



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO  
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA  
SOFTWARE

**INTEGRANTES:**

Jheyson Monje	7188
Jonathan Chamorro	7167

**CURSO:**

Octavo "A"

**ASIGNATURA:**

Aplicaciones Informáticas II

**DOCENTE:**

Ing. Julio Santillan

**TEMA:**

Análisis y Elección de Herramientas Para el Desarrollo del Proyecto

**07 octubre 2024 - 28 febrero 20**

## Herramientas para el Desarrollo

La elección de herramientas específicas para el desarrollo de una aplicación debe basarse en su capacidad para cumplir con los requerimientos del proyecto de manera eficiente.

### Lenguajes de Programación y Framework para Backend

Django es un framework web de alto nivel de Python el cual simplifica el desarrollo de aplicaciones web al promover el desarrollo rápido y un diseño limpio y pragmático, uno de los beneficios más notables que ofrece Django es el principio (Don't Repeat Yourself) DRY, que fomenta la reutilización de código y reduce la redundancia, además entre las características que distinguen a Django está la robusta protección contra amenazas comunes como la inyección SQL, el scripting entre sitios y la falsificación de solicitudes entre los mismos, su documentación clara y variedad de bibliotecas hacen de Django una opción popular para desarrolladores web que buscan crear aplicaciones seguras, escalables y fáciles de mantener.

#### Características

- Rapidez de Desarrollo: Django es un framework de alto nivel que incluye características listas para usar como autenticación, gestión de usuarios y manejo de formularios, acelerando el desarrollo.
- Seguridad: Ofrece protección contra ataques comunes como XSS, CSRF y SQL Injection, lo que es crucial para una aplicación que maneja datos sensibles como facturación y contratos.
- Estructura Escalable: Django sigue el patrón Model-View-Template (MVT), separando lógica, datos y vistas, lo que facilita la escalabilidad y el mantenimiento.
- Soporte para APIs: Con Django REST Framework (DRF), puedes construir una API robusta y bien estructurada, lo cual es necesario para conectar el frontend y el backend, así como para permitir la integración con otros sistemas (por ejemplo, notificaciones o aplicaciones móviles).
- Gestión de Tareas Automáticas: Con Celery y Django, puedes implementar tareas en segundo plano, como la activación/desactivación automática de servicios basados en los pagos.

### Estudio comparativo de los lenguajes de programación para backend

Características	Python	Java	JavaScript	PHP	Ruby	GO
Facilidad Aprendizaje	Alta	Media	Alto	Media	Alta	Media
Rendimiento	Medio	Alto	Bueno	Medio	Medio	Muy Alto
Escalabilidad	Bueno	Excelente	Bueno	Adecuado	Adecuado	Excelente

<b>Seguridad</b>	Bueno con librerías	Excelente	Moderado según configuración	Modereado	Bueno con Ruby on Rails	Muy Buena
<b>Concurrencia</b>	Limitado GIL	Excelente	Bueno	Limitado	Limitado	Excelente
<b>Comunidad</b>	Enorme, active	Enorme	Enorme	Grande	Grande	Creciente rápidamente
<b>Paradigma</b>	Multiparadigma	Orientado a objetos	Multiparadigma	Procedural, orientado a objetos	Orientado a objetos	Concurrente, orientado a objeto
<b>Sintaxis</b>	Clara, legible	Robusta	Flexible, dinámica	Similar a C	Elegante, expresiva	Concisa, eficiente
<b>Popularidad</b>	Muy alta, versátil	Muy alta, empresarial	Muy alta, frontend, backend	Alta, dominante en web	Alta	Creciente rápidamente
<b>Usos</b>	Web, ciencia de datos, machine learning	Grandes aplicaciones, Android, big data	Web	Web, CMS	Web	Microservicios, sistemas distribuidos, DevOps
<b>Ventajas</b>	Legible, versátil, excelente comunidad	Portabilidad, robustez, fuerte soporte empresarial	Asíncrono, rápido en I/O, full-stack con Node.js	Amplia base de código existente, fácil de implementar	Sintaxis clara, desarrollo ágil	Rendimiento sobresaliente, concurrencia nativa, código limpio
<b>Desventajas</b>	Rendimiento inferior en tareas intensivas de CPU	Robusta, curva de aprendizaje moderada	Menor rendimiento en tareas intensivas de CPU, callbacks complejos	Malas prácticas históricas, menor modernización en el ecosistema	Menor rendimiento en proyectos grandes, comunidad más pequeña	Comunidad más pequeña, menor soporte para bibliotecas en comparación con otros lenguajes

### Estudio comparativo de los framework para backend

Características	Flask	Django	Express.js	SpringBoot	Laravel
<b>Lenguaje</b>	Python	Python	JavaScript/Node.js	Java	PHP
<b>Arquitectura</b>	Microframework	Full-stack MVC	Minimalista, middleware-based	Modular, microservicios	Full-stack MVC
<b>Curva aprendizaje</b>	Facil	Moderado	Facil	Moderado	Moderado
<b>Popularidad</b>	Alta	Alta	Muy Alta	Alta	Muy Alta
<b>Usos</b>	<ul style="list-style-type: none"> <li>- APIs REST</li> <li>- Prototipos rápidos</li> <li>- Microservicios</li> </ul>	<ul style="list-style-type: none"> <li>- Desarrollo rápido de aplicaciones</li> <li>- Sitios web con bases de datos complejas</li> </ul>	<ul style="list-style-type: none"> <li>- APIs REST</li> <li>- Aplicaciones en tiempo real</li> <li>- Microservicios</li> </ul>	<ul style="list-style-type: none"> <li>- Sistemas empresariales</li> <li>- Aplicaciones basadas en microservicios</li> </ul>	<ul style="list-style-type: none"> <li>- Aplicaciones web completas</li> <li>- APIs REST con soporte rápido</li> </ul>
<b>Seguridad</b>	Moderada	Alta	Moderada	Alta	Alta
<b>Escalabilidad</b>	Moderada	Alta	Moderada	Alta	Alta
<b>Ventajas</b>	<ul style="list-style-type: none"> <li>- Minimalista y flexible</li> <li>- Fácil de aprender</li> <li>- Ideal para microservicios y APIs rápidas</li> </ul>	<ul style="list-style-type: none"> <li>- Herramientas integradas</li> <li>- Seguridad robusta</li> <li>- Gran documentación</li> <li>- Escalable</li> </ul>	<ul style="list-style-type: none"> <li>- Ligero y minimalista</li> <li>- Gran comunidad y soporte</li> <li>- Amplia compatibilidad con paquetes</li> </ul>	<ul style="list-style-type: none"> <li>- Alta escalabilidad</li> <li>- Ideal para aplicaciones empresariales</li> <li>- Comunidad sólida</li> </ul>	<ul style="list-style-type: none"> <li>- Fácil de aprender</li> <li>- Integración con bases de datos y ORM (Eloquent)</li> <li>- Soporte para plantillas Blade</li> </ul>
<b>Desventajas</b>	<ul style="list-style-type: none"> <li>- Carece de herramientas integradas para proyectos grandes</li> <li>- Requiere</li> </ul>	<ul style="list-style-type: none"> <li>- Poco flexible para proyectos fuera del modelo estándar MVC</li> </ul>	<ul style="list-style-type: none"> <li>- Requiere configuración adicional para proyectos grandes</li> <li>- No sigue patrones estrictos</li> </ul>	<ul style="list-style-type: none"> <li>- Curva de aprendizaje alta</li> <li>- Sobrecarga en proyectos pequeños</li> </ul>	<ul style="list-style-type: none"> <li>- Rendimiento más bajo comparado con otros frameworks</li> <li>- No ideal para microservicios</li> </ul>

	configuración manual				
--	----------------------	--	--	--	--

## Lenguajes de Programacion y Framework para Frontend

React es una potente biblioteca de JavaScript para crear interfaces de usuario, consistentes y coherentes, su enfoque declarativo simplifica la creación de la interfaz de usuario y mejora la legibilidad del código, mientras que el DOM virtual garantiza un alto rendimiento al actualizar los elementos necesarios, de igual forma, su arquitectura promueve la modularidad, escalabilidad y un mantenimiento más sencillo a través de componentes reutilizables.

### Características

- **Interactividad y Componentización:** React es ideal para construir interfaces de usuario dinámicas y altamente interactivas. Su arquitectura basada en componentes permite modularizar el frontend, facilitando el desarrollo, mantenimiento y escalabilidad.
- **Velocidad y Rendimiento:** Utiliza el Virtual DOM, que actualiza solo las partes de la interfaz que cambian, mejorando el rendimiento en aplicaciones con interacciones complejas.
- **Ecosistema Amplio:** React tiene una gran comunidad y una rica variedad de librerías y herramientas para extender su funcionalidad, como React Router para la navegación o Redux para la gestión del estado global.
- **Facilidad para Integración de Mapas:** Herramientas como Leaflet.js o Google Maps API se integran fácilmente con React, lo cual es esencial para implementar el mapa interactivo de torres de internet.
- **Experiencia del Usuario (UX):** Permite crear aplicaciones de una sola página (SPAs), ofreciendo una experiencia fluida sin recargar la página.

## Estudio comparativo de los lenguajes de programación para frontend

Características	JavaScript	Kotlin	Swift	Html/Css	TypeScript
<b>Naturaleza</b>	Lenguaje de scripting	Lenguaje compilado, multiplataforma	Lenguaje compilado, para sistemas Apple	Lenguajes de marcado y estilo	Superconjunto tipado de JavaScript
<b>Facilidad aprendizaje</b>	Alta	Media	Medio	Muy alta	Media
<b>Rendimiento</b>	Bueno	Bueno	Alta	N/A	Bueno
<b>Seguridad</b>	Moderado necesita configuracion	Alta	Alta	N/A	Alta

<b>Compatibilidad</b>	Excelente	Limitada con frameworks	Limitada	Excelente	Excelente
<b>Ecosistema</b>	Amplio	Creciente	Enfocado en iOS	Amplio	Amplio
<b>Usos</b>	Desarrollo web interactivo, SPAs	Aplicaciones multiplataforma con Kotlin/JS	Aplicaciones frontend para iOS	Diseño y estructura web	Desarrollo web, SPAs
<b>Ventajas</b>	Versátil, amplio ecosistema, Node.js	Seguro, conciso, interoperable con Java, multiplataforma	Rendimiento, seguro, integrado con Xcode	Estándar de la web, fácil de aprender	Tipado estático, escalabilidad, mejor refactorización
<b>Desventajas</b>	Tipado dinámico puede llevar a errores, prototipado puede ser confuso	Curva de aprendizaje más pronunciada para desarrolladores web tradicionales	Ecosistema más pequeño que JavaScript, principalmente para Apple	No es un lenguaje de programación, requiere JavaScript para interactividad	Mayor curva de aprendizaje para desarrolladores JavaScript puros

### Estudio comparativo de los framework para el frontend

Características	React	Angular	Vue	Ember	Meteor
<b>Lenguaje</b>	JavaScript/JSX	TypeScript	JavaScript	JavaScript	JavaScript
<b>Arquitectura</b>	Biblioteca	MVC completo	Se adapta segun las necesidades	MVC	Full-stack
<b>Enfoque</b>	Componentes UI	Aplicaciones web de una sola pagina	Componentes UI	Aplicaciones ambiciosas	Desarrollo rápido de aplicaciones web
<b>Curva aprendizaje</b>	Moderado	Alta	Baja	Alta	Moderado

<b>Rendimiento</b>	Alto	Alto	Alto	Bueno en proyectos grandes	Adecuado para aplicaciones tiempo real
<b>Ecosistema</b>	Extenso herramientas	Amplio	Creciente y modular	Moderado	Completo
<b>Comunidad</b>	Muy grande y activa	Grande y activa	Creciente	Establecida pero mas pequeña	Activa
<b>Documentacion</b>	Excelente	Muy buena	Excelente	Buena	Buena
<b>Usos</b>	Aplicaciones web interactivas	Aplicaciones empresariales grandes	Proyectos pequeños a medianos	Aplicaciones ambiciosas	Aplicaciones en tiempo real y prototipos
<b>Ventajas</b>	Flexibilidad, rendimiento gracias al DOM virtual, fuerte comunidad	Solución integral, soporte oficial de Google, herramientas robustas	Fácil de aprender, documentación clara, ligereza	Convención sobre configuración, enfoque estructurado	Integración nativa con backend, desarrollo rápido
<b>Desventajas</b>	Necesita configuración adicional para proyectos grandes, dependencia en bibliotecas externas	Curva de aprendizaje pronunciada, puede ser pesado para proyectos pequeños	Ecosistema más pequeño que Angular o React, menor adopción empresarial	Menor comunidad y soporte, sobrecarga inicial para proyectos pequeños	Menor flexibilidad en comparación con otros frameworks, limitado para proyectos frontend-only

## Bases de Datos

### Características

- Soporte Avanzado para Consultas Complejas: PostgreSQL admite características avanzadas como consultas espaciales, que son útiles para trabajar con mapas (ubicación de torres y zonas de cobertura).
- Confiabilidad y Escalabilidad: Es conocido por su estabilidad y capacidad para manejar grandes volúmenes de datos, lo cual es importante para gestionar datos de clientes, facturación y monitoreo de red.

- Características de Integridad de Datos: Permite definir claves foráneas, restricciones y reglas avanzadas para garantizar la integridad de los datos (por ejemplo, validación de pagos o relación entre torres y clientes).
- Compatibilidad con Django: Django tiene soporte nativo para PostgreSQL, lo que permite aprovechar características avanzadas como índices específicos o tipos de datos personalizados.
- Optimización de Rendimiento: PostgreSQL permite manejar simultáneamente múltiples conexiones, asegurando un buen rendimiento incluso en aplicaciones con muchos usuarios concurrentes.

### Estudio comparativo de los DBMS

Características	SQLServer	PostgreSQL	MySQL	Oracle
<b>Licencia</b>	Comercial	Open Source	GPL y Comercial	Comercial
<b>Rendimiento</b>	Alto	Excelente	Alto en lecturas	Muy alto
<b>Escalabilidad</b>	Escalabilidad vertical y horizontal limitada	Escalabilidad horizontal con herramientas adicionales	Escalabilidad vertical y limitada horizontalmente	Escalabilidad vertical y horizontal (Oracle RAC)
<b>Soporte</b>	SQL estándar con características propias	SQL estándar con soporte avanzado	SQL estándar con extensiones	SQL estándar con características propias
<b>Transacciones</b>	Soporta ACID, con opciones avanzadas	Soporta ACID, con transacciones complejas	Soporta ACID	Soporta ACID, con opciones avanzadas
<b>Integridad</b>	Claves primarias, foráneas y restricciones avanzadas	Claves primarias, foráneas y restricciones avanzadas	Claves primarias y foráneas	Claves primarias, foráneas y restricciones avanzadas
<b>Popularidad</b>	Dominante en el mercado empresarial, especialmente en	Muy popular en comunidades de código abierto y aplicaciones que	Muy popular, especialmente en entornos web de menor escala.	Líder histórico en el mercado empresarial, conocido por su



	entornos de Microsoft.	requieren alta escalabilidad y flexibilidad.		alta disponibilidad y rendimiento.
<b>Ventajas</b>	Buena integración con servicios de Microsoft, herramientas gráficas	Muy robusto, extensible, soporte nativo para JSON	Fácil de usar, amplia comunidad, soporte de replicación	Muy seguro, alta disponibilidad, rendimiento optimizado
<b>Limitaciones</b>	Falta de soporte empresarial robusto en comparación con MySQL	Mayor curva de aprendizaje para configuraciones avanzadas	Menor rendimiento en consultas complejas, falta de soporte completo para algunas funciones avanzadas	Requiere licencia, alto consumo de recursos en entornos complejos

## Entornos de Desarrollo

Visual Studio Code es una herramienta completa que se adapta perfectamente a las necesidades del proyecto, optimizando el tiempo de desarrollo y asegurando una experiencia de programación eficiente y organizada.

### Características

- **Eficiencia:** VS Code es liviano pero potente, ideal para proyectos que combinan múltiples tecnologías.
- **Versatilidad:** Soporta todo el stack tecnológico (React.js, Django, PostgreSQL) con extensiones específicas.
- **Flujo de Trabajo Optimizado:** Su integración de terminal, Git y depuración permite un desarrollo fluido y centralizado.

Características	Netbeans	Eclipse	Sublime Text	Aptana	Visual Studio Code
<b>Facilidad de uso</b>	Media	Media	Alta	Media	Alta
<b>Rendimiento</b>	Medio	Medio	Muy Alto	Medio	Muy alto

<b>Lenguajes Soportados</b>	Java, PHP, C++, HTML/CSS	Java, C++, Python, PHP	Multilenguaje (limitado sin plugins)	HTML, CSS, JavaScript, PHP	Multilenguaje (TypeScript, Python, Java, etc.)
<b>Extensibilidad</b>	Moderado	Alta	Alta	Moderado	Muy alta
<b>Soporte</b>	Bueno	Muy bueno	Limitada	Bueno	Excelente
<b>Ventajas</b>	Completo, fácil de usar	Muy extensible, comunidad grande	Rápido, ligero, personalizable	Enfocado en web, integración con Git	Muy popular, extensible, ligero, depuración
<b>Desventajas</b>	Puede ser pesado para proyectos pequeños, menos personalizable que Sublime Text	Puede ser lento al iniciar, interfaz puede resultar compleja	Pocas funcionalidades integradas, requiere plugins	Menos popular y actualizado que otros IDEs	Aún en desarrollo activo, algunas funcionalidades pueden estar en preview