|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Versión | Autor | Organización | Descripción |
| 2024-11-18 | **1.0** | **Luis Adrian Méndez**  **Kevin Saeteros** | **Espoch** | **Elección de las tecnologías para el desarrollo del e-comerce** |
| 2024-11-21 | **1.1** | **Luis Adrian Méndez**  **Kevin Saeteros** | **Espoch** | **Actualización de tecnologías** |
|  |  |  |  |  |
|  |  |  |  |  |

**Lenguajes de Programación**

|  |  |  |  |
| --- | --- | --- | --- |
| Lenguaje | Ventajas | Desventajas | Recomendación |
| Python | Fácil de aprender, amplia comunidad, excelente para desarrollo web con Django/Flask y para IA/ML. | Menor rendimiento que lenguajes como Java o C++. | ✅ Ideal para Backend y AI. |
| JavaScript | Lenguaje universal para frontend y backend (Node.js), gran comunidad, soporte de frameworks modernos como React y Vue. | Manejo complejo de asincronía y gran cantidad de bibliotecas desactualizadas. | ✅ Ideal para Frontend. |
| PHP | Popular para e-commerce con herramientas como WordPress o Laravel. | Considerado más antiguo, menos eficiente en grandes escalas frente a Python o Node.js. | ❌ No recomendado. |
| Ruby | Conciso, utilizado en plataformas como Shopify. | Menor comunidad que Python y JS, menor rendimiento en procesos intensivos. | ❌ No recomendado. |

**Elección**:

* **Backend**: Javascript, ya que al utilizar el mismo lenguaje tanto en el frontend como en el backend, se reduce la curva de aprendizaje, se optimiza el desarrollo y se facilita la comunicación entre equipos.
* **Frontend**: JavaScript con frameworks modernos para una experiencia dinámica y fluida.

**IDEs (Entornos de Desarrollo Integrados)**

|  |  |  |  |
| --- | --- | --- | --- |
| IDE | Ventajas | Desventajas | Recomendación |
| Visual Studio Code | Gratuito, ligero, extensiones para todos los lenguajes y frameworks, excelente para JS y Python. | No tiene funciones avanzadas de debugging integradas (sin extensiones). | ✅ Ideal para todo el stack. |
| PyCharm | Excelente para Python, con herramientas de debugging avanzadas y gestión de entornos virtuales. | Pesado en sistemas con pocos recursos. | ✅ Para backend en Python. |

**Elección**:

* **General**: **Visual Studio Code** por su flexibilidad, soporte para múltiples lenguajes y extensiones.

**Frameworks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Framework | Lenguaje | Ventajas | Desventajas | Recomendación |
| Django | Python | Full-stack, incluye autenticación, administración, escalabilidad, excelente documentación. | Menos flexible que Flask si se requiere personalización. | ✅ Ideal para Backend. |
| Flask | Python | Microframework, ligero, flexible, ideal para APIs personalizadas. | No incluye funcionalidades preconstruidas como Django. | ✅ Alternativa flexible. |
| Node.js con Express | JavaScript | Unifica frontend y backend con JavaScript, rápido, soporte para APIs REST. | Curva de aprendizaje para manejo de asincronía. | ✅ Complemento JS Backend. |
| React | JavaScript | Framework específicamente para diseño de interfaz de usuarios | Dificultad en el manejo de componentes | ✅ Complemento para el lenguaje de programación Javascript |

**Elección**:

* **Backend**: **Node.js con express**, ya que ofrece todas las herramientas necesarias para realizar un e-commerce.
* **Frontend**: **React.js**, por su capacidad para crear interfaces rápidas y dinámicas.

**Sistemas de Gestión de Bases de Datos (SGBD)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SGBD | Tipo | Ventajas | Desventajas | Recomendación | |
| PostgreSQL | Relacional | Alto rendimiento, escalabilidad, excelente manejo de datos complejos, código abierto. | Requiere más configuración inicial comparado con MySQL. | ✅ Ideal para el proyecto. | |
| MySQL | Relacional | Popularidad, fácil de usar, amplia documentación. | Menos eficiente en manejo de datos complejos que PostgreSQL. | ✅ Alternativa válida. | |
| SQLite | Relacional | Ligero, ideal para pruebas locales. | No recomendado para sistemas con alto tráfico. | | ❌ Solo para pruebas. |
| Supabase | Plataforma de base de datos basada en PostgreSQL | Fácil de usar, ofrece autenticación y API REST integradas, ideal para proyectos ágiles con tiempos ajustados. | Dependencia de un proveedor externo, menos opciones de personalización avanzada comparado con PostgreSQL  puro. | ✅ Ideal para el proyecto. | |

Elección:

* **Principal**: PostgreSQL para asegurar escalabilidad y robustez.
* **Bibliografía:** Al Sweigart, A. (2019). *Automate the Boring Stuff with Python: Practical Programming for Total Beginners* (2nd ed.). No Starch Press.
* Flanagan, D. (2020). *JavaScript: The Definitive Guide* (7th ed.). O'Reilly Media.
* Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python* (2nd ed.). O'Reilly Media.
* Pollock, J. D. (2020). *React Quickly: Painless web apps with React, JSX, Redux, and GraphQL*. Manning Publications.
* Schütz, K. (2019). *Pro Express.js: Master Express.js—The Node.js Framework for Your Web Development*. Apress.
* PostgreSQL Global Development Group. (n.d.). PostgreSQL Documentation. Retrieved from <https://www.postgresql.org/docs/>
* Oracle Corporation. (n.d.). MySQL Documentation. Retrieved from [https://dev.mysql.com/doc/](https://dev.mysql.com/doc/" \t "_new)
* Supabase. (n.d.). *Supabase Documentation*. Retrieved from https://supabase.com/docs
* JetBrains. (n.d.). PyCharm Documentation. Retrieved from <https://www.jetbrains.com/pycharm/documentation/>
* Microsoft. (n.d.). Visual Studio Code Documentation. Retrieved from <https://code.visualstudio.com/docs>