ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA DE SOFTWARE

TEMA:

HERRAMIENTAS Y TECNOLOGÍAS

NOMBRE:

DARWIN MORALES

ASIGNATURA:

APLICACIONES INFORMÁTICAS II

PERIODO:

OCTUBRE 2024 – FEBRERO 2025

Herramientas y tecnologías

Las herramientas para desarrollar software son tecnologías que agilizan y optimizan el desarrollo de este. El desarrollo de software es un proceso complejo que consiste en traducir objetos del mundo real en representaciones matemáticas y electrónicas que las máquinas puedan entender y manipular. Las herramientas actúan como una interfaz entre la realidad física y los procesos informáticos. Incluyen lenguajes de programación, marcos y plataformas que abstraen diferentes niveles de complejidad. Esto significa que puede interactuar con las computadoras más fácilmente y resolver problemas más complejos. En lugar de trabajar con componentes de hardware y lenguajes de codificación de bajo nivel, puede trabajar con bibliotecas, API y otras abstracciones que prioricen los casos de uso empresarial. Estas herramientas también incluyen aplicaciones de software, componentes y servicios que simplifican el proceso de codificación (AWS, 2024).

Entre las herramientas utilizadas para el desarrollo de software tenemos las siguientes:

a) Lenguajes de programación

Los lenguajes de programación son sistemas creados para formular tareas computacionales que serán ejecutadas por ordenadores. Estos lenguajes incluyen un conjunto de términos específicos, símbolos, y reglas de sintaxis y semántica que definen cómo deben estructurarse e interpretarse sus componentes y expresiones (Martínez, 2018)

b) Base de datos

Una base de datos es una representación unificada de los conjuntos de entidades instanciadas que corresponden a las diversas entidades tipo y sus interrelaciones. Este conjunto estructurado de datos debe ser accesible y utilizable de manera compartida por múltiples usuarios de diferentes perfiles (Camps et al., 2005). Las bases de datos son la opción preferida para guardar datos de manera organizada. Desde las aplicaciones más grandes con múltiples usuarios, hasta los teléfonos y agendas electrónicas, utilizan tecnología de bases de datos para garantizar que los datos se mantengan completos y facilitar el trabajo tanto de los usuarios como de los desarrolladores que las crearon.

c) IDE

Un IDE es un programa diseñado para facilitar la labor de diseño y creación de un nuevo software. Los IDEs integran distintas herramientas y funcionalidades en un entorno único, siendo las más importantes un editor de código fuente, herramientas de construcción automáticas y un depurador (Dumitru, 2019)

d) Framework

Son un marco de trabajo, que incluyen modelos, conceptos, paradigmas, criterios a seguir, conteniendo herramientas y funciones predeterminadas que facilitan el incremento de la velocidad de las funciones de programación. Estas herramientas ayudan a evitar un código repetitivo, mantener un orden, asegurando la actividad de programación, organizada, clara, para desarrollar mucho más rápido el proyecto deseado (Sosa & Silva, 2021).

Lenguajes de programación

Para la selección del lenguaje de programación a usar en el proyecto, se realizó un cuadro comparativo de los siguientes lenguajes:

Tabla 1: Cuadro comparativo de lenguajes de programación

Lenguaje / Criterio	TypeScript	JavaScript	Python	Java	Ruby
Compatibilidad con herramientas y framework	Funciona perfectamente con herramientas modernas como Angular, React y NestJS, que son muy útiles para construir aplicaciones dinámicas y atractivas. Esto permite un desarrollo más rápido y organizado	También se lleva bien con las mismas herramientas, pero como no incluye algunas características adicionales (como el control de errores antes de que sucedan), puede complicar un poco el manejo de proyectos grandes.	Funciona bien con herramientas como Flask o Django para aplicaciones web, pero estas no son tan utilizadas como Angular o React en desarrollos modernos.	Es confiable para aplicaciones grandes gracias a herramientas como Spring, pero su configuración inicial puede ser más compleja.	Se usa principalmente con on Rails, que es fácil de aprender y rápido para proyectos pequeños o medianos, aunque es menos popular para aplicaciones grandes.
Velocidad de la aplicación	Es lo suficientemente rápido porque convierte el código en algo que los navegadores entienden fácilmente. Además, su diseño ayuda a	También es rápido, pero si no se organiza bien el código, puede haber problemas de rendimiento a medida que crece el proyecto.	Es algo más lento porque no está diseñado especialmente para aplicaciones web de alto rendimiento, aunque funciona bien	Es muy rápido y eficiente para manejar aplicaciones grandes, aunque requiere más tiempo para prepararlo correctamente.	Suele ser más lento, especialmente para aplicaciones grandes, porque su enfoque principal está en ser fácil de

	mantener un buen rendimiento en proyectos grandes		para proyectos simples.		usar y no en ser el más rápido
Cantidad de herramientas disponibles	Tiene una gran cantidad de herramientas modernas y actualizadas, lo que lo convierte en una excelente opción para construir proyectos sólidos y fáciles de mantener.	También tiene muchas herramientas disponibles, pero algunas pueden no ser tan completas ni adecuadas para proyectos de gran tamaño.	Ofrece herramientas útiles para diversos tipos de proyectos, pero no siempre están diseñadas para las necesidades de aplicaciones web modernas	Tiene herramientas muy potentes, ideales para aplicaciones de gran escala, aunque estas pueden ser más complicadas de usar.	Las herramientas son fáciles de usar y aprender, pero no son tan variadas ni actualizadas como las de otros lenguajes.
Soporte para evitar errores en el código	Ayuda a identificar errores desde que escribes el código, lo que te ahorra tiempo en pruebas y correcciones. Esto es muy útil cuando trabajas en proyectos grandes o en equipo.	No tiene esta función, por lo que es más fácil cometer errores que solo se detectan cuando ejecutas la aplicación. Esto puede ser problemático en proyectos grandes.	Tampoco incluye esta función de forma nativa, aunque puedes usar otras herramientas para simularla, pero no es tan eficiente.	Es bueno para evitar errores porque requiere que seas muy específico al escribir el código, pero esto puede hacer que sea más difícil de aprender.	No tiene esta función, por lo que los errores pueden pasar desapercibidos hasta que se pruebe la aplicación.
Facilidad de aprendizaje	Es relativamente fácil si ya conoces JavaScript, y las herramientas modernas lo hacen más sencillo de usar. Además, al ser organizado, es ideal para	Es muy fácil de aprender, especialmente para principiantes, ya que tiene una estructura simple y muchas personas lo usan. Sin	Es bastante sencillo y popular entre quienes empiezan a programar, pero no está tan orientado al desarrollo web avanzado como otros lenguajes.	Puede ser difícil de aprender al principio porque tiene muchas reglas y su estructura es más compleja, pero es muy útil para proyectos grandes.	Es fácil de aprender porque su estructura es simple y directa. Es perfecto para proyectos pequeños, pero puede no ser suficiente para

	aprender a manejar proyectos más grandes.	embargo, puede complicarse cuando el proyecto crece mucho.			aplicaciones muy grandes.
Integración con APIs de IA	Funciona muy bien con herramientas de inteligencia artificial como OpenAI o Google AI, gracias a que tiene soporte para librerías modernas y es fácil de integrar con sistemas web.	También funciona bien con estas herramientas, pero sin algunas características (como el control de errores), puede ser más complicado en proyectos avanzados	Es muy bueno para trabajar con IA gracias a librerías especializadas como transformers, aunque estas no siempre están enfocadas en aplicaciones web.	Puede usarse con herramientas de IA, pero no es tan popular para proyectos enfocados exclusivamente en chatbots o asistentes virtuales.	También puede usarse con APIs de IA, pero no es tan eficiente ni popular para este propósito.

Tabla 2: Comparación final

Lenguajes	TypeScript
JavaScript	TypeScript previene errores desde el momento en que escribes el código gracias a su tipado estático, lo que hace que los proyectos sean más seguros y fáciles de mantener.
Python	TypeScript tiene mejor integración con herramientas y frameworks web modernos como Angular o React, lo que lo hace ideal para aplicaciones web complejas y dinámicas.
Java	TypeScript es más fácil de aprender y configurar para aplicaciones web, además de ser más ágil para proyectos que necesitan rápido desarrollo y despliegue.
Ruby	TypeScript es más rápido y escalable, además de tener un ecosistema más moderno y popular para el desarrollo de aplicaciones web grandes y complejas.

Conclusión

TypeScript se destaca como el lenguaje ganador para este proyecto debido a su capacidad para mejorar la calidad del código a través del tipado estático, su compatibilidad con JavaScript y su integración con frameworks modernos como Angular y Node.js. Esto lo convierte en una opción ideal para el desarrollo tanto del frontend como del backend de la aplicación.

Base de datos

Para la selección de la base de datos a usar en el proyecto, se realizó un cuadro comparativo señalando las ventajas y desventajas de las distintas bases de datos:

Base de datos	Ventajas	Desventajas
PostgreSQL	• Es una base de datos muy estable y robusta, ideal para manejar grandes cantidades de datos y asegurar que todo funcione correctamente. • Puede manejar consultas muy detalladas y con muchas conexiones entre diferentes tipos de información • Es recomendable para proyectos que van creciendo con el tiempo, ya que puede adaptarse sin perder rendimiento a medida que aumentan los usuarios o los datos	• Puede ser un poco más lenta en aplicaciones simples donde no se necesita manejar tantas relaciones o datos complejos. • Puede ser más difícil de configurar correctamente para aprovechar todas sus ventajas
MySQL	• Es muy conocida y fácil de usar, por lo que mucha gente sabe cómo trabajar con ella y hay muchos recursos disponibles en línea. • Debido a su popularidad, es fácil encontrar soluciones a problemas o pedir ayuda en línea • Funciona bien cuando el proyecto no es demasiado grande, por lo que es una buena opción para proyectos de tamaño medio.	• No maneja tan bien ciertas funciones avanzadas, como el tipo de datos JSON • Si el proyecto requiere consultas muy complicadas o con muchas relaciones entre los datos, MySQL puede no ser tan rápido o eficiente como PostgreSQL.
MongoDB	• Es recomendada cuando tienes información que no se ajusta fácilmente a un formato tradicional, como conversaciones o datos que cambian constantemente. • Te permite ajustar y ampliar fácilmente tu base de datos a medida que crece tu proyecto, sin complicaciones.	• Si se tiene información que debe conectarse de manera muy estructurada, como usuarios con reservas o laboratorios, MongoDB no es la mejor opción. • A medida que el proyecto crece, puede ser más difícil mantener la consistencia de los datos, es decir, asegurarse de que todo esté sincronizado correctamente.
SQL Server	• Es una base de datos muy potente, diseñada para	• Requiere de una inversión considerable y es más difícil

	manejar grandes cantidades de datos que deben ser gestionados de manera eficiente. • Si se trabaja en una empresa que ya usa otras herramientas de Microsoft, SQL Server se integra bien con ellas.	de gestionar, por lo que puede no ser ideal para proyectos más pequeños o para personas que no están acostumbradas a su uso. • Aunque es excelente para grandes empresas, puede ser más rígida o menos compatible con tecnologías modernas de desarrollo web.
Firebase	• Es ideal para proyectos que usan tecnologías como Angular, ya que se integra fácilmente con el frontend. • Es buena para aplicaciones que necesitan actualizaciones instantáneas, como chats o reservas en tiempo real.	• Si necesitas hacer preguntas muy detalladas o con relaciones complejas entre los datos, Firebase no es tan eficaz. • A medida que crece el proyecto, el uso de Firebase puede volverse más costoso. Además, depende completamente de Google, lo que puede ser una desventaja si no te gusta estar atado a una sola empresa.

Conclusión

Se decidió optar por PostgreSQL ya que combina lo mejor de los mundos relacional y moderno. Para el proyecto, donde se tiene relaciones complejas entre usuarios, inventarios y reservas, PostgreSQL ofrece un sistema sólido que maneja consultas complejas sin problemas. Además, PostgreSQL es la mejor opción por su equilibrio entre rendimiento, escalabilidad y flexibilidad.

IDE

Para la selección del IDE a usar para el desarrollo del proyecto, se realizó un cuadro comparativo tomando en cuenta los siguientes criterios:

IDE / **Criterio**	**Visual Studio Code**	**Eclipse**	**IntelliJ IDEA**	**NetBeans**	**PyCharm**
Lenguajes soportados	Soporta múltiples lenguajes como JavaScript, TypeScript, Python, C++, Java, PHP, y más a través de extensiones	Principalmente Java, pero también soporta C/C++, PHP, Python, y otros mediante plugins.	Soporta Java de forma nativa, y también otros lenguajes como Kotlin, Groovy, Scala, y más a través de plugins.	Soporta Java, PHP, JavaScript, HTML, CSS, y otros lenguajes mediante plugins.	Especializado en Python, pero también soporta JavaScript, HTML, CSS, y otros a través de plugins.

Facilidad de Uso	Interfaz intuitiva y personalizable, ideal para principiantes y desarrolladores experimentados	Puede ser complejo para nuevos usuarios debido a su amplia funcionalidad y configuración	Interfaz amigable y fácil de usar, con muchas características automatizadas que facilitan el desarrollo.	Interfaz sencilla, pero puede ser menos intuitiva que otros IDEs para algunos usuarios.	Muy fácil de usar, especialmente para desarrolladores de Python, con muchas características automatizadas.
Rendimiento	Ligero y rápido, se carga rápidamente y maneja proyectos grandes sin problemas.	Puede ser más pesado y lento, especialmente con muchos plugins instalados	Generalmente rápido, pero puede volverse lento con proyectos muy grandes o configuraciones complejas.	Rendimiento aceptable, aunque puede ser más lento en comparación con otros IDEs en proyectos grandes.	Muy eficiente para proyectos de Python, con un buen rendimiento en la ejecución y depuración.
Integración de Herramientas	Excelente integración con Git, terminal integrado, y una amplia gama de extensiones para herramientas de desarrollo.	Buena integración con herramientas de desarrollo, pero puede requerir configuración adicional.	Excelente integración con herramientas de desarrollo y sistemas de control de versiones.	Buena integración con herramientas de desarrollo, pero menos extensible que otros IDEs.	Muy buena integración con herramientas de desarrollo y sistemas de control de versiones, especialmente para Python.
Costo	Gratuito y de código abierto, con una amplia gama de extensiones gratuitas	Gratuito y de código abierto, aunque algunas herramientas pueden tener costos adicionales.	Versión Community gratuita, pero la versión Ultimate es de pago.	Gratuito y de código abierto.	Versión Community gratuita, pero la versión Professional es de pago.

Conclusión

Se optó inclinarse por Visual Studio Code, puesto que, es una buena opción para proyectos que requieren trabajar con múltiples lenguajes y herramientas. Además, su comunidad activa y la amplia gama de extensiones disponibles permiten personalizar el entorno de desarrollo según las necesidades específicas del proyecto. Esto lo hace especialmente adecuado para el desarrollo de aplicaciones web, como la que se plantea en el proyecto.

Framework

Para la selección los frameworks, se realizó un cuadro comparativo tomando en cuenta los siguientes criterios:

Framework	Descripción	Facilidad de uso	Rendimiento	Integración con TypeScript
Angular	Framework robusto para construir aplicaciones web de una sola página (SPA) utilizando TypeScript. Ofrece una arquitectura basada en componentes y un sistema de inyección de dependencias	Requiere un tiempo de aprendizaje inicial, pero es muy productivo.	Muy eficiente en la gestión de grandes aplicaciones.	Totalmente compatible con TypeScript, ya que fue diseñado para ser utilizado con él.
React	Biblioteca para construir interfaces de usuario, que permite crear componentes reutilizables. Aunque no es un framework completo, se puede utilizar con TypeScript.	Relativamente fácil de aprender, especialmente para quienes ya conocen JavaScript.	Muy rápido en la actualización de la interfaz de usuario gracias a su virtual DOM.	Compatible con TypeScript, aunque puede requerir configuración adicional para tipos
Vue.js	Framework progresivo para construir interfaces de usuario. Permite la integración gradual en proyectos existentes.	Fácil de aprender y utilizar, con una curva de aprendizaje suave.	Buen rendimiento, especialmente en aplicaciones pequeñas y medianas.	Compatible con TypeScript, aunque la integración puede no ser tan fluida como en Angular.

Svelte	Framework que compila componentes en código JavaScript altamente optimizado. Ofrece una experiencia de desarrollo fluida.	Muy fácil de aprender y utilizar, con una sintaxis clara y concisa.	Es de buen rendimiento, ya que el código se compila en tiempo de construcción	Compatible con TypeScript, aunque la integración puede requerir algunos ajustes.
NestJS	Framework para construir aplicaciones del lado del servidor utilizando TypeScript. Basado en Express, permite crear aplicaciones escalables y mantenibles.	Requiere conocimientos previos de Node.js y Express, pero es muy estructurado.	Muy eficiente para aplicaciones del lado del servidor.	Diseñado desde cero para TypeScrip
Express.js	Framework minimalista para Node.js que permite crear aplicaciones web y APIs de manera rápida y sencilla.	Resulta fácil de aprender y utilizar, y además es conveniente para principiantes.	Rápido y ligero, aunque puede requerir optimización para aplicaciones grandes.	Compatible con TypeScript, aunque puede requerir configuración adicional.
Fastify	Framework web para Node.js que se centra en la velocidad y la eficiencia.	Fácil de usar y configurar, con una buena documentación.	Muy alto rendimiento, optimizado para manejar muchas solicitudes.	Totalmente compatible con TypeScript.
Node.js	Entorno de ejecución para JavaScript en el lado del servidor, que permite construir aplicaciones escalables y rápidas.	Muy accesible para desarrolladores que ya conocen JavaScript.	Excelente rendimiento, especialmente en aplicaciones I/O intensivas	Compatible con TypeScript, aunque puede requerir configuración adicional.

Conclusión

Front-end: Angular es el framework más adecuado para proyectos que requieren una estructura amplia. Su integración nativa con TypeScript y su enfoque en la

arquitectura basada en componentes lo hacen ideal para aplicaciones grandes y complejas. Además, su amplia comunidad y recursos disponibles facilitan el aprendizaje y la resolución de problemas.

Back-end: Se optó por Node.js ya que utiliza un modelo de I/O no bloqueante, lo que permite manejar múltiples conexiones simultáneamente sin que el rendimiento se vea afectado. Esto es especialmente útil para aplicaciones que requieren un alto rendimiento y escalabilidad, como servicios en tiempo real. Además, cuenta con una variedad de módulos y paquetes disponibles a través de npm (Node Package Manager), lo que permite a los desarrolladores integrar fácilmente funcionalidades adicionales en sus aplicaciones.

Bibliografía

AWS. (2024). *Herramientas para desarrolladores*. Obtenido de https://aws.amazon.com/es/what-is/developer-tools/

Dumitru, R. (2019). *Herramientas y buenas prácticas para el desarrollo, mantenimiento y evolución de software en Java.* Madrid: Escuela Técnica Superior de Ingeniería de Sistemas Informáticos. Obtenido de https://oa.upm.es/55663/1/TFG_RADU_DUMITRU_BOBOIA.pdf#page=42&zoom=100,109,625

Sosa, S., & Silva, E. (2021). Herramientas de desarrollo de software utilizadas en el estado de Tabasco, México. *INNOVACIÓN Y DESARROLLO TECNOLÓGICO*, 2-10. Obtenido de https://iydt.wordpress.com/wp-content/uploads/2021/03/1_14_herramientas-de-desarrollo-de-software-utilizadas-en-el.pdf

Camps, R., Casillas, L. A., Costal, D., Gibert, M., Escofet, M., & Pérez, O. (2005). *Bases de datos* (S. Eureca Media, Ed.; Primea Edición). https://www.uoc.edu/pdf/masters/oficiales/img/913.pdf

Martínez, J. (2018). *Fundamentos de programación en Java* (Editorial EME, Ed.). https://www.tesuva.edu.co/phocadownloadpap/Fundamentos%20de%20programcion%20en%20Java.pdf