**UNIVERSIDAD REGIONAL AUTÓNOMA DE LOS ANDES “UNIANDES”**



**CARRERA DE INGENIERÍA DE SOFTWARE**

**Aplicación Móvil para la Gestión de Parqueaderos en la Universidad Autónoma de Los Andes**

**Requerimientos funcionales y no funcionales.**

**ESTUDIANTE:**

1. **Bunshe Aguirre Janio Xavier**
2. **Calapi Muñoz Yolanda Patricia**
3. **Llanganate Muñoz Pintag Duchicela**

**NIVEL: SEPTIMO SOFTWARE**

**JUNIO 2024**

**Definición de Requisitos para el Desarrollo de una Aplicación Móvil para la Gestión de Parqueaderos en la Universidad Autónoma de Los Andes**

La aplicación móvil para la gestión de parqueaderos en la Universidad Autónoma de Los Andes, mediante la lectura de credenciales institucionales, deberá cumplir con una serie de requisitos funcionales, no funcionales, técnicos, legales y de implementación.

**1. Requisitos Funcionales**

**1.1 Gestión de Usuarios**

* **Autenticación de Usuarios:**
  + Los usuarios deben poder autenticarse utilizando sus credenciales institucionales (por ejemplo, correo electrónico y contraseña de la universidad).
  + Implementación de Single Sign-On (SSO) utilizando el sistema de autenticación institucional.
* **Registro de Nuevos Usuarios:**
  + Provisión para el registro de nuevos usuarios utilizando las credenciales institucionales.
  + Validación de la identidad a través de la base de datos institucional.
* **Gestión de Perfiles de Usuario:**
  + Diferenciación entre perfiles de usuario (estudiantes, profesores, personal administrativo).
  + Capacidad para que los usuarios actualicen su información personal.

**1.2 Gestión de Parqueaderos**

* **Reserva de Espacios:**
  + Los usuarios deben poder ver la disponibilidad de espacios de parqueo en tiempo real.
  + Funcionalidad para reservar un espacio de parqueo específico.
* **Check-in y Check-out:**
  + Escaneo de credenciales institucionales para registrar la entrada y salida del parqueadero.
  + Registro automático del tiempo de uso del espacio de parqueo.
* **Visualización de Información:**
  + Mapa interactivo que muestre la ubicación de los parqueaderos y la disponibilidad de espacios.
  + Información detallada sobre cada espacio (tamaño, restricciones, etc.).
* **Notificaciones y Alertas:**
  + Envío de notificaciones push sobre la proximidad de la reserva y cambios en la disponibilidad.
  + Alertas sobre eventos especiales que afecten la disponibilidad de parqueaderos (por ejemplo, eventos universitarios).

**1.3 Administración del Sistema**

* **Panel de Administración:**
  + Interfaz para que los administradores gestionen los espacios de parqueo (habilitación, deshabilitación).
  + Visualización en tiempo real del estado de los parqueaderos y la ocupación.
* **Reportes y Análisis:**
  + Generación de reportes sobre la utilización de los espacios de parqueo.
  + Análisis de patrones de uso para optimizar la gestión del parqueadero.

**2. Requisitos No Funcionales**

**2.1 Seguridad**

* **Protección de Datos:**
  + Implementación de encriptación para proteger la información sensible.
  + Cumplimiento de normativas de protección de datos personales.
* **Autenticación Segura:**
  + Implementación de autenticación multifactor (MFA) si es necesario.
  + Protocolo seguro para la comunicación entre la aplicación móvil y el servidor (por ejemplo, HTTPS).

**2.2 Usabilidad**

* **Interfaz de Usuario:**
  + Diseño intuitivo y fácil de usar, compatible con diferentes tamaños de pantalla y sistemas operativos (iOS, Android).
* **Accesibilidad:**
  + Asegurar que la aplicación sea accesible para usuarios con discapacidades.

**2.3 Rendimiento**

* **Escalabilidad:**
  + Capacidad para manejar incrementos en el número de usuarios y transacciones.
* **Tiempo de Respuesta:**
  + Respuesta rápida en la visualización de disponibilidad y en la autenticación.

**3. Requisitos Técnicos**

**3.1 Integración**

* **Sistema de Credenciales Institucionales:**
  + Integración con el sistema de gestión de credenciales de la universidad.
* **Sistemas Existentes:**
  + Integración con otros sistemas universitarios (por ejemplo, sistema de horarios, gestión de estudiantes).

**3.2 Desarrollo y Tecnologías**

* **Frameworks y Herramientas:**
  + Uso de frameworks de desarrollo móvil como Flutter o React Native.
  + Backend con tecnologías como Node.js, Django, o Ruby on Rails.
* **Base de Datos:**
  + Diseño de la base de datos para almacenar información de usuarios, reservas, y parqueaderos.
* **API:**
  + Desarrollo de una API segura para la comunicación entre la aplicación móvil y el servidor.

**4. Requisitos Legales y Reglamentarios**

* **Cumplimiento Normativo:**
  + Asegurar cumplimiento con las leyes y regulaciones aplicables (protección de datos, accesibilidad).
* **Políticas Institucionales:**
  + Alineación con las políticas y regulaciones internas de la universidad.

**5. Requisitos de Implementación y Mantenimiento**

**5.1 Despliegue**

* **Lanzamiento Gradual:**
  + Despliegue inicial a un grupo pequeño de usuarios para pruebas en producción.
* **Feedback y Ajustes:**
  + Recoger feedback de los primeros usuarios y realizar ajustes antes del despliegue completo.

**5.2 Soporte y Mantenimiento**

* **Monitoreo Continuo:**
  + Monitorear el rendimiento y el uso de la aplicación continuamente.
* **Actualizaciones Regulares:**
  + Implementar actualizaciones y mejoras basadas en el feedback de los usuarios y nuevas necesidades.

**6. Pruebas y Validación**

**6.1 Pruebas Funcionales**

* **Verificación de Funcionalidades:**
  + Asegurar que todas las funcionalidades cumplen con los requerimientos especificados.

**6.2 Pruebas de Usabilidad**

* **Experiencia de Usuario:**
  + Validar que la aplicación sea fácil de usar y accesible.

**6.3 Pruebas de Seguridad**

* **Vulnerabilidades:**
  + Identificar y mitigar posibles vulnerabilidades de seguridad.

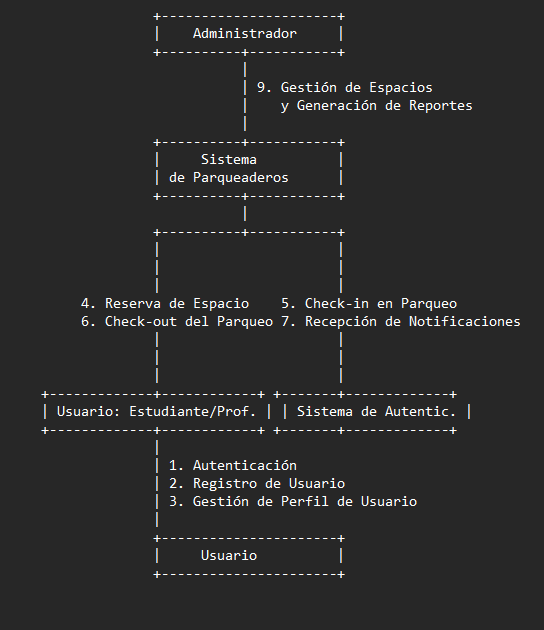
**Diagrama de Casos de Uso**

**Actores Principales**

* **Usuario:** Incluye estudiantes, profesores y personal administrativo que utilizan la aplicación para gestionar su parqueo.
* **Administrador:** Personal responsable de la gestión y administración de los parqueaderos.
* **Sistema de Autenticación:** Sistema institucional para la validación de credenciales.

**Casos de Uso**

1. **Autenticación de Usuario**
2. **Registro de Usuario**
3. **Gestión de Perfil de Usuario**
4. **Reserva de Espacio de Parqueo**
5. **Check-in en Parqueo**
6. **Check-out del Parqueo**
7. **Recepción de Notificaciones**
8. **Gestión de Espacios por Administrador**
9. **Generación de Reportes**

**Diagrama de Caso de Uso General**

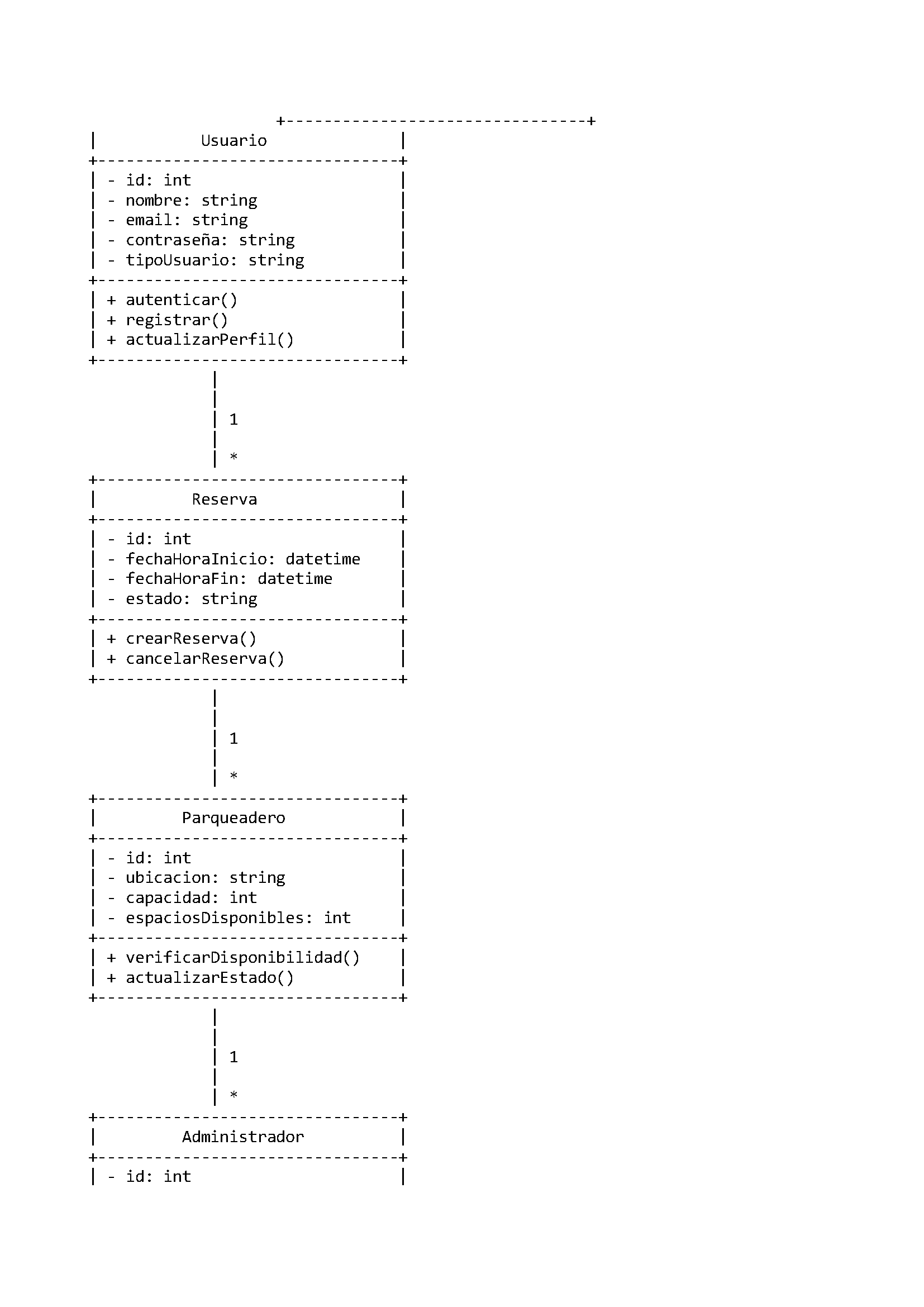
**Descripción de Casos de Uso**

1. **Autenticación de Usuario**
   * **Actor Primario:** Usuario
   * **Descripción:** El usuario se autentica utilizando sus credenciales institucionales.
   * **Precondiciones:** El usuario debe tener credenciales válidas.
   * **Flujo Principal:**
     1. El usuario abre la aplicación.
     2. El usuario ingresa sus credenciales.
     3. El sistema valida las credenciales con el Sistema de Autenticación.
     4. El usuario accede a la aplicación.
2. **Registro de Usuario**
   * **Actor Primario:** Usuario
   * **Descripción:** Registro de nuevos usuarios utilizando credenciales institucionales.
   * **Precondiciones:** El usuario debe tener credenciales válidas.
   * **Flujo Principal:**
     1. El usuario selecciona la opción de registro.
     2. El usuario ingresa sus credenciales.
     3. El sistema valida las credenciales y registra al usuario.
3. **Gestión de Perfil de Usuario**
   * **Actor Primario:** Usuario
   * **Descripción:** Actualización de información personal por parte del usuario.
   * **Precondiciones:** El usuario debe estar autenticado.
   * **Flujo Principal:**
     1. El usuario accede a su perfil.
     2. El usuario actualiza su información personal.
     3. El sistema guarda los cambios.
4. **Reserva de Espacio de Parqueo**
   * **Actor Primario:** Usuario
   * **Descripción:** Reserva de un espacio de parqueo específico.
   * **Precondiciones:** El usuario debe estar autenticado.
   * **Flujo Principal:**
     1. El usuario visualiza la disponibilidad de espacios de parqueo.
     2. El usuario selecciona un espacio y realiza la reserva.
     3. El sistema confirma la reserva.
5. **Check-in en Parqueo**
   * **Actor Primario:** Usuario
   * **Descripción:** Registro de entrada al parqueadero mediante el escaneo de credenciales.
   * **Precondiciones:** El usuario debe tener una reserva activa.
   * **Flujo Principal:**
     1. El usuario llega al parqueadero.
     2. El usuario escanea su credencial.
     3. El sistema registra el check-in.
6. **Check-out del Parqueo**
   * **Actor Primario:** Usuario
   * **Descripción:** Registro de salida del parqueadero mediante el escaneo de credenciales.
   * **Precondiciones:** El usuario debe haber realizado el check-in.
   * **Flujo Principal:**
     1. El usuario escanea su credencial al salir.
     2. El sistema registra el check-out.
7. **Recepción de Notificaciones**
   * **Actor Primario:** Usuario
   * **Descripción:** Recepción de notificaciones push sobre la proximidad de la reserva y cambios en la disponibilidad.
   * **Precondiciones:** El usuario debe tener notificaciones habilitadas.
   * **Flujo Principal:**
     1. El sistema envía notificaciones push al usuario.
     2. El usuario recibe y visualiza las notificaciones.
8. **Gestión de Espacios por Administrador**
   * **Actor Primario:** Administrador
   * **Descripción:** Gestión de espacios de parqueo (habilitación, deshabilitación).
   * **Precondiciones:** El administrador debe estar autenticado.
   * **Flujo Principal:**
     1. El administrador accede al panel de administración.
     2. El administrador gestiona los espacios de parqueo.
     3. El sistema guarda los cambios.
9. **Generación de Reportes**
   * **Actor Primario:** Administrador
   * **Descripción:** Generación de reportes sobre la utilización de los espacios de parqueo.
   * **Precondiciones:** El administrador debe estar autenticado.
   * **Flujo Principal:**
     1. El administrador accede al panel de administración.
     2. El administrador selecciona la opción de generar reportes.
     3. El sistema genera y muestra los reportes.

### Diagrama UML

#### Diagrama de Clases

A continuación, se presenta el diagrama de clases UML que representa los principales componentes de la aplicación móvil para la gestión de parqueaderos.



Texto, Carta

Descripción generada automáticamente

**Descripción de Clases**

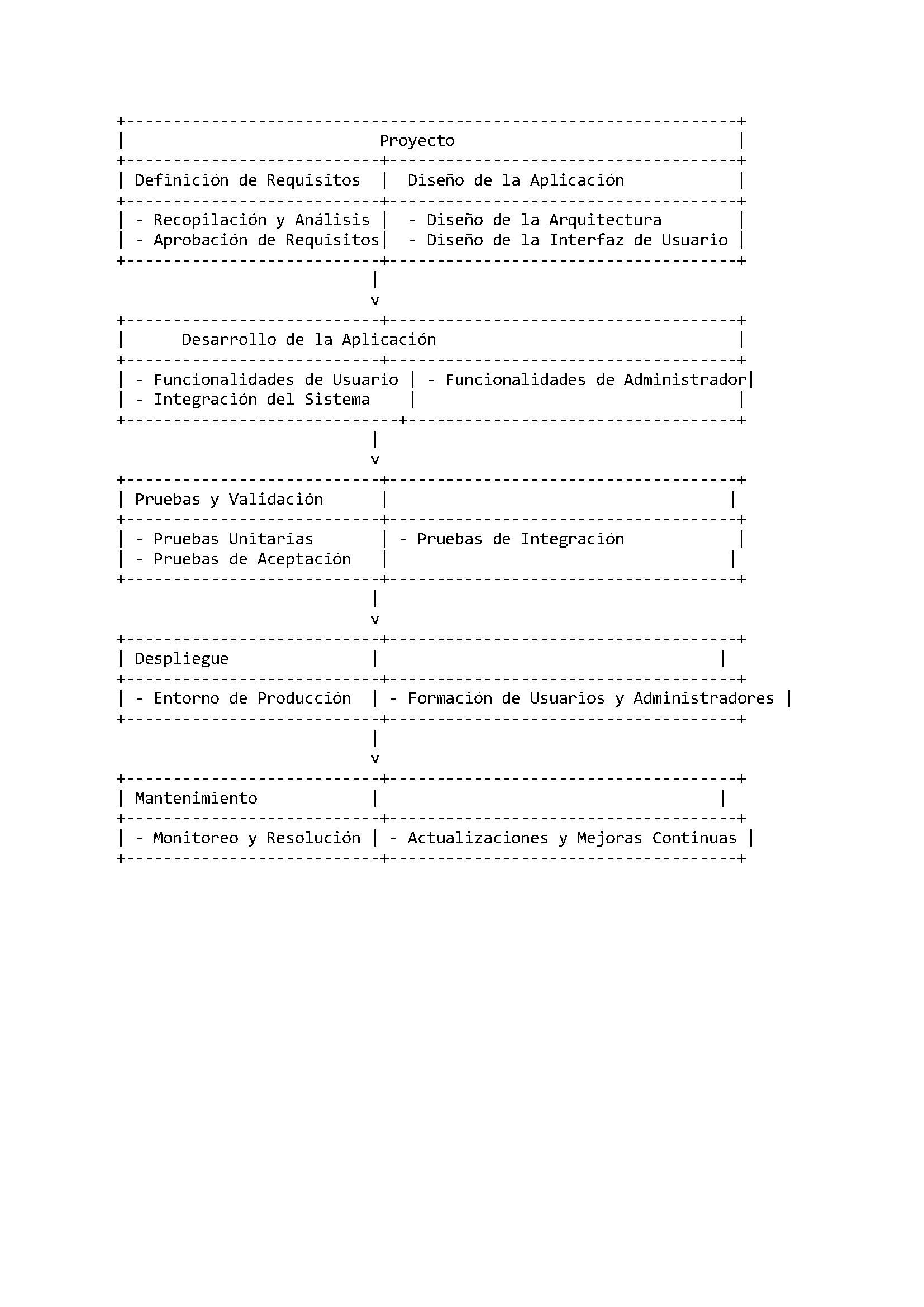
1. **Usuario:**
   * Atributos:
     + id: Identificador único del usuario.
     + nombre: Nombre del usuario.
     + email: Correo electrónico del usuario.
     + contraseña: Contraseña del usuario.
     + tipoUsuario: Tipo de usuario (estudiante, profesor, personal administrativo).
   * Métodos:
     + autenticar(): Autentica al usuario en el sistema.
     + registrar(): Registra un nuevo usuario en el sistema.
     + actualizarPerfil(): Actualiza la información del perfil del usuario.
2. **Reserva:**
   * Atributos:
     + id: Identificador único de la reserva.
     + fechaHoraInicio: Fecha y hora de inicio de la reserva.
     + fechaHoraFin: Fecha y hora de fin de la reserva.
     + estado: Estado de la reserva (activa, cancelada).
   * Métodos:
     + crearReserva(): Crea una nueva reserva.
     + cancelarReserva(): Cancela una reserva existente.
3. **Parqueadero:**
   * Atributos:
     + id: Identificador único del parqueadero.
     + ubicacion: Ubicación del parqueadero.
     + capacidad: Capacidad total del parqueadero.
     + espaciosDisponibles: Número de espacios disponibles en el parqueadero.
   * Métodos:
     + verificarDisponibilidad(): Verifica la disponibilidad de espacios en el parqueadero.
     + actualizarEstado(): Actualiza el estado de los espacios de parqueo.
4. **Administrador:**
   * Atributos:
     + id: Identificador único del administrador.
     + nombre: Nombre del administrador.
     + email: Correo electrónico del administrador.
     + contraseña: Contraseña del administrador.
   * Métodos:
     + gestionarParqueadero(): Gestiona los espacios del parqueadero.
     + generarReporte(): Genera reportes sobre la utilización del parqueadero.
5. **Notificacion:**
   * Atributos:
     + id: Identificador único de la notificación.
     + mensaje: Mensaje de la notificación.
     + fechaHoraEnvio: Fecha y hora de envío de la notificación.
   * Métodos:
     + enviarNotificacion(): Envía una notificación al usuario.

**Diagrama de PERT**

El diagrama de PERT (Program Evaluation Review Technique) es una herramienta de gestión de proyectos que permite planificar, programar y coordinar las tareas.

**Tareas Clave:**

1. **Definición de Requisitos:**
   * Recopilación y análisis de requisitos.
   * Aprobación de requisitos.
2. **Diseño de la Aplicación:**
   * Diseño de la arquitectura del sistema.
   * Diseño de la interfaz de usuario.
3. **Desarrollo de la Aplicación:**
   * Implementación de funcionalidades de usuario.
   * Implementación de funcionalidades de administrador.
   * Integración con el sistema de autenticación institucional.
4. **Pruebas y Validación:**
   * Pruebas unitarias.
   * Pruebas de integración.
   * Pruebas de aceptación por usuarios.
5. **Despliegue:**
   * Despliegue en el entorno de producción.
   * Formación a usuarios y administradores.
6. **Mantenimiento:**
   * Monitoreo y resolución de problemas.
   * Actualizaciones y mejoras continuas.



**Diagrama de Gantt**

El diagrama de Gantt es una herramienta de gestión de proyectos que muestra el cronograma de las tareas a lo largo del tiempo.

**Fases del Proyecto:**

1. **Definición de Requisitos**
2. **Diseño de la Aplicación**
3. **Desarrollo de la Aplicación**
4. **Pruebas y Validación**
5. **Despliegue**
6. **Mantenimiento**

Interfaz de usuario gráfica

Descripción generada automáticamente

**Cronograma:**

1. **Definición de Requisitos (2 semanas)**
   * Semana 1 y 2: Recopilación, análisis y aprobación de requisitos.
2. **Diseño de la Aplicación (2 semanas)**
   * Semana 3 y 4: Diseño de la arquitectura del sistema y la interfaz de usuario.
3. **Desarrollo de la Aplicación (6 semanas)**
   * Semana 5 a 10: Implementación de funcionalidades de usuario, administrador y la integración con el sistema de autenticación institucional.
4. **Pruebas y Validación (4 semanas)**
   * Semana 9 a 12: Pruebas unitarias, de integración y de aceptación por usuarios.
5. **Despliegue (2 semanas)**
   * Semana 11 y 12: Despliegue en el entorno de producción y formación a usuarios y administradores.
6. **Mantenimiento (continuo a partir de la semana 12)**
   * A partir de la semana 12: Monitoreo, resolución de problemas y mejoras continuas.