extreme Programming: A review of the literature. *International Journal of Software Engineering and Knowledge Engineering*, 31(4), 639-670. Obtenido de <https://www.worldscientific.com/worldscinet/ijseke>

- Bellman, C. (2018). Studying Developer Build Issues and Debugger Usage via Timeline Analysis in Visual Studio IDE. *MSR '18: Proc. 15th Int. Conf. Mining Softw. Repositories*.
- Bustamante Granda, W., & Palmay Vargas, B. L. (2022). *Aplicación web para la administración de escenarios deportivos del GADM Riobamba, utilizando arquitectura SOA*. Tesis, Universidad Nacional de Chimborazo, Ingeniería en Sistemas Computacionales, Riobamba. Obtenido de <http://dspace.unach.edu.ec/handle/51000/8757>
- Castillo Yagual, C. A., & Coronel Suárez, M. A. (Junio de 2023). Frameworks PHP basados en la arquitectura Modelo-Vista-Controlador para desarrollo de aplicaciones web. *Revista Científica y Tecnológica UPSE*, X(1), 70-78. doi:10.26423/rctu.v10i1.703
- Cockburn, A. (2020). *Crystal: A modern and practical guide to agile development*. Addison-Wesley.
- DB-Engines. (2023). *DB-Engines Ranking*. (DB-Engines) Obtenido de <https://db-engines.com/en/ranking>
- Garcés Freire, E. X., Pailiacho Mena, V. M., & Balseca Manzano, J. M. (2018). La usabilidad en el desarrollo de software en el Ecuador. *PUCESA*.
- García, A. (2022). *Estudio sobre el desarrollo de software en Colombia*. Universidad de los Andes, Bogotá, Colombia.
- Germain Ramírez, C. (2020). *Desarrollo de aplicaciones web utilizando JavaScript*. Universidad Politécnica de Sinaloa, Mazatlán, Sinaloa.
- Illic, M., Kopanja, L., Zlatkovic, D., & Trajkovic, M. (2021). MICROSOFT SQL SERVER AND ORACLE: COMPARATIVE PERFORMANCE ANALYSIS. *The 7th International conference*, 35-38.
- JESUÏTES educació Formació Profesional. (21 de Octubre de 2019). *JESUÏTES educació Formació Profesional*. Obtenido de ¿Qué es Oracle y por qué es tan importante para las empresas?: <https://fp.uoc.fje.edu/blog/que-es-oracle-por-que-importante-para-empresas/>
- JetBrains. (2023). *IDEs de JetBrains*. (JetBrains) Obtenido de <https://www.jetbrains.com/es-es/idea/>
- Kinsta. (13 de Enero de 2023). *Kinsta*. Obtenido de Kinsta: <https://kinsta.com/es/base-de-conocimiento/que-es-postgresql/>
- Larrondo, M. Á. (2022). Arquitecturas de software: tipos y tendencias. *Revista Iberoamericana de Sistemas, Cibernética e Informática*, XVIII(2), 1-13.
- Muqoribin, M., & Rozaq Rais, M. (2022). Comparison of PHP Programming Language with Codeigniter Framework in Project CRUD. *International Journal of Computer and Information System (IJCIS)*, III(3), 94-98. doi:<https://doi.org/10.29040/ijcis.v3i3.77>
- Rawat, A. (Diciembre de 2020). A Review on Python Programming. *International Journal of Research in Engineering, Science and Management (IJRESM)*, III(12), 8-11.
- Rivas, J., Corona, J. C., Gutiérrez, J., & Hernández, J. (2021). *Metodologías de desarrollo de software*. México: Pearson.
- Robledano, A. (28 de Octubre de 2019). *¿Qué es MongoDB y características?* (OpenWebinars.net) Obtenido de <https://openwebinars.net/blog/que-es-mongodb/>
- Sacndeva, D., & Nafim, R. (2022). *Mastering Android Studio: A Beginner's Guide* (Vol. III). London: CRC Press.

- Santiago Gonzáles, L. F., Huerta Mendoza, J. C., Mendoza Luna, Y., Rodríguez Rodríguez, A. J., Vargas Requena, D. T., Martínez Rodríguez, J. L., & Malacara Navejar, J. G. (30 de Noviembre de 2021). Aplicación Web Basada en el Patrón de Arquitectura de Software Modelo-Vista- Controlador (MVC) para Incrementar el Desempeño Académico en la Asignatura de Matemáticas Básicas. *Tecnología Educativa Revista CONAIC*, VIII(1). doi:<https://doi.org/10.32671/terc.v8i1.187>
- Schwaber, K., & Sutherland, J. (2022). *Scrum: A Pocket Guide*. Obtenido de Scrum.org: <https://scrumguides.org/scrum-guide.html>
- Singh, S. (2020). MVC Framework: A Modern Web Application Development Approach and Working. *International Research Journal of Engineering and Technology (IRJET)*, VII(1), 1-5. doi:10.15680/IJIRJET/2020/v7n01/001
- Stack Overflow. (2022). *The State of Developer Ecosystem 2022*. (Stack Overflow) Obtenido de <https://survey.stackoverflow.co/2022>
- Villacres Tadazo, H. (2022). *Estudio comparativo sobre las herramientas de lenguaje de programación Java y Python en el desarrollo de aplicaciones Android*. Universidad Técnica de Babahoyo, Babahoyo.
- W3Techs. (Noviembre de 2023). *Usage statistics of Ecuador as server location of websites*. (W3Techs) Obtenido de <https://w3techs.com/technologies/details/sl-ec->
- Wahyudi, J., Asbari, M., Sasono, I., Pramono, T., & Novitasari, D. (2020). Database Management Education in MYSQL. *Edumaspul*, 2413-2417.
- Yasvi, M. A., & Yadav, K. S. (2019). Review On Extreme Programming-XP. *International Conference on Robotics, Smart Technology and Electronics Engineering (RSTEE)*. India. Obtenido de <https://www.researchgate.net/publication/332465869>