1. Product.

Product Description. El carpooling es una práctica de viaje compartido donde varias personas se organizan para utilizar un mismo vehículo privado para ir a lugares similares, ya sea en el día a día, como al trabajo o a la universidad, o en trayectos más largos entre ciudades.

UADYCAR es un servicio de carpooling diseñado para los estudiantes de la Facultad de Matemáticas (FMAT) que vivan en la misma zona y necesiten trasladarse hacia ella o regresar de la misma. Esto consiste en que los alumnos que cuenten con vehículos propios puedan ponerse de acuerdo con otros estudiantes de la facultad que vivan en la misma zona para irse y regresar juntos

Users/Client. Los alumnos de la FMAT tanto los que cuentan con carro particular, como los que no cuenten con vehículo y necesiten transporte.

Propuesta de valor. Nuestro proyecto no solo busca que los alumnos tengan una manera más rápida, eficaz y sobre todo segura de moverse de ida y vuelta de la facultad, sino que contribuye a reducir el tráfico en la ciudad de Mérida, así como disminuir la huella de carbono y fortalecer a la comunidad universitaria mediante una solución colaborativa que optimiza tiempo, recursos y confianza entre el alumnado.

1. Requisitos.

Requisitos funcionales

1. Solicitud del viaje: los usuarios (pasajeros) deberán poder solicitar un viaje en la aplicación.
2. Abordar viajes: Los pasajeros solicitaran al conductor llegar hasta un punto común, para no desviar tanto la ruta del viaje.
3. Inicio de sesión: los usuarios deberán poder ingresar a la APP de diferentes maneras. (Correo electrónico, número de celular, contraseña, etc.). Además de poder cerrar la sesión de forma segura.
4. Mostrar ruta: Los pasajeros al iniciar el viaje deberán poder ver su ubicación hasta el destino en tiempo real.
5. Cancelación del viaje: Los pasajeros deberán tener la opción de cancelar el viaje, bajarse en una zona segura y así no llegar hasta el destino si así lo requieren.
6. Remuneración disponible: A pesar de ser los viajes gratis para los pasajeros, el conductor deberá tener la opción de recibir una remuneración por su servicio, esto de forma voluntaria por los pasajeros.
7. Forma de pago: los pasajeros que quieran compensar el servicio podrán hacerlo de la forma más rápida y posible como transferencias bancarias, paypal o efectivo.
8. Servicio compartido: Más de un usuario puede acceder al viaje para el mismo vehículo, siempre y cuando no sea un desvió largo de la ruta y sea este un punto común para varias personas del mismo viaje.

Non-Functional Requirements

Product Requirements.

Efficiency Requirements:

* Performance**:** Ability to handle multiple users simultaneously without performance loss.
* User location: The location on the map must update in real time with a maximum delay of 10 seconds.
* Portability: It must be compatible with the most widely used mobile operating systems (iOS and Android).
* Storage: The application must have a total size of less than 100 MB.

Security Requirements:

* Driver safety: Allow drivers to record trips for their safety in case of any incident or accident.
* User PIN: Require all users to enter a PIN in order to start a trip.
* Privacy: Location data must only be shared when the user is on an active trip.

Requerimientos de usabilidad

* Usability: The interface must be intuitive so that a new user can easily create a trip.
* Ease of use: Passengers will not require any training to use the application’s features.

External Requirements

Legal Requirements:

* Car license plates: Require drivers to provide their car’s license plates at the time of registration in the application.
* Driver’s license: Drivers must have a valid driver’s license in order to provide the service.
* Protection/Security: Require official identification (INE or driver’s license) from users when registering in the application.

Regulatory requirements:

* Car insurance: Drivers providing the transportation service are required to have valid car insurance.

Organizational Requirements

Development requirements:

1. The code will be developed using Kotlin, an open-source programming language for application development.
2. Visual Studio Code will be used to efficiently test and fix the code.
3. We will store all code in a GitHub repository.
4. Only allow automobiles: For greater user safety, drivers may only use automobiles to provide the service.
5. Data deletion: Allow users to delete their account and all associated data.
6. Route storage: The application may store information about vehicle routes.

Priorización.

Artefactos.

**Caso de Uso: Solicitud de Viaje**

**Descripción general:** El caso de uso describe cómo un usuario interactúa con la aplicación para solicitar un viaje a un destino específico. El proceso incluye desde la selección de destino hasta la confirmación del viaje y la llegada del conductor.

**Actor Principal:** Usuario (Persona que solicita el viaje).

**Sistema de aplicación**: Aplicación UADYCAR.

**Objetivo en Contexto:** Solicitar un viaje al destino requerido.

**Precondiciones:**

* El usuario debe tener instalada la aplicación UADYCAR.
* El usuario debe tener acceso a internet.
* El usuario debe tener una cuenta activa en la aplicación UADYCAR, estar dispuesto a ingresar con ella o a crear una.
* El sistema se ha preparado para introducir un destino y arrojar las opciones de viajes publicados por los conductores.

**Disparador:** El usuario requiere dirigirse a un lugar en específico, necesita transporte y abre la aplicación para solicitar un viaje.

**Planteamiento:**

1. El usuario (pasajero) abre la aplicación desde su dispositivo móvil. Si no tiene su sesión iniciada, ingresa su información de usuario (nombre o correo electrónico y contraseña) y accede a su cuenta.
2. El usuario ingresa el destino al que quiere ir. El destino ingresado por el usuario se compara con los destinos ofrecido por los conductores en la aplicación. Si el destino del usuario coincide con uno o varios de los destinos ofrecidos por los conductores, entonces la aplicación los presenta como opciones de viaje disponibles; las opciones de viaje muestran quién es el conductor, cuál es su destino, el punto de reunión, la hora de salida y el número de usuarios con los que se compartirá el viaje. Si no, la pantalla muestra que no hay opciones disponibles.
3. El usuario selecciona la opción que le resulte más conveniente y confirma la solicitud de viaje. El sistema envía la solicitud de viaje al conductor, quién decide si la acepta o la rechaza.
4. El usuario recibe una notificación de que el viaje ha sido aceptado, junto con la información del conductor (nombre, foto, número de matrícula, modelo y características del vehículo) y la de los otros pasajeros con los que se compartirá el viaje (si los hay).
5. El usuario se dirige al punto de reunión establecido y espera a que el conductor llegue. A medida que el conductor se aproxima, la aplicación muestra su ubicación en tiempo real en el mapa, así como el tiempo y hora estimados de llegada.
6. Una vez que el usuario llega a su destino, el conductor finaliza el viaje. Si así lo decide, el usuario puede otorgar una compensación por el servicio pagando a través de la aplicación o en efectivo.
7. Con el viaje ya finalizado, la aplicación le pide al usuario calificar al conductor y el servicio. El usuario puede compartir sus comentarios sobre la experiencia.

**Escenario:**

1. Usuario: Abre la aplicación desde su dispositivo móvil.
2. Usuario: Ingresa el destino al que quiere ir.
3. Usuario: Selecciona la opción de viaje de su preferencia y confirma la solicitud.
4. Usuario: Recibe la notificación de “servicio aceptado”.
5. Usuario: Va al punto de reunión y espera la llegada del conductor.
6. Usuario: Paga al conductor (Opcional).
7. Usuario: Califica al conductor y el servicio.

**Excepciones:**

* Si el usuario no ha iniciado sesión: debe acceder a su cuenta antes de continuar.
* Si el usuario no tiene acceso a internet: la aplicación mostrará un mensaje de error y le pedirá que se conecte antes de continuar con la solicitud.
* Si el conductor no acepta la solicitud de viaje: el sistema notificará al usuario.
* Si el usuario cancela el viaje antes de que el conductor llegue: se le muestra una notificación.
* Si el pago a través de la aplicación falla: el sistema notifica al usuario y le pide que elija otro método de pago.

**Prioridad:** Esencial, ya que es la funcionalidad base sobre la cual se construye el servicio ofrecido por la aplicación.

**Cuando estará disponible:** En todo momento, ya que los usuarios pueden requerir el servicio en cualquier hora del día.

**Frecuencia de uso:** Alta, pues los usuarios pueden solicitar un viaje múltiples veces al día, dependiendo de sus necesidades, en especial durante las horas pico.

**Canal para el actor:** A través de la interfaz de la aplicación.

**Actores secundarios:** Conductor (Persona que acepta o rechaza la solicitud de viaje).

**Canales para actores secundarios:** A través de la interfaz de la aplicación para conductores.

**Aspectos Pendientes:**