**Análisis de Softwares a Utilizar**

**Nombre del Proyecto:** Aplicación Móvil para la Gestión de Parqueaderos en la Universidad Autónoma de Los Andes

**Fecha de Inicio:**

El objetivo de este documento es evaluar y seleccionar los softwares y herramientas necesarias para el desarrollo de la aplicación móvil, considerando aspectos técnicos, económicos y de compatibilidad con las necesidades del proyecto.

**Proceso de Análisis**

**1. Identificación de Necesidades del Proyecto**

Se realizó una reunión con el equipo de desarrollo para identificar las necesidades específicas del proyecto en cuanto a tecnologías y herramientas. Las necesidades incluyen:

* **Desarrollo de Backend**
* **Desarrollo de Frontend Móvil**
* **Gestión del Proyecto**
* **IDE (Entorno de Desarrollo Integrado)**
* **Base de Datos**

**2. Evaluación de Opciones**

Se identificaron varias opciones para cada necesidad y se evaluaron en base a los siguientes criterios:

* **Android Studio con Flutter**
* **Flutter**
* **Android Studio**
* **Node.js**
* **MySQL o PostgreSQ**
* **GitHub**

**Softwares Analizados.**

**Desarrollo de Backend**

|  |  |  |  |
| --- | --- | --- | --- |
| **Software** | **Descripción** | **Ventajas** | **Desventajas** |
| **Node.js** | Entorno de ejecución para JavaScript en el servidor. | Alta escalabilidad, gran comunidad de desarrolladores, soporte para APIs REST. | Requiere experiencia en JavaScript. |

**Bases de Datos**

|  |  |  |  |
| --- | --- | --- | --- |
| **Software** | **Descripción** | **Ventajas** | **Desventajas** |
| **MySQL** | Sistema de gestión de bases de datos relacional de código abierto. | Amplia adopción y soporte comunitario, alta performance y fiabilidad, fácil de aprender y usar. | Menos eficiente en consultas complejas en comparación con otros SGBD. |
| **PostgreSQL** | Sistema de gestión de bases de datos relacional y objeto-relacional de código abierto. | Soporte avanzado para tipos de datos y extensibilidad, alta fiabilidad y robustez, buen rendimiento en consultas complejas. | Mayor curva de aprendizaje comparada con MySQL. |

**Desarrollo de Frontend Móvil**

|  |  |  |  |
| --- | --- | --- | --- |
| **Software** | **Descripción** | **Ventajas** | **Desventajas** |
| **Flutter** | SDK de Google para el desarrollo de aplicaciones móviles nativas utilizando un solo código base en Dart. | Excelente rendimiento, soporte multiplataforma, interfaz atractiva y personalizable, hot reload. | Curva de aprendizaje debido a Dart, comunidad más pequeña comparada con otras tecnologías como React Native. |
| **Android Studio** | IDE oficial de Google para el desarrollo de aplicaciones Android. | Completo y potente entorno de desarrollo, herramientas avanzadas de depuración y profiling, integración nativa con herramientas de Google como Firebase, soporte para Flutter mediante plugins. | Puede ser pesado en recursos del sistema, curva de aprendizaje para nuevos desarrolladores. |

**Control de Versiones y Colaboración**

|  |  |  |  |
| --- | --- | --- | --- |
| **Software** | **Descripción** | **Ventajas** | **Desventajas** |
| **GitHub** | Plataforma de control de versiones basada en Git que facilita la colaboración entre equipos de desarrollo. | Control de versiones distribuido, colaboración en tiempo real y manejo de pull requests, integración con numerosas herramientas de CI/CD, gran comunidad y recursos disponibles. | Requiere conocimiento de Git, puede ser complejo para usuarios novatos. |

**Softwares que van a ser utilizados en el proyecto.**

|  |  |  |
| --- | --- | --- |
| **Área** | **Software Recomendado** | **Justificación** |
| **Backend** | Node.js | Alta escalabilidad, gran comunidad de desarrolladores y soporte para APIs REST. |
| **Bases de Datos** | PostgreSQL | Soporte avanzado para tipos de datos, alta fiabilidad y robustez en consultas complejas. |
| **Desarrollo de Frontend Móvil** | Flutter con Android Studio | Excelente rendimiento y soporte multiplataforma con un entorno de desarrollo completo y potente. |
| **Control de Versiones y Colaboración** | GitHub | Control de versiones distribuido, colaboración en tiempo real y una gran comunidad de desarrolladores. |

Los softwares que se utilizara son softwares que no tiene costo.

**Conclusiones y Recomendaciones**

* La selección de software se basó en la funcionalidad, facilidad de uso, compatibilidad, costo y soporte.
* Las herramientas recomendadas cumplen con los requisitos del proyecto y facilitarán un desarrollo eficiente y efectivo.

**Aprobaciones**

* **Aprobado por:**
* **Fecha de Aprobación:**
* **Firma:**