| DUOC UC - ESCUELA DE INFORMÁTICA Y TELECOMUNICACIONES |
| --- |
| Especificación de Requisitos de Software |
| *Proyecto: Itinerario Inteligente* |
|  |
| **Revisión*: [2.0]*** |
| **[26/09/2025]** |

| Especificación de Requisitos según estándar de IEEE 830. |
| --- |

**Contenido**

[**Ficha del documento 3**](#_heading=h.ud5bp8xsroks)

[**1. Introducción 4**](#_heading=h.qk3huxy0llzi)

[1.1.](#_heading=h.og0x8rlcpggr) Propósito 4

[1.2.](#_heading=h.923zolvo0l9t) Ámbito del Sistema 4

[1.3.](#_heading=h.tdtwgir8iic) Definiciones, Acrónimos y Abreviaturas 4

[1.4.](#_heading=h.w0eoiifr6n64) Referencias 4

[1.5.](#_heading=h.96uvpmuyq6cx) Visión General del Documento 4

[**2.**](#_heading=h.r86yrswdot5u) **Descripción General 5**

[2.1.](#_heading=h.68akvcn7p7bf) Perspectiva del Producto 5

[2.2.](#_heading=h.cg920aexcrej) Funciones del Producto 5

[2.3.](#_heading=h.xb0vxixj2oc9) Características de los Usuarios 5

[2.4.](#_heading=h.vw45v5x7gws7) Restricciones 5

[2.5.](#_heading=h.nh7r0egnm9ik) Suposiciones y Dependencias 6

[2.6.](#_heading=h.q63jjfo1wqdo) Requisitos Futuros 6

[**3.**](#_heading=h.ir5n24hek3z2) **Requisitos Específicos 7**

[3.1](#_heading=h.8pn1mbv68gag) Requisitos comunes de las interfaces 8

[*3.1.1*](#_heading=h.ainp1pkengsq) *Interfaces de usuario 8*

[*3.1.2*](#_heading=h.t3d7z96i569v) *Interfaces de hardware 8*

[*3.1.3*](#_heading=h.lgiw75nnxd15) *Interfaces de software 8*

[*3.1.4*](#_heading=h.s0egvjczjhcd) *Interfaces de comunicación 8*

[3.2](#_heading=h.9a2mx4q04nj5) Requisitos funcionales 9

[3.3](#_heading=h.b68ps6fdzm13) Requisitos no funcionales 9

[*3.3.1*](#_heading=h.dflqnb205q0i) *Requisitos de rendimiento 9*

[*3.3.2*](#_heading=h.el2uamjdgu8n) *Seguridad 10*

[*3.3.3*](#_heading=h.jotfdn4ofglw) *Fiabilidad 10*

[*3.3.4*](#_heading=h.wzm2txbah1vn) *Disponibilidad 10*

[*3.3.5*](#_heading=h.l0bhu138dlq9) *Mantenibilidad 10*

[*3.3.6*](#_heading=h.zbda8e4jtco5) *Portabilidad 10*

[3.4](#_heading=h.iypraffgblov) Otros Requisitos 10

**1. Introducción**

La presente Especificación de Requisitos de Software (ERS) tiene como objetivo principal definir de manera detallada los requisitos funcionales y no funcionales necesarios para el desarrollo de la aplicación móvil de Itinerario Inteligente "TripGO" propuesto por nuestro grupo de estudiantes de Ingeniería Informática del Instituto Duoc UC, en Santiago, Chile. Este documento servirá como guía fundamental para el diseño, implementación y verificación del sistema, permitiendo una comprensión clara y completa de las necesidades y expectativas del cliente.

La iniciativa de implementar una aplicación móvil de itinerario surge como respuesta a los desafíos de organizar y hacer más eficientes las actividades del usuario promedio que tiene un día ocupado de actividades. Dadas las problemáticas de un usuario común a la falta de organización y eficiencia a la hora de crear el itinerario del día, nace TripGO para mejorar la eficiencia y organización de las actividades de un usuario.

En la sección de "Descripción General", se proporciona una visión general de la aplicación móvil, sus funciones y sus usuarios. A continuación, en la sección de "Requisitos Específicos", se detallan los requisitos funcionales y no funcionales, clasificados y organizados de manera precisa.

Es importante destacar que esta ERS se mantendrá actualizada a lo largo del ciclo de desarrollo del proyecto, reflejando cualquier cambio o adición de requisitos que pueda surgir durante el proceso. Además, servirá como punto de referencia para la validación y verificación de la aplicación móvil final, asegurando que cumple con las expectativas y necesidades de sus usuarios.

La colaboración y comunicación efectiva entre todas las partes involucradas en este proyecto serán fundamentales para lograr un sistema exitoso que contribuya a la eficiencia y organización del usuario promedio.

**1.1. Propósito**

El propósito de este documento de Especificación de Requisitos de Software (ERS) es definir y documentar los requisitos necesarios para el desarrollo de la aplicación móvil de itinerario, en Santiago de Chile. Este documento sirve como referencia para todas las partes involucradas en el proyecto, incluyendo desarrolladores, diseñadores, probadores y las partes interesadas.

**1.2. Ámbito de la app móvil**

**Nombre de la Futura App móvil:** La app móvil se denominará "TripGo".

**Descripción de lo que el Sistema Hace:** La app móvil "TripGo" permitirá a los usuarios registrados en la aplicación móvil crear el itinerario del día según los horarios de cada lugar teniendo preferencias por lugares en proximidad y el cálculo en el tiempo de transporte de un punto a otro para hacer la organización de las actividades de manera eficiente y conveniente. Los usuarios podrán realizar todo el proceso de crear el itinerario a través de una app móvil en línea, que incluirá una aplicación móvil y un sitio web para las empresas o negocios que quieran pagar una suscripción para poder aparecer en la barra de búsqueda como lugar preferido. Las principales funcionalidades de la app móvil incluirán:

* Registro de usuarios.
* Autenticación de usuarios.
* Georreferenciación para ubicar al usuario.
* Vista en el mapa de la ruta del itinerario.
* Aparición de locales en la barra de búsqueda como preferencias.
* Visualización de información del día en la pantalla de inicio.
* Generación de informes y estadísticas de uso.
* Cálculo de tiempo de los recorridos.
* Calificación de usuarios y retroalimentación del servicio.

**Lo que la app móvil NO Hará:** La app móvil “TripGO” no realizará las siguientes funciones:

* No se realizará pagos de inicio en la app.
* No proporcionará recordatorios de actividades como en el calendario de google.
* No dará información extra que no tenga que ver con hacer itinerarios en lugares públicos.
* No almacenará información de tarjetas de crédito sin garantizar la seguridad de los datos.
* No funcionará sin conexión a internet; se requiere conectividad para mapas y datos de rutas.
* No incluirá funciones de mensajería entre usuarios.
* No permitirá reservas o compras directas en los locales; la app solo mostrará la información de los lugares.
* No generará itinerarios para actividades privadas o eventos cerrados que no estén en la base de datos de locales públicos.
* No ofrecerá soporte para versiones antiguas de sistemas operativos que no sean compatibles con la app.

**Beneficios, Objetivos y Metas:** Los beneficios, objetivos y metas esperados con la implementación del sistema son:

* Fomentar el ahorro al hacer cálculos eficientes para llegar a un punto.
* Reducir la congestión del tráfico en la comuna.
* Disminuir la contaminación del aire.
* Facilitar el uso de la organización de horarios de manera eficiente.
* Proporcionar una solución de movilidad conveniente para los usuarios de la app móvil.
* Recopilar datos y estadísticas para una mejor planificación del sistema de transporte.
* Mejorar la saturación de lugares.

El sistema tiene como objetivo principal contribuir al desarrollo de una comuna más sostenible y amigable con el medio ambiente al promover el uso de la bicicleta como medio de transporte.

**1.3. Definiciones, Acrónimos y Abreviaturas**

**ERS:** Especificación de Requerimientos de Software.

* **Descripción:** Este término se refiere al documento actual que detalla los requisitos y especificaciones del software a desarrollar para el proyecto "TripGO".

**GPS:** Sistema de Posicionamiento Global.

* **Descripción:** El sistema de GPS se utilizará para rastrear la ubicación de los usuarios y los locales en tiempo real, lo que permite identificar dónde se moviliza el usuario y garantizar la seguridad del sistema.

**ISO/IEC:** Organización Internacional de Normalización/Comisión Electrotécnica Internacional.

* **Descripción:** ISO/IEC es un organismo internacional que establece normas y estándares relacionados con la calidad del software y la seguridad del desarrollo de software.

**Usuario:** Persona que ocupa los servicios de forma virtual en la aplicación móvil.

* **Descripción:** La persona que utiliza los servicios de forma virtual es un usuario fundamental para el desarrollo de la aplicación además del cumplimiento de los servicios dados.

**1.4. Referencias**

* **ISO/IEC 25000:2005:** Esta norma se titula "Software Engineering - Software Product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE." SQuaRE significa "Software Quality Requirements and Evaluation." Esta norma proporciona directrices y un marco para evaluar la calidad de los productos de software. Define un conjunto de características de calidad del software y proporciona una estructura para medir y evaluar esas características. Ayuda a asegurar que el software cumpla con los estándares de calidad deseados.
* **ISO/IEC 9126:** Esta norma se enfoca en la calidad del producto de software y se titula "Software Engineering - Product Quality." Proporciona un marco para evaluar la calidad del software desde varias perspectivas, como la funcionalidad, la confiabilidad, la usabilidad y otras. También define métricas y atributos que pueden utilizarse para medir la calidad del software.
* **ISO/IEC 14598:** Esta norma se titula "Information Technology - Software Product Evaluation." Está relacionada con la evaluación de productos de software y proporciona directrices y métodos para llevar a cabo evaluaciones de software de manera sistemática. Ayuda a las organizaciones a evaluar y mejorar la calidad de sus productos de software.
* **ISO/IEC 15408:** Esta norma se enfoca en la seguridad de la tecnología de la información y se titula "Information technology - Security techniques - Evaluation criteria for IT security." Es más conocida como el estándar Common Criteria. Proporciona un marco para la evaluación de la seguridad de los sistemas y productos de tecnología de la información. Define criterios de seguridad que se utilizan para evaluar y certificar la seguridad de un sistema o producto.

**1.5. Visión General del Documento**

**Área de Definición del Negocio:**

La aplicación móvil tendrá rango de dar servicio a toda la región metropolitana.

**Área de Especificación de Requisitos:**

El sistema debe permitir a los usuarios registrarse, realizar pagos con tarjetas bancarias, y proporcionar información de georreferenciación para localizar problemas de tráfico o problemas ambientales. Además, debe cumplir con los estándares de seguridad definidos por la Norma ISO/IEC 15408 y los atributos de calidad establecidos por la norma ISO/IEC 25000:2005.

**Formularios de Casos de Uso (Anexos):**

Como anexos a este documento, se proporcionan formularios de casos de uso que ofrecen una representación visual de las interacciones entre los usuarios y el sistema. Estos casos de uso ayudan a comprender mejor los flujos de trabajo y las funcionalidades clave del sistema.

**2. Descripción General**

El proyecto "TripGo" busca abordar el servicio de creación de itinerarios para personas que busquen optimizar los tiempos de ocio. La iniciativa promueve la mejora en cuanto a la organización y el tiempo de ocio para personas ocupadas. El sistema se basa en una aplicación móvil, ofreciendo la creación de itinerarios. El objetivo es mejorar el tiempo de ocio y la mejora de la organización de las actividades del usuario.

**2.1. Perspectiva del Producto**

El sistema "TripGo" es independiente y autónomo, con vínculos con otros sistemas. Interactúa con usuarios a través de aplicaciones móviles, aprovechando el hardware del dispositivo del usuario. Se conecta a servicios bancarios para pagos, pero no depende de otros sistemas complejos.

**2.2. Funciones del Producto**

Las funciones clave de "TripGo" incluyen:

* Registro de Usuarios.
* Mostrar información del clima y el tráfico de transporte público.
* Autenticación de Tarjetas Bancarias.

Creación de itinerarios.

* Buscar lugares donde ir.
* Agregar a la lista de itinerarios.
* Mostrar la ruta en el mapa.
* Usar IA como asistencia en la creación de itinerarios.

**2.3. Características de los Usuarios**

El sistema "TripGo" atenderá a un tipo de perfil de usuario con las siguientes características generales:

* Usuario Cliente:
  + Perfil variado sin requisitos técnicos específicos.
  + Responsabilidades: Utilización de la aplicación móvil para la creación de itinerarios.

**2.4. Restricciones**

* Interfaces con Otras Aplicaciones:
  + Integración segura y cifrada con servicios de validación de tarjetas bancarias para garantizar la confidencialidad de los datos financieros de los usuarios.
* Funciones de Control y Auditoría:
  + Implementación de controles de acceso y seguimiento de todas las actividades de los usuarios para garantizar la seguridad y privacidad de los datos.
* Consideraciones de Seguridad:
  + Implementación de medidas de seguridad robustas para proteger la información de los usuarios y garantizar transacciones seguras y confidenciales.

**2.5. Suposiciones y Dependencias**

El proyecto "TripGo" considera las siguientes suposiciones y dependencias:

* Cumplimiento Normativo: Se asume que las políticas de la aplicación se mantendrán constantes.
* Proveedores Bancarios: El sistema depende de servicios de validación de tarjetas bancarias.
* Tecnologías Externas: Requiere dispositivos móviles, GPS y conectividad a Internet.
* Uso Responsable: Se espera que los usuarios sigan las normas de tráfico y regulaciones de la aplicación.
* Mantenimiento Continuo: Necesitará mantenimiento y soporte a lo largo del tiempo.

**2.6. Requisitos Futuros**

Se considerarán futuras mejoras, como la expansión a otras áreas, integración con transporte público, mejoras en la experiencia del usuario, tecnología inteligente en la creación de itinerarios y programas de fidelización.

**3. Requisitos Específicos**

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describe comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Esta es la sección más larga e importante de la ERS. Deberán aplicarse los siguientes principios:

• El documento debería ser perfectamente legible por personas de muy distintas formaciones e intereses.

• Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.

• Todo requisito deberá ser unívocamente identificable mediante algún código o sistema de numeración adecuado.

• Lo ideal, aunque en la práctica no siempre realizable, es que los requisitos posean las siguientes características:

* **Corrección**: La ERS es correcta si y sólo si todo requisito que figura aquí (y que será implementado en el sistema) refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado.
* **No ambiguos**: Cada requisito tiene una sola interpretación. Para eliminar la ambigüedad inherente a los requisitos expresados en lenguaje natural, se deberán utilizar gráficos o notaciones formales. En el caso de utilizar términos que, habitualmente, poseen más de una interpretación, se definirán con precisión en el glosario.
* **Completos**: Todos los requisitos relevantes han sido incluidos en la ERS. Conviene incluir todas las posibles respuestas del sistema a los datos de entrada, tanto válidos como no válidos.
* **Consistentes**: Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorios no es implementable.
* **Clasificados**: Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por importancia (esenciales, condicionales u opcionales) o por estabilidad (cambios que se espera que afecten al requisito). Esto sirve, ante todo, para no emplear excesivos recursos en implementar requisitos no esenciales.
* **Verificables**: La ERS es verificable si y sólo si todos sus requisitos son verificables. Un requisito es verificable (testable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito. Un requisito ambiguo no es, en general, verificable. Expresiones como a veces, bien, adecuado, etc. Introducen ambigüedad en los requisitos. Requisitos como “en caso de accidente la nube tóxica no se extenderá más allá de 25 Km" no es verificable por el alto costo que conlleva.
* **Modificables**: La ERS es modificable si y sólo si se encuentra estructurada de forma que los cambios a los requisitos pueden realizarse de forma fácil, completa y consistente. La utilización de herramientas automáticas de gestión de requisitos facilita enormemente esta tarea.
* **Trazables**: La ERS es trazable si se conoce el origen de cada requisito y se facilita la referencia de cada requisito a los componentes del diseño y de la implementación. La trazabilidad hacia atrás indica el origen (documento, persona, etc.) de cada requisito. La trazabilidad hacia delante de un requisito R indica qué componentes del sistema son los que realizan el requisito R.

**3.1 Requisitos comunes de las interfaces**

**Entradas del Sistema:**

**Datos del Usuario:** El sistema debe aceptar datos de usuario como nombre, dirección, número de teléfono, dirección de correo electrónico, número de tarjeta de crédito y RUT (Rol Único Tributario). Estos datos son proporcionados por el usuario al registrarse en la plataforma.

**Ubicación del Usuario:** El sistema debe recibir la ubicación del usuario a través del sistema de georreferenciación (GPS) de su dispositivo móvil o navegador web. Esto se utiliza para mostrar al usuario los lugares que él requiera.

**Datos de Tarjeta de Crédito o débito :**Para el pago de suscripción, el usuario debe ingresar los datos de su tarjeta de crédito o débito, incluido el número de tarjeta, la fecha de vencimiento y el código de seguridad.

**Datos de lugares:** El usuario ingresa datos de los lugares para completar el itinerario

**Salidas del Sistema:**

**Confirmación de Itinerario creado:** Después de recibir una solicitud de arriendo, el sistema debe generar una confirmación que incluye detalles como el número de bicicleta asignada, la ubicación de desbloqueo y el tiempo estimado de uso. Esta confirmación se muestra al usuario.

**Registro de Transacciones:** El sistema debe mantener un registro de todas las transacciones, incluidos los detalles de los arriendos anteriores, como la hora de inicio y finalización, la tarifa cobrada y la ubicación de bloqueo.

**Muestra de información del día:** Cuando un usuario entra en la aplicación en la vista de inicio verá el clima en grados e información de los vehículos con prohibición de recorrido.

**Factura y Recibo:** Al pagar la suscripción de la cuenta, el sistema genera una factura detallada que muestra el costo total del arriendo y la información de la tarjeta de crédito utilizada. También emite un recibo para el usuario como comprobante de pago.

**Alertas y Notificaciones:** El sistema puede generar alertas y notificaciones para informar al usuario sobre eventos importantes, como problemas en rutas del Gran Santiago.

**3.1.1 Interfaces de usuario**

**Estilo y Diseño:**

**Diseño Minimalista y Amigable:** El diseño de la aplicación móvil debe ser minimalista y fácil de usar, con una apariencia amigable para los usuarios. Se deben evitar diseños complicados y abrumadores.

**Colores de la Marca:** Se utilizarán los colores de Google, que pueden incluir el Azul y el blanco para mantener la coherencia con la identidad visual de un itinerario.

**Distribución de la Interfaz de Usuario:**

**Menú-Inicio:** La interfaz de Inicio o Menú contendrá la api que el tiempo para dar información del clima, además, de web scraping de noticias sobre el transporte público.

**Itinerario:** En el área de Itinerario se creará el itinerario para poder visualizarlo en google maps.

**Perfil:** En esta interfaz se mostrará el perfil del usuario y el resto de configuraciones de la aplicación e información legal.

**Inicio de Sesión y Registro:**

**Registro de Usuarios:** Los nuevos usuarios podrán registrarse en el sistema completando un formulario que incluirá campos para ingresar su nombre,número de teléfono, RUT y datos de tarjeta de crédito.

**Proceso de Suscripción:**

* **Pago de suscripción:** Cuando un usuario elija arrendar una bicicleta, se le dirigirá a un formulario donde especificará la hora de inicio y el tiempo de arriendo deseado. El sistema calculará automáticamente el costo del arriendo en función de esta información.

**3.1.2 Interfaces de hardware**

**Interfaz con Dispositivos Táctiles Móviles:**

**Compatibilidad con Dispositivos Móviles:** El sistema debe ser compatible con dispositivos móviles que admitan pantallas táctiles, como teléfonos inteligentes y tabletas.

**Interfaz Táctil Intuitiva:** El sistema debe ofrecer una interfaz táctil intuitiva que permita a los usuarios interactuar fácilmente con las funciones del sistema utilizando gestos táctiles comunes, como tocar, deslizar y pellizcar.

**Optimización para Pantallas Pequeñas:** La interfaz debe estar optimizada para pantallas más pequeñas, asegurando que todos los elementos de la aplicación sean legibles y accesibles en dispositivos móviles.

**Compatibilidad con Diversos Sistemas Operativos:** El sistema debe ser compatible con diversos sistemas operativos móviles, como Android e iOS, para garantizar que los usuarios puedan acceder al sistema independientemente del dispositivo que utilicen.

**Resolución de Pantalla Adecuada:** El sistema debe ajustar automáticamente la resolución y el diseño de la interfaz para adaptarse a la pantalla del dispositivo móvil en uso, garantizando una experiencia de usuario consistente y adecuada.

**Características de Configuración:**

**Configuración de GPS:** El sistema debe poder acceder y utilizar las capacidades de GPS de los dispositivos móviles para rastrear la ubicación del usuario y mostrar la disponibilidad de bicicletas en el mapa.

**Acceso a la Cámara:** En caso de ser necesario, el sistema podría requerir acceso a la cámara del dispositivo móvil para permitir a los usuarios escanear códigos QR o tomar fotos relacionadas con el arriendo de bicicletas.

**3.1.3 Interfaces de software**

**Integración con Sistemas de Pago:**

* **Descripción del Producto Software Utilizado:** El sistema debe integrarse con un sistema de procesamiento de pagos en línea. Puede utilizarse un servicio de pago en línea ampliamente reconocido, como Stripe o PayPal, para facilitar transacciones seguras con tarjetas de crédito o débito.
* **Propósito del Interfaz:** El propósito de esta interfaz es permitir que los usuarios realicen pagos en línea de manera segura y eficiente para la suscripción.
* **Definición del Interfaz:** La interfaz debe incluir un formulario de pago en línea en el que los usuarios puedan ingresar los detalles de su tarjeta de crédito, incluido el número de tarjeta, la fecha de vencimiento y el código de seguridad (CVV). La información de pago se debe transmitir de forma segura al sistema de procesamiento de pagos externo a través de una conexión segura (por ejemplo, HTTPS) para su procesamiento. El formato de los datos debe cumplir con los estándares de seguridad y privacidad aplicables.

**Integración con Servicios de GPS:**

* **Descripción del Producto Software Utilizado:** El sistema debe integrarse con servicios de geolocalización basados en GPS, como Google Maps, entre otros.
* **Propósito del Interfaz:** El propósito de esta interfaz es permitir que el sistema acceda a la ubicación del usuario y muestre información en tiempo real sobre negocios y puntos de interés en ubicaciones cercanas.
* **Definición del Interfaz:** El interfaz debe permitir que el sistema obtenga datos de ubicación del dispositivo del usuario y los utilice para mostrar un mapa interactivo que muestre las ubicaciones a las que quiera ir el usuario. El formato de los datos debe ser compatible con el estándar de intercambio de datos geoespaciales utilizado por el servicio de GPS.

**3.1.4 Interfaces de comunicación**

**Comunicación con Aplicaciones Móviles y Páginas Web:**

* Requisitos de la Interfaz de Comunicación: El sistema debe ser capaz de comunicarse de manera bidireccional con las aplicaciones móviles y las páginas web a través de protocolos estándar de comunicación web. Esto permite que los clientes realicen transacciones, desde su navegador web.
* Protocolos de Comunicación: Para la comunicación con aplicaciones móviles y páginas web, se deben utilizar protocolos web estándar, como HTTP (Hypertext Transfer Protocol) o HTTPS (HTTP Secure) para garantizar la seguridad de las transacciones y la privacidad de los datos del usuario.

**Comunicación con Servicios de GPS:**

* Requisitos de la Interfaz de Comunicación: El sistema debe ser capaz de comunicarse con los servicios de GPS para obtener datos de ubicación en tiempo real y mostrar información sobre la disponibilidad de lugares en ubicaciones cercanas.
* Protocolos de Comunicación: La comunicación con los servicios de GPS generalmente se realiza a través de protocolos de servicios web estándar, como REST (Representational State Transfer) o SOAP (Simple Object Access Protocol), dependiendo de la implementación específica de los servicios de GPS. Además, se debe garantizar que la comunicación sea segura y esté cifrada.

**Comunicación con el Sistema de Procesamiento de Pagos:**

* Requisitos de la Interfaz de Comunicación: El sistema debe ser capaz de comunicarse de manera segura con el sistema de procesamiento de pagos para autorizar y procesar transacciones de pago con tarjeta de crédito.
* Protocolos de Comunicación: La comunicación con el sistema de procesamiento de pagos se realiza generalmente a través de API (Interfaz de Programación de Aplicaciones) proporcionadas por el proveedor de servicios de pago. Los protocolos de comunicación pueden incluir HTTPS para garantizar la seguridad de las transacciones y el cumplimiento de los estándares de seguridad de la industria de tarjetas de pago (PCI DSS).

**3.2 Requisitos funcionales**

**Requisito Funcional 1: Registro de Usuarios**

* Descripción: El sistema debe permitir a los usuarios registrarse en la plataforma proporcionando información personal y financiera, garantizando la seguridad de los datos.
* Actor Principal: Usuario
* Precondiciones: El usuario debe acceder a la plataforma de registro.
* Flujo de Eventos:

1. El usuario accede desde la aplicación móvil.
2. El usuario completa un formulario de registro que incluye campos para nombre, dirección, número de teléfono y datos de tarjeta de crédito o débito (opcional).
3. El sistema verifica la validez de los datos ingresados, incluyendo la autenticación de la tarjeta de crédito o débito.
4. Si los datos son válidos, el sistema registra al usuario en la base de datos y genera un ID único para el usuario.
5. El sistema muestra un mensaje de confirmación de registro.

* Postcondiciones: El usuario queda registrado en el sistema y puede acceder a las funcionalidades adicionales si ingresó tarjeta de crédito o débito.

**Requisito Funcional 2: Creación de Itinerario**

* **Descripción:** Los usuarios registrados deben poder crear Itinerario desde la plataforma móvil, verificando la disponibilidad y asegurando una experiencia de usuario fluida.
* **Actor Principal:** Usuario
* **Precondiciones:** El usuario debe haber descargado la aplicación para usar los servicios básicos de la aplicación.
* **Flujo de Eventos:**

1. El usuario inicia sesión en la plataforma.
2. El usuario se dirige al módulo de itinerario.
3. El usuario presiona el botón “+” para crear un nuevo itinerario.
4. El usuario escribe los lugares que debe visitar, ya sea de forma específica (eg. Banco de Chile) o de forma no específica (eg. Supermercado).
5. El sistema verifica la disponibilidad de los destinos utilizando información pública y la ubicación del usuario.
6. El sistema genera un itinerario en base a las especificaciones y preferencias del usuario.
7. El usuario presiona el botón “Ver Ruta” para abrir el mapa donde se observan los destinos y su ruta a seguir y por diferentes medios de transporte.
8. (Opcional) El usuario presiona el botón “Volver al itinerario” y modifica uno o más destinos.
9. (Opcional) El sistema recalcula el itinerario y genera uno nuevo con una nueva ruta.

* **Postcondiciones:** El usuario tiene un itinerario completo, claro y bien definido y puede comenzar su día.

**Requisito Funcional 3: Registro de subscripción**

* **Descripción:** Los usuarios deben poder acceder a una subscripción que les permita acceder a herramientas o beneficios que apoyen o mejore su experiencia de usuario en la creación de itinerarios.
* **Actor Principal:** Usuario
* **Precondiciones:** El usuario debe tener una cuenta registrada.
* **Flujo de Eventos:**

1. El usuario inicia sesión en la plataforma.
2. El usuario selecciona la opción “Mi perfil”.
3. El usuario selecciona la opción “Mi subscripción”.
4. El sistema, en caso de que el usuario no posea una suscripción activa, mostrará los diferentes niveles o duración de suscripción disponibles con los precios en la moneda local (CLP o Peso Chileno) y un botón de pago.
5. El usuario selecciona el tipo de suscripción, donde el sistema le redirigirá al portal de pago, donde el usuario deberá llenar los datos solicitados.
6. Una vez realizado el pago, el sistema permitirá al usuario usar las funcionalidades avanzadas del sistema mientras la subscripción se encuentre activa.
7. La pestaña de “Mi subscripción” ahora mostrará la información relacionada a la suscripción activa y habrá un botón “Cancelar suscripción” para su cancelación prematura.

* **Postcondiciones:** El usuario podrá beneficiarse de las funcionalidades avanzadas de la aplicación, y este podrá cancelar en cualquier momento dicha suscripción.

**Categorías adicionales:**

* **Gestión de Cuentas de Usuario:**

1. Registro de usuario.
2. Inicio de sesión.
3. Actualización de datos de usuario.
4. Cambio de contraseña.

* **Itinerarios e información relevante:**

1. Creación de itinerarios personalizados.
2. Asistencia por IA.
3. Visualización de ruta, tiempos y método de transporte.
4. Visualización de clima, restricción de vehículos catalíticos.

* **Seguridad y Privacidad:**

1. Protección de datos personales y financieros.
2. Autenticación de usuarios.
3. Prevención de fraudes y abusos.

* **Gestión de Pagos:**

1. Procesamiento de pagos con tarjeta de crédito y débito.
2. Historial de transacciones.
3. Facturación y emisión de recibos.

* **Notificaciones y Comunicación:**

1. Notificaciones push o por correo electrónico.
2. Comunicación entre usuarios y soporte técnico.
3. Comunicación de políticas y actualizaciones del servicio.

* **Informes y Estadísticas:**

1. Generación de informes de uso.
2. Estadísticas de uso.
3. Análisis de tendencias y patrones de uso.

* **Interacción con el Soporte Técnico:**

1. Solicitud de ayuda o asistencia.
2. Resolución de problemas técnicos.

* **Integración con Sistemas Externos:**

1. Integración con sistemas de navegación GPS.
2. Integración con Transantiago / Scrapping

**3.3 Requisitos no funcionales**

**3.3.1 Requisitos de rendimiento**

1.- Número de usuarios simultáneos:

* + Se espera que el sistema pueda manejar al menos un número específico de usuarios conectados simultáneamente durante las horas pico.

2.- Tiempo de respuesta:

* + El 95% de los itinerarios deben generarse en 5 segundos o menos.
  + El tiempo de carga de la aplicación en dispositivos móviles no debe superar los 3 segundos.

3.- Número de transacciones por segundo:

* + El sistema debe ser capaz de manejar al menos [número específico] de transacciones por suscripción.

4.- Escalabilidad:

* + El sistema debe ser escalable horizontalmente para poder aumentar la capacidad de forma dinámica en función de la demanda. Se debe poder agregar [número específico] de servidores adicionales automáticamente cuando sea necesario.

5.- Pruebas de carga:

* + Se deben realizar pruebas de carga regulares para verificar que el sistema cumple con los requisitos de rendimiento establecidos. Estas pruebas deben simular condiciones de carga máxima y medir el tiempo de respuesta, la capacidad y la estabilidad del sistema.

6.- Alta disponibilidad:

* + El sistema debe tener una disponibilidad del 99.9% durante el horario de servicio regular.

7.- Tolerancia a fallos:

* + El sistema debe ser capaz de recuperarse automáticamente de fallos menores sin afectar negativamente a la experiencia del usuario. Por ejemplo, si un servidor falla, el sistema debe redirigir automáticamente las solicitudes a servidores en funcionamiento.

8.- Optimización de datos:

* + Las consultas a la base de datos deben estar optimizadas para garantizar un tiempo de respuesta rápido, incluso con grandes volúmenes de datos. Por ejemplo, las consultas de búsqueda de estaciones de bicicletas cercanas deben ser eficientes.

9.- Seguridad bajo carga:

* + El sistema debe mantener los niveles de seguridad incluso bajo una carga máxima. Por ejemplo, las transacciones financieras deben seguir siendo seguras y protegidas.

10.- Capacidad de cacheo:

* + El sistema debe implementar un mecanismo eficiente de almacenamiento en caché para reducir la carga en el servidor y acelerar las respuestas a las solicitudes frecuentes.

**3.3.2 Seguridad**

1.- Autenticacion y autorizacion:

* + Los usuarios deben autenticarse antes de acceder a la aplicación.
  + Se deben implementar controles de autorización para garantizar que los usuarios solo tengan acceso a las funcionalidades que les corresponden.

2.- Cifrado de datos:

* + Empleo de Técnicas Criptográficas: Todos los datos sensibles, como información de usuarios y transacciones financieras, deben ser cifrados en reposo y en tránsito utilizando algoritmos de cifrado seguros.

3.- Registro de actividad (Logs):

* + Registrar Eventos de Actividad: La aplicación debe mantener registros detallados de actividad (logs) que registran eventos significativos, como inicios de sesión, transacciones y cambios en la configuración.

4.- Monitorización de seguridad:

* + Implementar un sistema de monitorización de seguridad que alerte a los administradores sobre actividades sospechosas o intentos de acceso no autorizados.

5.- Actualizaciones seguras:

* + Las actualizaciones de la aplicación deben ser seguras y autenticadas para evitar modificaciones maliciosas. Utilizar firmas digitales para verificar la autenticidad de las actualizaciones.

6.- Restricciones de comunicación:

* + Limitar la comunicación entre determinados módulos para reducir el riesgo de que un acceso no autorizado se propague a través de la aplicación.

7.- Comprobación de integridad:

* + Implementar comprobaciones de integridad de la aplicación para detectar modificaciones no autorizadas en el código fuente o en los archivos de configuración.

8.- Gestión de sesiones segura:

* + Asegurarse de que las sesiones de usuario sean seguras, utilizando tokens de sesión seguros y expiración de sesiones inactivas.

9.- Protección contra ataques comunes:

* + Implementar protección contra ataques comunes como SQL injection, cross-site scripting (XSS) y cross-site request forgery (CSRF) mediante filtrado de entrada y validación de datos.

10.- Pruebas de seguridad:

* + Realizar pruebas de seguridad regulares, como pruebas de penetración, para identificar posibles vulnerabilidades y debilidades de seguridad en la aplicación.

11.- Recuperación ante desastres:

* + Implementar un plan de recuperación ante desastres que permita restaurar rápidamente la funcionalidad en caso de un ataque o falla crítica.

12.- Educacion y concientizacion:

* + Proporcionar capacitación en seguridad a los desarrolladores y al personal de la empresa para aumentar la conciencia sobre las mejores prácticas de seguridad.

**3.3.3 Fiabilidad**

1.-MTBF (Mean Time Between Failures): Este enfoque mide la fiabilidad en términos del tiempo promedio que el sistema puede funcionar sin experimentar una falla crítica. En el contexto de una aplicación de generación de itinerarios, un MTBF deseable podría expresarse de la siguiente manera: "La aplicación debe tener un MTBF de al menos 1000 horas de uso continuo antes de experimentar una falla crítica". Esto significa que se espera que la aplicación funcione sin problemas durante al menos 1000 horas de uso antes de que ocurra una falla significativa.

2.-Total de incidentes permisibles: En lugar de medir la fiabilidad en función del tiempo, este enfoque se centra en la cantidad de incidentes o errores permitidos en un período determinado. Para una aplicación de generación de itinerarios, podría expresarse de la siguiente manera: "La aplicación debe permitir un máximo de 10 incidentes críticos por cada 1000 itinerarios generados". Esto significa que, en promedio, no debe haber más de 10 problemas graves por cada 1000 itinerarios generados.

**3.3.4 Disponibilidad**

1. Disponibilidad del sistema: Esto se refiere al tiempo total durante el cual el sistema en su conjunto debe estar disponible para su uso. En muchos casos, se busca una disponibilidad del 99% o más. Esto significa que la aplicación debería estar disponible al menos el 99% del tiempo.

2. Disponibilidad de la aplicación móvil: Esto se refiere específicamente a la disponibilidad de la aplicación móvil en los dispositivos de los usuarios. Generalmente, se busca una disponibilidad del 99% o más para la aplicación en sí.

3. Disponibilidad de los servicios backend: Muchas aplicaciones móviles dependen de servicios backend para funcionar correctamente, como bases de datos, servidores web, sistemas de autenticación, etc. Estos servicios también deben tener un alto nivel de disponibilidad, con un objetivo similar al de la aplicación móvil, es decir, al menos un 99% de disponibilidad.

4. Tiempo de respuesta: Además de la disponibilidad, es importante establecer objetivos para el tiempo de respuesta de la aplicación. Los usuarios esperan que la aplicación responda de manera rápida y eficiente. Por ejemplo, puede definirse que la aplicación debe responder a las solicitudes de los usuarios en menos de 2 segundos en un cierto porcentaje de casos (por ejemplo, el 95% de las veces).

5. Tiempo de recuperación: En caso de que ocurran interrupciones o fallas en la aplicación, es importante tener un tiempo objetivo para la recuperación. Esto se refiere al tiempo que se necesita para restaurar la disponibilidad después de una interrupción. Dependiendo de la crítica de la aplicación, este tiempo puede variar, pero generalmente se busca que sea lo más rápido posible.

6. Plan de continuidad del negocio: Además de la disponibilidad técnica, es importante tener un plan de continuidad del negocio en caso de fallas graves. Esto puede incluir respaldos de datos, sistemas de recuperación de desastres y planes de comunicación con los usuarios en caso de interrupciones prolongadas.

**3.3.5 Mantenibilidad**

1.- Identificación del tipo de mantenimiento necesario:

a.- Mantenimiento correctivo: Se realiza para corregir errores o problemas identificados en el sistema, como fallos o errores de funcionamiento. Este tipo de mantenimiento se realiza cuando surge una incidencia.

b.- Mantenimiento preventivo: Se lleva a cabo de manera planificada y regular para evitar que ocurran problemas en el sistema, como fallos o errores de funcionamiento. Este tipo de mantenimiento se realiza cuando surge una incidencia.

c.- Mantenimiento adaptativo: Se realiza para adaptar el sistema a cambios en el entorno tecnológico o en los requisitos del usuario. Puede incluir actualizaciones de hardware o software.

d.- Mantenimiento correctivo: Se enfoca en mejorar la funcionalidad, rendimiento o la usabilidad del sistema sin corregir problemas específicos. Esto puede incluir la optimización de código o mejoras en la interfaz de usuario.

2.- Especificación de quién debe debe realizar las tareas de mantenimiento :

a.- Usuario finales: Los usuarios pueden realizar tareas de mantenimiento simples, como actualizar su información de perfil, cambiar contraseñas o reportar errores.

b.- Equipo de desarrollo: Los desarrolladores son responsables de mantener y mejorar el sistema. Esto incluye la corrección de errores, actualizaciones de software, y mejoras en la funcionalidad.

c.- Equipo de TI: El equipo de tecnología de la información puede encargarse de tareas de mantenimiento preventivo, como aplicar parches de seguridad y mantener la estructura subyacente.

d.- Administradores del sistema: Estos se encargan de tareas de mantenimiento a nivel de servidor, como la administración de base de datos y la gestión de recursos.

3.- Especificación de cuándo debe realizarse las tareas de mantenimiento:

a.- Generación de estadísticas semanales: Se deben generar estadísticas de acceso a la aplicación móvil de itinerarios inteligentes al final de cada semana. Esto permitirá evaluar el rendimiento semanal y tomar decisiones basadas en datos.

b.- Generación de estadísticas mensuales: La estadísticas mensuales pueden ayudar a obtener una vista más amplia de rendimiento a lo largo del tiempo. Pueden generarse al final de cada mes.

c.- Mantenimiento Preventivo: Las actualizaciones de la aplicación, parches de seguridad y pruebas de rendimiento pueden programarse periódicamente para mantener el sistema funcionando sin problemas.

**3.3.6 Portabilidad**

1.- Arquitectura de servidor versátil:

* + Porcentaje de Componentes Dependientes del Servidor: Mantener la mayor parte de la lógica de negocio y funcionalidad en el servidor puede aumentar la portabilidad. Intenta minimizar la lógica específica de la plataforma móvil.

2.- Código independiente del servidor:

* + Porcentaje de Código Dependiente del Servidor: Reduce la dependencia del servidor en la medida de lo posible, manteniendo el código del cliente móvil independiente y enfocado en la presentación y la interacción del usuario.

3.- Lenguaje de programación multiplataforma:

* + Uso de un Lenguaje Multiplataforma: Utiliza un lenguaje de programación que sea conocido por su portabilidad, como JavaScript (para aplicaciones web móviles), Dart (para Flutter), o Kotlin/Java (para Android). Estos lenguajes tienden a funcionar bien en múltiples plataformas.

4.- Framework de desarrollo multiplataforma:

* + Uso de un Framework Multiplataforma: Considera el uso de frameworks como React Native, Flutter o Xamarin, que permiten desarrollar aplicaciones móviles para varias plataformas utilizando un solo código base.

5.- APIs y estándares abiertos:

* + Adherencia a Estándares Abiertos: Utiliza estándares y protocolos abiertos para la comunicación entre el cliente móvil y el servidor, como RESTful API, GraphQL o WebSocket. Esto facilitará la integración con diferentes sistemas y plataformas.

6.- Diseño responsivo y adaptativo:

* + Diseño Responsivo: Diseña la interfaz de usuario de manera que se adapte automáticamente a diferentes tamaños de pantalla y orientaciones, lo que facilita la transición a diferentes dispositivos móviles y tamaños de pantalla.

7.- Gestión de recursos externos:

* + Minimización de Dependencias Externas: Limita la dependencia de bibliotecas externas específicas de la plataforma. Siempre que sea posible, utiliza bibliotecas y herramientas que sean compatibles con múltiples plataformas.

8.- Pruebas cruzadas rigurosas:

* + Pruebas en Diferentes Plataformas: Realiza pruebas exhaustivas en diversas plataformas y entornos para garantizar que la aplicación funcione de manera coherente en todas partes.

9.- Documentación detallada:

* + Documentación Extensa: Proporciona documentación completa y detallada sobre la configuración, requisitos y dependencias del software, lo que facilitará la implementación en diferentes entornos.

10.- Seguridad y cumplimiento de regulaciones

* + Asegúrate de que el software cumple con las regulaciones de seguridad y privacidad en todas las plataformas y entornos a los que pueda ser trasladado.