1. **INTRODUCCIÓN:** La Universidad Autónoma de Los Andes enfrenta desafíos en la gestión de sus parqueaderos debido al crecimiento de la población universitaria. Para mejorar la eficiencia, transparencia y comunicación en la administración de estos espacios, se propone implementar una Aplicación Móvil para la Gestión de Parqueaderos. Esta herramienta tecnológica busca optimizar el uso de los estacionamientos, beneficiando a estudiantes, personal y visitantes mediante una gestión más organizada y efectiva.
   1. **PROPOSITO:** El propósito de este proyecto es implementar una Aplicación Móvil para la Gestión de Parqueaderos en la Universidad Autónoma de Los Andes, con el objetivo de mejorar la eficiencia, transparencia y comunicación en la administración de los espacios de estacionamiento. Esto permitirá optimizar el uso de los parqueaderos, facilitando el acceso y gestión tanto para los estudiantes y el personal administrativo como para los visitantes del campus.
   2. **ÁMBITO DEL SISTEMA:** Para definir el ámbito del sistema, es importante considerar los límites y alcances de la Aplicación Móvil para la Gestión de Parqueaderos en la Universidad Autónoma de Los Andes. Este incluye todas las funcionalidades relacionadas con la reserva de espacios de estacionamiento, gestión de permisos de acceso, notificaciones en tiempo real, generación de reportes y estadísticas, así como la integración con sistemas de credenciales universitarios. El sistema abarca tanto a estudiantes, personal administrativo y visitantes, facilitando la gestión eficiente y transparente de los parqueaderos dentro del campus universitario, se utilizarán algunos recursos como:

* **RECURSOS DE SOFTWARE**

|  |  |
| --- | --- |
| **Requerimientos de Software del Sistema** | |
| **Sistema Operativo** | Android-IOS |
| **Explorador** | Play Store – App Store |
| **Gestor de Base de Datos** | Mysql - PostgreSQL |
| **Entorno de Desarrollo** | Node.js  PostgreSQL  Flutter con Android Studio  **Metodología Mobile-D** |

* **RECURSOS DE HARDWARE**

|  |  |
| --- | --- |
| **Requerimientos de Hardware del Sistema** | |
| **Requerimiento** | **Detalle** |
| **Espacio Físico** | Espacio Físico adecuado con su respectivo aire acondicionado lugar libre de humedad |
| **Estaciones de Trabajo** | Sistema Operativo Compatible: (iOS, Android, etc.)..  Memoria RAM: Generalmente, al menos 2GB de RAM es recomendable.  Conectividad: Acceso a internet, ya sea a través de Wi-Fi o datos móviles, especialmente si la aplicación requiere conexión constante. |
|  |  |
| **Equipo Suplementario**  **de Energía**  . | UPS APC 1000 VA 120 V SMT1000 |
| **Servidor** | Servidor de Aplicaciones con las siguientes  características:  Procesador: Intel Xeon o AMD EPYC.  Memoria RAM: Mínimo 16 GB, preferiblemente 32 GB o más.  Almacenamiento: Discos SSD para mayor velocidad y discos duros adicionales para almacenamiento masivo.  Red: Tarjetas de red de alta velocidad (1 Gbps o superior) |
|  |  |

* **PERSONAL INVOLUCRADO**

|  |  |
| --- | --- |
| Nombre | Yolanda Patricia Calapi Muñoz |
| Rol | LÍDER DE PROYECTOS |
| Categoría profesional | ING. DE SOFTWARE |
| Responsabilidades | Principal responsable del proyecto y del personal |
| Información de contacto | ypcalapi@gmail.com |
| Aprobación | si |

|  |  |
| --- | --- |
| Nombre | Janio Xavier Bunshe Aguirre |
| Rol | Desarrollador/Analista |
| Categoría profesional | Ingeniero de Software |
| Responsabilidades | Segundo responsable del proyecto y del personal |
| Información de contacto | jbunshe@gmail.com |
| Aprobación | si |

|  |  |
| --- | --- |
| Nombre | Pintag Duchicela Llanganate Muñoz |
| Rol | Desarrollador/Analista |
| Categoría profesional | Ingeniero de Software |
| Responsabilidades | Responsable del proyecto y del personal |
| Información de contacto | pintagllm@hotmail.com |
| Aprobación | si |

|  |  |
| --- | --- |
| Nombre | David Alexander Echeverría Merizalde |

**1.3. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS**

|  |  |
| --- | --- |
| Uniandes | Universidad para la cual se desarrolla la aplicación movil |
| UML | Unified Modeling Language, un lenguaje de modelado estándar en la ingeniería de software para visualizar, especificar, construir y documentar artefactos de sistemas. |
| Flutter | Un framework de desarrollo de aplicaciones de código abierto creado por Google, que permite crear aplicaciones nativas para iOS, Android, web y escritorio desde una única base de código. |
| Android Studio | Entorno de desarrollo integrado (IDE) oficial de Google para el desarrollo de aplicaciones Android. Proporciona herramientas para diseñar, codificar, depurar y probar aplicaciones Android. |

**1.4. REFERENCIAS**

Se listas a continuación otros documentos a los que se hace referencia desde éste**:**

|  |  |  |
| --- | --- | --- |
| **# TITULO** | **NUMERO** | **FECHA** |
| 1. IEEE Guide for Software Requirements Specification IEEE | Std 830-84 | 1994 |
| 1. OMG Unified Modeling Language Specification Version 1.4 | formal/2001-09-67 | 2001 |
| 1. Entrevista | Estudiante | 06/2024 |
| 1. Entrevista | Ex Estudiante | 06/2024 |

**2. DESCRIPCIÓN GENERAL**

**2.1. PERSPECTIVA DEL PRODUCTO:** La Aplicación Móvil para la Gestión de Parqueaderos en la Universidad Autónoma de Los Andes proporcionará una solución integrada y accesible para la reserva, gestión y supervisión de espacios de estacionamiento, mejorando la eficiencia, transparencia y experiencia del usuario en el campus.

**2.2. FUNCIONES DEL PRODUCTO**

***Funciones***

**Lista de Parqueaderos Campus**

**Parqueadero Disponibles**

**Alertas de cierre parqueo**

**Reporte de horas parqueadas**

**Integración con carnet**

Ilustración 1. Diagrama de descomposición de requerimientos de AWGESBE

**2.2.1 Descripción, entrada, proceso, salida, requerimientos no funcionales**

**Descripción**

### Lista de Parqueaderos Campus

**Descripción:** Proporciona una lista completa de todos los parqueaderos disponibles en el campus, incluyendo detalles como ubicación y capacidad.

**Entrada:**

* Solicitud del usuario para ver la lista de parqueaderos.

**Proceso:**

* Consultar la base de datos para obtener la información actualizada de los parqueaderos.
* Filtrar y organizar la información según criterios relevantes (por ejemplo, ubicación).

**Salida:**

* Visualización de la lista de parqueaderos con detalles de cada uno.

**Requerimientos no funcionales:**

* La respuesta debe ser rápida, mostrando la lista en menos de 2 segundos.
* Interfaz de usuario intuitiva y fácil de navegar.
* Datos actualizados en tiempo real.

### Parqueaderos Disponibles

**Descripción:** Muestra los espacios de estacionamiento disponibles en tiempo real.

**Entrada:**

* Solicitud del usuario para ver los espacios disponibles.

**Proceso:**

* Consultar la base de datos en tiempo real para verificar la disponibilidad de espacios.
* Actualizar y filtrar los datos según la ubicación del usuario o preferencias seleccionadas.

**Salida:**

* Visualización de los espacios de estacionamiento disponibles.

**Requerimientos no funcionales:**

* Actualización en tiempo real con un retraso máximo de 1 segundo.
* Alta disponibilidad y precisión de los datos.
* Baja latencia en la comunicación con la base de datos.

### Alertas de Cierre de Parqueo

**Descripción:** Notifica a los usuarios sobre el cierre temporal o permanente de los parqueaderos.

**Entrada:**

* Datos de cierre proporcionados por la administración del campus.

**Proceso:**

* Recibir y procesar la información de cierre.
* Enviar notificaciones a los usuarios afectados.

**Salida:**

* Alertas enviadas a los usuarios vía notificaciones push y correo electrónico.

**Requerimientos no funcionales:**

* Alta confiabilidad en la entrega de notificaciones.
* Interfaz de notificaciones clara y comprensible.
* Capacidad de manejar un alto volumen de notificaciones simultáneas.

### Reporte de Horas Parqueadas

**Descripción:** Permite a los usuarios ver un resumen de las horas que han utilizado los parqueaderos.

**Entrada:**

* Solicitud del usuario para ver su historial de estacionamiento.

**Proceso:**

* Consultar la base de datos para extraer el historial de horas parqueadas del usuario.
* Compilar y formatear la información en un reporte legible.

**Salida:**

* Visualización del reporte con detalles de las horas parqueadas.

**Requerimientos no funcionales:**

* Generación de reportes rápida y eficiente.
* Alta disponibilidad del servicio de reportes.
* Interfaz de usuario amigable para visualizar y entender los reportes.

### Integración con Carnet

**Descripción:** Permite a los usuarios utilizar su carnet universitario para acceder a los parqueaderos.

**Entrada:**

* Escaneo del carnet universitario del usuario.

**Proceso:**

* Verificación del carnet contra la base de datos de usuarios.
* Autorización de acceso al parqueadero basado en permisos asociados al carnet.

**Salida:**

* Autorización o denegación de acceso al parqueadero.

**Requerimientos no funcionales:**

* Alta seguridad y precisión en la verificación de identidad.
* Rápida respuesta en la autorización de acceso.
* Integración fluida con el sistema de control de acceso existente.

Estos componentes aseguran que la aplicación móvil no solo sea funcional, sino también eficiente, segura y fácil de usar, mejorando significativamente la experiencia de estacionamiento en la Universidad Autónoma de Los Andes.

**2.3. CARACTERÍSTICAS DE LOS USUARIOS**

#### Estudiantes

**Características:**

**Perfil Demográfico:** Jóvenes de entre 18 y 25 años, mayoría de tiempo parcial en el campus.

**Necesidades:**

* Acceso rápido y fácil a espacios de estacionamiento.
* Información en tiempo real sobre disponibilidad y alertas de cierres.
* Historial de uso de parqueaderos para gestionar el tiempo.

**Comportamiento:**

* Uso frecuente de dispositivos móviles.
* Preferencia por soluciones tecnológicas y aplicaciones móviles.
* Valoran la conveniencia y la rapidez.

#### Personal Administrativo

**Características:**

**Perfil Demográfico:** Adultos de entre 25 y 60 años, trabajan tiempo completo en el campus.

**Necesidades:**

* Gestión eficiente de sus permisos de estacionamiento.
* Información clara y accesible sobre cambios en la disponibilidad de parqueaderos.
* Herramientas para reportar y resolver problemas relacionados con el estacionamiento.

**Comportamiento:**

* Uso regular de dispositivos móviles y de escritorio.
* Necesidad de acceso a información detallada y reportes de uso.
* Valoran la precisión y la fiabilidad.

#### Visitantes

**Características:**

**Perfil Demográfico:** Varía ampliamente, incluyen conferenciantes, padres, proveedores y otros visitantes.

**Necesidades:**

* Información previa sobre disponibilidad de espacios de estacionamiento.
* Facilidad para hacer reservas anticipadas.
* Instrucciones claras y fáciles de seguir para acceder y utilizar los parqueaderos.

**Comportamiento:**

* Uso ocasional de la aplicación móvil.
* Preferencia por interfaces intuitivas y de fácil uso.
* Valoran la comodidad y la simplicidad en la información proporcionada.

**2.4. RESTRICCIONES**

**2.4.1. Protocolos de comunicación**

Los protocolos de comunicaciones entre los diferentes nodos de la infraestructura hardware de soporte serán los siguientes:

**2.4.2. Consideraciones de seguridad**

* Para acceso al sistema aplicar claves robustas

**2.4.3. Criticalidad de la aplicación**

* máximo permite 1000 usuarios conectados a la vez

**2.5. SUPOSICIONES Y DEPENDENCIAS**

* Se asume que los requisitos descritos en este documento son estables y que cumplen con todos los requerimientos una vez que sea aprobado por todo el personal involucrado en el desarrollo de esta aplicación.

**2.6. REQUERIMIENTOS FUTUROS**

* Alertas cuando un estacionamiento queda libre

**3.** **INTERFACES EXTERNAS**

**3.1.1 Interfaces de Usuario**

Administrador, profesional, control

**3.1.2 Interfaces Hardware**

**3.1.3 Interfaces Software**

Aplicación orientada a la Aplicación móvil

**3.1.4 Interfaces de Comunicación**

**N/A**

**3.2. REQUERIMIENTOS DE RENDIMIENTO**

* Número de usuarios conectados 150

**3.3. REQUERIMIENTOS ESPECIFICOS:**

3.3.1 Integración con el carnet estudiantil.

3.3.2 Listado de lugares de estacionamiento disponibles.