
Historias de usuario

Proyecto: FluxLane

Sistema web y móvil híbrido par recolección muestral de datos sobre flujo vehicular en la zona de regeneración urbana de la ciudad de Loja.

Revisión 1.0
Fecha: 16/12/2018





HISTORIAS DE USUARIO

El sistema FluxLane se encuentra compuesto por tres aplicaciones: una Api REST; una aplicación móvil híbrida; y, una aplicación web. Los 3 subsistemas interactúan entre sí. La Api REST permite una separación entre el clásico enfoque cliente/servidor, por lo que es accesible desde cualquier otro sistema. La aplicación web está destinada a usuarios con un rol administrador. El objetivo de esta web es ofrecer un sistema completo para la gestión de rutas y usuarios. La aplicación móvil realiza la función de recolector de información, es decir, desde la web se establecen las rutas y la aplicación móvil será la encargada de obtener los datos y enviarlos a la Api, para su posterior gestión.

A continuación, se detallarán las historias de usuario asociadas a cada aplicación.

API REST FluxLane

Las historias de usuario están divididas en distintos módulos. Éstos se definen a continuación:

MODULO	ID	COLOR
Recurso ruta	RR	Yellow
Recurso usuario	RU	Blue
Recurso subpuntos	RS	Red
Recurso clientes	RC	Green

Las historias de usuario son construidas de acuerdo al formato indicado por la metodóloga Mobile-D con modificaciones para el caso de estudio, para ello en la siguiente figura mostramos la nomenclatura usada.

NOMENCLATURA			
TIPO	API	Arreglo	Mejora
DIFICULTAD	Fácil	Moderada	Difícil
PRIORIDAD	Baja	Normal	Alta
ESFUERZO	Bajo	Normal	Alto

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RR-01	API	MODERADA	MODERADO	NORMAL
Descripción: El usuario debe poder crear una nueva ruta mediante el envío una petición POST cuyo cuerpo sea el Modelo Ruta.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición POST.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ El cuerpo de la petición contiene un Modelo de Ruta válido.○ Se almacena en la base de datos la nueva ruta.○ La base de datos retorna un mensaje de éxito y un ID único para la ruta.				



<ul style="list-style-type: none">○ Se responde la petición con un código 201, junto con el ID de la nueva ruta. <ul style="list-style-type: none">● Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si el cuerpo de la petición, es decir, el Modelo Ruta no es válido, se responde con un código 400.
Observaciones: <ul style="list-style-type: none">● El formato de respuesta es JSON.● La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.● El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RR-02	API	MODERADA	MODERADO	NORMAL
Descripción: El usuario debe poder modificar una ruta existente mediante el envío una petición PUT cuyo cuerpo sea el Modelo Ruta.				
Criterios de Aceptación: <ul style="list-style-type: none">● Éxito: Petición PUT.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ La URI de la petición contiene un ID de ruta válido.○ El cuerpo de la petición contiene un Modelo de Ruta válido.○ Se actualiza en la base de datos la ruta.○ La base de datos retorna un mensaje de éxito.○ Se responde la petición con un código 200.● Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si la ruta no se encuentra en la base de datos, se responde con un código 404.○ Si ocurre algún error no provocado por el usuario, se responde con un código 500.				
Observaciones: <ul style="list-style-type: none">● El formato de respuesta es JSON.● La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.● El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.				
ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RR-03	API	MODERADA	MODERADO	NORMAL
Descripción:				



El usuario debe poder modificar algún campo de una ruta existente mediante una petición PATCH cuyo cuerpo contenga el elemento en el tipo de dato correspondiente.

Criterios de Aceptación:

- **Éxito:** Petición PATCH.
 - La cabecera de la petición contiene un TOKEN de autenticación válido.
 - La URI de la petición contiene un ID de ruta válido y un código que hace referencia al campo a modificar.
 - Se busca en la base de datos la ruta según el ID.
 - Se actualiza el campo deseado.
 - Se guarda la ruta modifica en la base de datos.
 - La base de datos retorna un mensaje de éxito.
 - Se responde la petición con un código 200.
- **Fallo:**
 - Si el usuario no posee un TOKEN válido, se responde con un código 403.
 - Si la ruta no se encuentra en la base de datos, se responde con un código 404.
 - Si ocurre algún error no provocado por el usuario, se responde con un código 500.
 - Si la URI contiene un código que no hace referencia a un campo de la ruta, se responde con un código 400.

Observaciones:

- El formato de respuesta es JSON.
- La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.
- El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.
- Se debe establecer un código (no http) que corresponda a un campo de la ruta.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RR-04	API	MODERADA	MODERADO	NORMAL

Descripción:

El usuario debe poder eliminar una ruta de la base de datos mediante una petición DELETE. El usuario deberá especificar en la URI de la petición un código. Dependiendo del código, se eliminará todo la ruta o un campo de la misma.

Criterios de Aceptación:

- **Éxito:** Petición DELETE.
 - La cabecera de la petición contiene un TOKEN de autenticación válido.
 - La URI de la petición contiene un ID de ruta válido y un código válido.
 - Se verifica el código:
 - Si el código hace referencia a la ruta entera, se procede a borrar la ruta de la base de datos.



<ul style="list-style-type: none">▪ Si el código hace referencia a un campo de la ruta, se procede a borrar la información contenida en el campo y se guarda, nuevamente, la ruta modificada en la base de datos.○ La base de datos retorna un mensaje de éxito.○ Se responde la petición con un código 200.• Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si la ruta no se encuentra en la base de datos, se responde con un código 404.○ Si ocurre algún error no provocado por el usuario, se responde con un código 500.
Observaciones: <ul style="list-style-type: none">• El formato de respuesta es JSON.• La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.• El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RR-05	API	MODERADA	MODERADO	NORMAL
Descripción: <p>El usuario debe poder obtener la información de una ruta mediante una petición GET. Además, dependiendo del código que se especifique en la URI de la petición se retornará todos los campos o uno en específico.</p>				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición GET.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ La URI de la petición contiene un ID de ruta válido y un código que hace referencia al elemento o elementos de retorno.○ Se busca en la base de datos la ruta.○ La base de datos retorna la ruta.○ Se responde la petición con un código 200 y él/los campo/s solicitado/s.• Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si la ruta no se encuentra en la base de datos, se responde con un código 404.○ Si ocurre algún error no provocado por el usuario, se responde con un código 500.				
Observaciones: <ul style="list-style-type: none">• El formato de respuesta es JSON.• La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.				



- El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RS-01	API	MODERADA	MODERADO	NORMAL
Descripción: El usuario/cliente-móvil debe poder crear un nuevo subpunto mediante el envío una petición POST cuyo cuerpo sea el Modelo Subpunto.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición POST.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ El cuerpo de la petición contiene un Modelo Subpunto válido.○ Se almacena en la base de datos el nuevo subpunto.○ La base de datos retorna un mensaje de éxito y un ID único para el subpunto.○ Se responde la petición con un código 201, junto con el ID del nuevo subpunto.• Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si el cuerpo de la petición, es decir, el Modelo Subpunto no es válido, se responde con un código 400.				
Observaciones: <ul style="list-style-type: none">• El formato de respuesta es JSON.• La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.• El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.				

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RS-02	API	MODERADA	MODERADO	NORMAL
Descripción: El usuario/cliente-móvil debe poder crear un conjunto de nuevos subpuntos mediante el envío una petición POST cuyo cuerpo sea un arreglo (array) del Modelo Subpunto.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición POST.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ El cuerpo de la petición contiene un arreglo (array) del Modelo Subpunto válido.○ Se almacena en la base de datos los nuevos subpuntos.○ La base de datos retorna un mensaje de éxito y un ID único para cada uno de los nuevos subpuntos.				



<ul style="list-style-type: none">○ Se responde la petición con un código 201, junto con un arreglo (array) que contenga los ID de cada nuevo subpunto. <ul style="list-style-type: none">● Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si el cuerpo de la petición, es decir, el Modelo Subpunto no es válido, se responde con un código 400.
Observaciones: <ul style="list-style-type: none">● El formato de respuesta es JSON.● La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.● El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RS-03	API	MODERADA	MODERADO	NORMAL
Descripción: El usuario debe poder modificar un subpunto existente mediante el envío una petición PUT cuyo cuerpo sea el Modelo Subpunto.				
Criterios de Aceptación: <ul style="list-style-type: none">● Éxito: Petición PUT.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ La URI de la petición contiene un ID de subpunto válido.○ El cuerpo de la petición contiene un Modelo de Subpunto válido.○ Se actualiza en la base de datos el subpunto.○ La base de datos retorna un mensaje de éxito.○ Se responde la petición con un código 200.● Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si el subpunto no se encuentra en la base de datos, se responde con un código 404.○ Si ocurre algún error no provocado por el usuario, se responde con un código 500.				
Observaciones: <ul style="list-style-type: none">● El formato de respuesta es JSON.● La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.● El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.				

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RS-04	API	MODERADA	MODERADO	NORMAL
Descripción:				



El usuario/cliente-móvil debe poder eliminar un subpunto de la base de datos mediante una petición DELETE.

Criterios de Aceptación:

- **Éxito:** Petición DELETE.
 - La cabecera de la petición contiene un TOKEN de autenticación válido.
 - La URI de la petición contiene un ID de subpunto válido.
 - Se borra de la base de datos el subpunto con el ID suministrado.
 - La base de datos retorna un mensaje de éxito.
 - Se responde la petición con un código 200.
- **Fallo:**
 - Si el usuario no posee un TOKEN válido, se responde con un código 403.
 - Si el subpunto no se encuentra en la base de datos, se responde con un código 404.
 - Si ocurre algún error no provocado por el usuario, se responde con un código 500.

Observaciones:

- El formato de respuesta es JSON.
- La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.
- El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RS-05	API	MODERADA	MODERADO	NORMAL

Descripción:

El usuario/cliente-móvil debería poder obtener todos los subpuntos asociados/pertenecientes a una ruta, mediante una petición GET. La respuesta debe ser un arreglo (array) en el que cada ítem sea un objeto del Modelo Subpunto.

Criterios de Aceptación:

- **Éxito:** Petición GET.
 - La cabecera de la petición contiene un TOKEN de autenticación válido.
 - La URI de la petición contiene un ID de ruta válido.
 - Se busca en la base de datos todos los subpuntos asociados al ID.
 - La base de datos retorna un arreglo de subpuntos.
 - Se responde la petición con un código 200 y el arreglo (array) solicitado.
- **Fallo:**
 - Si el usuario no posee un TOKEN válido, se responde con un código 403.
 - Si no se encuentran subpuntos asociados al ID de la ruta suministrado, se responde con un código 404.
 - Si ocurre algún error no provocado por el usuario, se responde con un código 500.

Observaciones:



- El formato de respuesta es JSON.
- La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.
- El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.
- El ID, presente en la URI, hace referencia a un identificador único de una Ruta.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RS-06	API	MODERADA	MODERADO	NORMAL
Descripción: El usuario/cliente-móvil debería poder obtener todos los subpuntos asociados/pertenecientes a una ruta, mediante una petición GET. La respuesta debe ser un arreglo (array) en el que cada ítem sea únicamente el ID del subpunto.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición GET.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ La URI de la petición contiene un ID de ruta válido.○ Se busca en la base de datos todos los subpuntos asociados al ID.○ La base de datos retorna un arreglo (array) de ID de subpuntos.○ Se responde la petición con un código 200 y el arreglo (array) solicitado.• Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si no se encuentran subpuntos asociados al ID de la ruta suministrado, se responde con un código 404.○ Si ocurre algún error no provocado por el usuario, se responde con un código 500.				
Observaciones: <ul style="list-style-type: none">• El formato de respuesta es JSON.• La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.• El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.• El ID, presente en la URI, hace referencia a un identificador único de una Ruta.• El arreglo que se retorna posee únicamente el ID de cada uno de los subpuntos asociados al ID de la ruta que se suministró en la URI de la petición.				

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RS-07	API	MODERADA	MODERADO	NORMAL
Descripción: El usuario debe poder obtener la información de un subpunto mediante una petición GET. Además, dependiendo del código que se especifique en la URI de la petición se retornará todos los campos o uno en específico.				
Criterios de Aceptación:				



- **Éxito:** Petición GET.
 - La cabecera de la petición contiene un TOKEN de autenticación válido.
 - La URI de la petición contiene un ID de subpunto válido y un código que hace referencia al elemento o elementos de retorno.
 - Se busca en la base de datos.
 - La base de datos retorna el subpunto.
 - Se responde la petición con un código 200 y él/los campo/s solicitado/s.
- **Fallo:**
 - Si el usuario no posee un TOKEN válido, se responde con un código 403.
 - Si el subpunto no se encuentra en la base de datos, se responde con un código 404.
 - Si ocurre algún error no provocado por el usuario, se responde con un código 500.

Observaciones:

- El formato de respuesta es JSON.
- La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.
- El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RU-01	API	MODERADA	MODERADO	NORMAL
Descripción: El usuario administrador debe poder crear un nuevo usuario mediante el envío de una petición POST, cuyo cuerpo sea el Modelo Usuario.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición POST.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ El cuerpo de la petición contiene un Modelo Usuario válido.○ Se almacena en la base de datos el nuevo usuario.○ La base de datos retorna un mensaje de éxito y un ID único para el usuario.○ Se responde la petición con un código 201, junto con el ID del nuevo usuario.• Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si el cuerpo de la petición, es decir, el Modelo Usuario no es válido, se responde con un código 400.				
Observaciones: <ul style="list-style-type: none">• El formato de respuesta es JSON.• La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.				



- El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RU-02	API	MODERADA	MODERADO	NORMAL
Descripción: El usuario administrador debe poder modificar un usuario existente mediante el envío una petición PUT, cuyo cuerpo sea el Modelo Usuario.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición PUT.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ La URI de la petición contiene un ID de usuario válido.○ El cuerpo de la petición contiene un Modelo Usuario válido.○ Se actualiza en la base de datos el usuario.○ La base de datos retorna un mensaje de éxito.○ Se responde la petición con un código 200.• Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si el usuario no se encuentra en la base de datos, se responde con un código 404.○ Si ocurre algún error no provocado por el usuario, se responde con un código 500.				
Observaciones: <ul style="list-style-type: none">• El formato de respuesta es JSON.• La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.• El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.				

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RU-03	API	MODERADA	MODERADO	NORMAL
Descripción: El usuario administrador debe poder eliminar un usuario de la base de datos mediante una petición DELETE. El usuario deberá especificar en la URI de la petición un código. Dependiendo del código, se eliminará todo el usuario o un campo del mismo.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición DELETE.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ La URI de la petición contiene un ID de usuario válido y un código válido.○ Se verifica el código:				



<ul style="list-style-type: none"><ul style="list-style-type: none">▪ Si el código hace referencia al usuario, se procede a borrar el usuario de la base de datos.▪ Si el código hace referencia a un campo del usuario, se procede a borrar la información contenida en el campo y se guarda, nuevamente, el usuario modificado en la base de datos.○ La base de datos retorna un mensaje de éxito.○ Se responde la petición con un código 200.• Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si el usuario no se encuentra en la base de datos, se responde con un código 404.○ Si ocurre algún error no provocado por el usuario, se responde con un código 500.
Observaciones: <ul style="list-style-type: none">• El formato de respuesta es JSON.• La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.• El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RU-04	API	MODERADA	MODERADO	NORMAL
Descripción: <p>El usuario administrador puede realizar una búsqueda de usuarios mediante una petición GET. En la URI debe contener la cadena de caracteres a buscar. Se debe buscar coincidencias en todos los campos del usuario.</p>				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición GET.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ La URI de la petición contiene un valor a buscar.○ Se busca en la base de datos todos los usuarios, en los cuales alguno de sus campos coincida o contenga el valor a buscar.○ La base de datos retorna un arreglo de usuarios.○ Se responde la petición con un código 200 y el arreglo (array) de usuarios.• Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si no se encuentran usuarios en los que algún campo coincida o contenga la cadena solicitada, se responde con un código 404.○ Si ocurre algún error no provocado por el usuario, se responde con un código 500.				
Observaciones: <ul style="list-style-type: none">• El formato de respuesta es JSON.				



- La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.
- El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.
- El ID, presente en la URI, hace referencia a un identificador único de una Ruta.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RU-05	API	MODERADA	MODERADO	NORMAL
Descripción: El usuario administrador puede obtener todos los usuarios registrados en la base de datos mediante una petición GET.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición GET.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ Se busca en la base de datos todos los usuarios.○ La base de datos retorna un arreglo de usuarios.○ Se responde la petición con un código 200 y el arreglo (array) de usuarios.• Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si no se encuentran usuarios, se responde con un código 404.○ Si ocurre algún error no provocado por el usuario, se responde con un código 500.				
Observaciones: <ul style="list-style-type: none">• El formato de respuesta es JSON.• La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.• El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.• El ID, presente en la URI, hace referencia a un identificador único de una Ruta.				

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RU-06	API	MODERADA	MODERADO	NORMAL
Descripción: El usuario administrador puede obtener un usuario en específico mediante una petición GET. En la URI debe contener el ID del usuario deseado.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición GET.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ La URI de la petición contiene un ID de usuario válido.○ Se busca en la base de datos.○ La base de datos retorna el usuario.○ Se responde la petición con un código 200 y el usuario solicitado.• Fallo:				



- Si el usuario no posee un TOKEN válido, se responde con un código 403.
- Si el usuario no se encuentra en la base de datos, se responde con un código 404.
- Si ocurre algún error no provocado por el usuario, se responde con un código 500.

Observaciones:

- El formato de respuesta es JSON.
- La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.
- El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.
- El ID, presente en la URI, hace referencia a un identificador único de una Ruta.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RU-07	API	MODERADA	MODERADO	NORMAL

Descripción:

El usuario administrador puede obtener un listado de usuarios dependiendo del rol asignado. Se realiza a través de una petición GET. En la URI debe contener el rol.

Criterios de Aceptación:

- **Éxito:** Petición GET.
 - La cabecera de la petición contiene un TOKEN de autenticación válido.
 - La URI de la petición contiene el rol a buscar.
 - Se busca en la base de datos los usuarios, cuyo rol coincida con el rol suministrado en la petición.
 - La base de datos retorna un arreglo (array) de usuarios.
 - Se responde la petición con un código 200 y el arreglo de usuarios.
- **Fallo:**
 - Si el usuario no posee un TOKEN válido, se responde con un código 403.
 - Si no se encuentran usuarios con el rol suministrado en la petición, se responde con un código 404.
 - Si ocurre algún error no provocado por el usuario, se responde con un código 500.

Observaciones:

- El formato de respuesta es JSON.
- La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.
- El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.
- El ID, presente en la URI, hace referencia a un identificador único de una Ruta.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RU-08	API	MODERADA	MODERADO	NORMAL

Descripción:



El usuario administrador debe poder modificar algún campo de un usuario existente mediante una petición PATCH, cuyo cuerpo contenga el elemento en el tipo de dato correspondiente.

Criterios de Aceptación:

- **Éxito:** Petición PATCH.
 - La cabecera de la petición contiene un TOKEN de autenticación válido.
 - La URI de la petición contiene un ID de usuario válido y un código que hace referencia al campo a modificar.
 - Se busca en la base de datos el usuario según el ID.
 - La base de datos retorna el usuario.
 - Se actualiza el campo deseado.
 - Se guarda la ruta modifica en la base de datos.
 - La base de datos retorna un mensaje de éxito.
 - Se responde la petición con un código 200.
- **Fallo:**
 - Si el usuario no posee un TOKEN válido, se responde con un código 403.
 - Si el usuario no se encuentra en la base de datos, se responde con un código 404.
 - Si ocurre algún error no provocado por el usuario, se responde con un código 500.
 - Si la URI contiene un código que no hace referencia a un campo de la ruta, se responde con un código 400.

Observaciones:

- El formato de respuesta es JSON.
- La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.
- El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.
- Se debe establecer un código (no http) que corresponda a un campo de la ruta.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RU-09	API	MODERADA	MODERADO	NORMAL
<p>Descripción:</p> <p>Para poder realizar peticiones a la API se debe de poseer un TOKEN. Para ello se debe realizar una petición PATCH. El cuerpo de la petición debe contener un email y una contraseña de un usuario registrado.</p> <p>En la API existen 3 tipos de roles: administrador, conductor y cliente. Cada rol posee un TOKEN que le otorga permisos a determinadas rutas. El rol administrador puede hacer peticiones a casi el total de rutas, mientras que el rol conductor solo puede hacer peticiones a un número reducido de rutas. Por su parte el rol cliente solo puede hacer peticiones a rutas tipo GET.</p>				
<p>Criterios de Aceptación:</p> <ul style="list-style-type: none">• Éxito: Petición PATCH – Login Administrador.				



- Se busca en la base de datos el usuario según el email.
- La base de datos retorna el usuario.
- Se verifica el rol del usuario, debe ser ADMINISTRADOR.
- Se compara la contraseña suministrada en la petición con la contraseña almacenada en la base de datos.
- Se crea un nuevo TOKEN con permisos de ADMINISTRADOR.
- Se actualiza el campo TOKEN del usuario.
- Se almacena el usuario modificado en la base de datos.
- La base de datos retorna un mensaje de éxito.
- Se responde la petición con un código 200.
- **Fallo:**
 - Si el usuario no posee un rol administrador, se responde con un código 403.
 - Si el usuario no posee una contraseña válida, se responde con un código 403.
 - Si se produce un error al comparar la contraseña suministrada en la petición con la contraseña almacenada en la base de datos, se responde con un código 500.
 - Si se produce un error al almacenar en la base de datos al usuario con el TOKEN modificado, se responde con un código 500.
 - Si el usuario no se encuentra en la base de datos, se responde con un código 404.
 - Si ocurre algún error no provocado por el usuario, se responde con un código 500.

Observaciones:

- El formato de respuesta es JSON.
- La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.
- El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RU-10	API	MODERADA	MODERADO	NORMAL
Descripción: El usuario puede realizar un cambio de contraseña mediante una petición PATCH. El URI debe contener el ID del usuario. El cuerpo de petición debe contener la nueva contraseña sin cifrar.				
Criterios de Aceptación: <ul style="list-style-type: none">● Éxito: Petición PATCH.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ La URI de la petición contiene un ID de usuario válido.○ Se busca en la base de datos el usuario según el ID.○ La base de datos retorna el usuario.○ Se encripta la nueva contraseña.				



<ul style="list-style-type: none">○ Se modifica el campo contraseña del usuario con la nueva contraseña encriptada.○ Se almacena el usuario modificado en la base de datos.○ La base de datos retorna un mensaje de éxito.○ Se responde la petición con un código 200. <ul style="list-style-type: none">● Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si el usuario no se encuentra en la base de datos, se responde con un código 404.○ Si el ID de usuario enviado en la petición no es correcto, se responde con un código 400.○ Si ocurre algún error no provocado por el usuario, se responde con un código 500.
Observaciones: <ul style="list-style-type: none">● El formato de respuesta es JSON.● La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.● El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RU-11	API	MODERADA	MODERADO	NORMAL
Descripción: <p>El usuario puede realizar el cambio de su avatar mediante una petición PATCH. El URI debe contener el ID del usuario y un código que haga referencia la campo contraseña. El cuerpo de petición debe contener el nuevo avatar.</p>				
Criterios de Aceptación: <ul style="list-style-type: none">● Éxito: Petición PATCH.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ La URI de la petición contiene un ID de usuario válido y un código que hace referencia al campo contraseña.○ Se busca en la base de datos el usuario según el ID.○ La base de datos retorna el usuario.○ Se cambia el nombre del archivo para el avatar basado en la fecha de subida y el id del usuario.○ Se almacena físicamente el avatar en el servidor.○ Se modifica el campo avatar del usuario con el nombre del nuevo avatar.○ Se almacena el usuario modificado en la base de datos.○ La base de datos retorna un mensaje de éxito.○ Se responde la petición con un código 200.● Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.				



- Si el usuario no se encuentra en la base de datos, se responde con un código 404.
- Si el ID de usuario enviado en la petición no es correcto, se responde con un código 400.
- Si ocurre algún error no provocado por el usuario, se responde con un código 500.

Observaciones:

- El formato de respuesta es JSON.
- La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.
- El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RU-12	API	MODERADA	MODERADO	NORMAL
<p>Descripción:</p> <p>Para poder realizar peticiones a la API se debe de poseer un TOKEN. Para ello se debe realizar una petición PATCH. El cuerpo de la petición debe contener un email y una contraseña de un usuario registrado.</p> <p>En la API existen 3 tipos de roles: administrador, conductor y cliente. Cada rol posee un TOKEN que le otorga permisos a determinadas rutas. El rol administrador puede hacer peticiones a casi el total de rutas, mientras que el rol conductor solo puede hacer peticiones a un número reducido de rutas. Por su parte el rol cliente solo puede hacer peticiones a rutas tipo GET.</p>				
<p>Criterios de Aceptación:</p> <ul style="list-style-type: none">• Éxito: Petición PATCH – Login Conductor.<ul style="list-style-type: none">○ Se busca en la base de datos el usuario según el email.○ La base de datos retorna el usuario.○ Se verifica el rol del usuario, debe ser CONDUCTOR.○ Se compara la contraseña suministrada en la petición con la contraseña almacenada en la base de datos.○ Se crea un nuevo TOKEN con permisos de CONDUCTOR.○ Se actualiza el campo TOKEN del usuario.○ Se almacena el usuario modificado en la base de datos.○ La base de datos retorna un mensaje de éxito.○ Se responde la petición con un código 200.• Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un rol conductor, se responde con un código 403.○ Si el usuario no posee una contraseña válida, se responde con un código 403.○ Si se produce un error al comparar la contraseña suministrada en la petición con la contraseña almacenada en la base de datos, se responde con un código 500.				



<ul style="list-style-type: none">○ Si se produce un error al almacenar en la base de datos al usuario con el TOKEN modificado, se responde con un código 500.○ Si el usuario no se encuentra en la base de datos, se responde con un código 404.○ Si ocurre algún error no provocado por el usuario, se responde con un código 500.
Observaciones: <ul style="list-style-type: none">• El formato de respuesta es JSON.• La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.• El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RC-01	API	MODERADA	MODERADO	NORMAL
Descripción: El cliente puede hacer una petición GET para obtener información sobre el proyecto.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición GET.<ul style="list-style-type: none">○ No se requiere autenticación para la petición.○ Se responde la petición con un código 200.• Fallo:<ul style="list-style-type: none">○ Si ocurre algún error no provocado por el usuario, se responde con un código 500.				
Observaciones: <ul style="list-style-type: none">• El formato de respuesta es JSON.• La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.• El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.				

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RC-02	API	MODERADA	MODERADO	NORMAL
Descripción: Para poder realizar peticiones a la API se debe de poseer un TOKEN. El usuario CLIENTE puede solicitar un TOKEN mediante una petición GET. El cuerpo de la petición debe contener un email de un cliente registrado. En la API existen 3 tipos de roles: administrador, conductor y cliente. Cada rol posee un TOKEN que le otorga permisos a determinadas rutas. El rol administrador puede hacer peticiones a casi el total de rutas, mientras que el rol conductor solo puede hacer peticiones a un número reducido de rutas. Por su parte el rol cliente solo puede hacer peticiones a rutas tipo GET.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición GET – TOKEN Cliente.				



<ul style="list-style-type: none">○ Se valida el email de la petición.○ Se busca en la base de datos el cliente.○ La base de datos retorna un cliente.○ Se crea un nuevo TOKEN con permisos de CLIENTE.○ Se actualiza el campo TOKEN del cliente.○ Se almacena el cliente modificado en la base de datos.○ La base de datos retorna un mensaje de éxito.○ Se responde la petición con un código 200. <ul style="list-style-type: none">● Fallo:<ul style="list-style-type: none">○ Si el cliente no se encuentra en la base de datos, se responde con un código 404.○ Si ocurre algún error no provocado por el usuario, se responde con un código 500.○ Si el email suministrado en la petición no es válido, se responde con un código 400.
Observaciones: <ul style="list-style-type: none">● El formato de respuesta es JSON.● La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.● El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RC-03	API	MODERADA	MODERADO	NORMAL
Descripción: Se puede obtener información sobre un cliente mediante una petición GET. En la URI debe contener el EMAIL del cliente deseado.				
Criterios de Aceptación: <ul style="list-style-type: none">● Éxito: Petición GET.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ Se valida el email suministrado.○ Se busca en la base de datos.○ La base de datos retorna el cliente.○ Se responde la petición con un código 200 y el cliente solicitado.● Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si el cliente no se encuentra en la base de datos, se responde con un código 404.○ Si ocurre algún error no provocado por el usuario, se responde con un código 500.○ Si el email suministrado en la petición no es válido, se responde con un código 400.				
Observaciones: <ul style="list-style-type: none">● El formato de respuesta es JSON.				



- La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.
- El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RC-04	API	MODERADA	MODERADO	NORMAL
Descripción: Se puede obtener las rutas creadas entre un intervalo de tiempo mediante una petición GET. La URI de la petición debe contener la fecha de inicio del intervalo, así como la fecha de fin del intervalo.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición GET.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.○ Se validan las fechas suministradas (inicio y fin).○ Se busca en la base de datos.○ La base de datos retorna un arreglo (array) de rutas.○ Se responde la petición con un código 200 y las rutas dentro del intervalo.• Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se responde con un código 403.○ Si no se encuentran rutas dentro del intervalo de fechas en la base de datos, se responde con un código 404.○ Si ocurre algún error no provocado por el usuario, se responde con un código 500.○ Si la fecha de inicio o la fecha de fin no tienen el formato adecuado, se responde con un código 400.				
Observaciones: <ul style="list-style-type: none">• El formato de respuesta es JSON.• La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.• El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.				

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RC-05	API	MODERADA	MODERADO	NORMAL
Descripción: El cliente debe poder obtener la información de una ruta mediante una petición GET. Además, dependiendo del código que se especifique en la URI de la petición se retornará todos los campos o uno en específico.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición GET.<ul style="list-style-type: none">○ La cabecera de la petición contiene un TOKEN de autenticación válido.				



- La URI de la petición contiene un ID de ruta válido y un código que hace referencia al elemento o elementos de retorno.
- Se busca en la base de datos la ruta.
- La base de datos retorna la ruta.
- Se responde la petición con un código 200 y él/los campo/s solicitado/s.
- **Fallo:**
 - Si el usuario no posee un TOKEN válido, se responde con un código 403.
 - Si la ruta no se encuentra en la base de datos, se responde con un código 404.
 - Si ocurre algún error no provocado por el usuario, se responde con un código 500.

Observaciones:

- El formato de respuesta es JSON.
- La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.
- El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RC-06	API	MODERADA	MODERADO	NORMAL

Descripción:

El cliente debe poder obtener la información de un subpunto mediante una petición GET. Además, dependiendo del código que se especifique en la URI de la petición se retornará todos los campos o uno en específico.

Criterios de Aceptación:

- **Éxito:** Petición GET.
 - La cabecera de la petición contiene un TOKEN de autenticación válido.
 - La URI de la petición contiene un ID de subpunto válido y un código que hace referencia al elemento o elementos de retorno.
 - Se busca en la base de datos.
 - La base de datos retorna el subpunto.
 - Se responde la petición con un código 200 y él/los campo/s solicitado/s.
- **Fallo:**
 - Si el cliente no posee un TOKEN válido, se responde con un código 403.
 - Si el subpunto no se encuentra en la base de datos, se responde con un código 404.
 - Si ocurre algún error no provocado por el cliente, se responde con un código 500.

Observaciones:

- El formato de respuesta es JSON.
- La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.



- El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
RC-07	API	MODERADA	MODERADO	NORMAL
Descripción: El cliente puede darse de alta en la API mediante el envío de una petición POST, cuyo cuerpo sea el Modelo Cliente.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Petición POST.<ul style="list-style-type: none">○ El cuerpo de la petición contiene un Modelo Cliente válido.○ Se almacena en la base de datos el nuevo usuario.○ La base de datos retorna un mensaje de éxito y un ID único para el cliente junto con el token.○ Se responde la petición con un código 201, junto con el ID del nuevo usuario.• Fallo:<ul style="list-style-type: none">○ Si ocurre algún error no provocado por el cliente, se responde con un código 500.○ Si el cuerpo de la petición, es decir, el Modelo Cliente no es válido, se responde con un código 400.				
Observaciones: <ul style="list-style-type: none">• El formato de respuesta es JSON.• La cabecera de la respuesta debe contener el código de estado http acorde al evento representado.• El cuerpo de la respuesta debe contener un mensaje acorde al código de estado http.				



FluxLane Mobile

Las historias de usuario están divididas en distintos módulos. Éstos se definen a continuación:

MODULO	ID	COLOR
Gestión ruta	GR	
Gestión conductor	GC	

Las historias de usuario son construidas de acuerdo al formato indicado por la metodóloga Mobile-D con modificaciones para el caso de estudio, para ello en la siguiente figura mostramos la nomenclatura usada.

NOMENCLATURA			
TIPO	Móvil	Arreglo	Mejora
DIFICULTAD	Fácil	Moderada	Difícil
PRIORIDAD	Baja	Normal	Alta
ESFUERZO	Bajo	Normal	Alto

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GR-01	MÓVIL	MODERADA	MODERADO	NORMAL
Descripción: Al ingresar a la aplicación móvil, el conductor debe observar las rutas que tiene pendiente de recolección. Estas rutas se muestran en forma de lista.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito:<ul style="list-style-type: none">○ Se realiza una petición GET a la API. La cabecera de la petición contiene un TOKEN válido. El cuerpo de la petición contiene el ID del conductor.○ La API retorna las rutas pendientes del conductor.○ Se realiza una petición GET a la API para obtener la información detallada de cada ruta pendiente del conductor.○ Se listan todas las rutas pendientes del conductor. Detallando el nombre y la fecha de captura.• Fallo:<ul style="list-style-type: none">○ Si el conductor no posee un TOKEN válido, la aplicación muestra la pantalla de Inicio de Sesión.○ Si el conductor no tiene rutas pendientes, se muestra una lista vacía.				
Observaciones: <ul style="list-style-type: none">• El formato de petición es JSON.• Se debe tomar en cuenta que el formato de fecha del sistema es UTC, por lo que al mostrar fechas al usuario se debe cambiar el formato a UTC-5 (zona horaria de Ecuador).• Se deben mostrar en una sola lista todas las rutas pendientes, en el orden en el que se hayan recuperado de la API.				



ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GR-02	MÓVIL	MODERADA	MODERADO	NORMAL
Descripción: Cuando el conductor seleccione una ruta pendiente, se le mostrará otra vista en la que se detallará la información de la ruta. La información a mostrar es: <ul style="list-style-type: none">• Fecha de captura.• Hora de captura.• Intervalo de captura.• Clima Además, se le proporcionará, mediante un botón, la posibilidad de iniciar la captura de datos.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito:<ul style="list-style-type: none">○ El conductor selecciona una ruta pendiente.○ Mediante el ID de la ruta seleccionada se obtiene la información detallada.○ Se crea una nueva ventana.○ En la nueva ventana se muestran los datos detallados de la ruta. Además, un botón para poder iniciar la captura.• Fallo:<ul style="list-style-type: none">○ Si la ruta seleccionada no posea información completa, se cierra la vista y se procede a volver a consultar las rutas pendientes.○ En caso de que no haya conexión a internet, se muestra el mensaje correspondiente al conductor.				
Observaciones: <ul style="list-style-type: none">• El clima es información referencial, no tiene que cumplirse para poder capturar datos.• Se debe tomar en cuenta que el formato de fecha del sistema es UTC, por lo que al mostrar fechas al usuario se debe cambiar el formato a UTC-5 (zona horaria de Ecuador).				

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GR-03	MÓVIL	MODERADA	MODERADO	NORMAL
Descripción: Cuando el conductor pulse el botón para iniciar la captura, se abrirá otra ventana. En esta ventana se le mostrará un mapa con la ruta que tiene que seguir. Con todos los puntos de referencia de la ruta. Además, se muestra la latitud, longitud y la velocidad en el intervalo de tiempo que se haya establecido para la ruta. Junto a estas variables se deben establecer 4 botones: botón “Iniciar”, botón “Parar”, botón “Cerrar” y botón “Enviar”.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito:<ul style="list-style-type: none">○ El conductor pulsa el botón “Iniciar Captura”.○ Se crea una nueva ventana.				



- La mitad de la pantalla es un mapa con la ruta. Además, se colocan pines para determinar los puntos de referencia de la ruta.
- En la segunda mitad de pantalla se muestra las variables de captura: la latitud, la longitud y la velocidad del dispositivo.
- Estas variables cambian en el intervalo de tiempo que se haya especificado en la ruta.
- La latitud, longitud y velocidad se almacenan en el intervalo de tiempo especificado en una lista no visible al conductor.
- El botón de enviar se encuentra deshabilitado.
- **Fallo:**
 - Si no es posible trazar la ruta sobre el mapa, se cierra la ventana.
 - Si no es posible determinar la latitud o longitud, no se determina la velocidad. En este caso se almacena una velocidad 0.

Observaciones:

- El botón de cerrar se encuentra desactivado mientras se está capturando datos.
- El mapa que se visualiza junto con los marcadores para las referencias es propio de cada plataforma. Es decir, en iOS se invocará al mapa nativo. En Android, se usará Google Maps.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GR-04	MÓVIL	MODERADA	MODERADO	NORMAL
Descripción: Cuando el conductor haya llegado al final de la ruta, deberá parar la captura de datos. Después, se habilitará el botón enviar. Al pulsar dicho botón se envían los datos capturados a la API.				
Criterios de Aceptación: <ul style="list-style-type: none">● Éxito:<ul style="list-style-type: none">○ El conductor llega al final de la ruta.○ El conductor pulsa el botón “Parar”.○ Se habilita el botón “Enviar”.○ El conductor pulsa el botón “Enviar”.○ Los datos se preparan para ser enviados.○ Se realiza una petición POST con los datos. La cabecera de la petición posee un TOKEN válido. El cuerpo contiene los datos.○ Se obtiene una respuesta 202 de la API.○ Se procede a consultar los ID de todos los datos(subpuntos) guardados en la base de datos.○ Se obtiene los ID. Se realiza una petición PATCH a la API para actualizar el campo subpuntos de la ruta.○ Se obtiene una respuesta 200 de la API.○ Se procede a borrar el ID de la ruta del campo “rutas_pendientes” del usuario.○ Se obtiene una respuesta 200 de la API.○ Se muestra un mensaje de éxito al usuario.				



<ul style="list-style-type: none">○ Al pulsar sobre el mensaje de error se cierra la vista del mapa y se navega hasta la pantalla principal.○ Se actualiza la lista de rutas pendientes del usuario. <ul style="list-style-type: none">● Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se navega hasta la pantalla de Inicio de sesión.○ Si no se cuenta con conexión a internet, se le notifica al usuario.○ Si no se encuentra habilitada la opción de conectividad (datos o wi-fi) en el dispositivo, se le notifica al usuario que debe encenderla.
Observaciones: <ul style="list-style-type: none">● El mapa que se visualiza junto con los marcadores para las referencias es propio de cada plataforma. Es decir, en iOS se invocará al mapa nativo. En Android se usará Google Maps.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GC-01	MÓVIL	MODERADA	MODERADO	NORMAL
Descripción: <p>Cuando el usuario ingresa por primera vez a la aplicación o cuando haya cerrado sesión se debe mostrar la vista de Inicio de Sesión. En ella se pide el correo y contraseña. Al rellenar ambos campos se puede iniciar sesión.</p>				
Criterios de Aceptación: <ul style="list-style-type: none">● Éxito:<ul style="list-style-type: none">○ El conductor llena el formulario de inicio de sesión.○ El conductor pulsa el botón “Iniciar Sesión”.○ Se realiza una petición PATCH hacia la API. Esta petición contiene el correo y la contraseña.○ La API retorna el ID del usuario y un TOKEN válido.○ Se almacena localmente (memoria interna del dispositivo) el ID y el TOKEN.○ Se cierra la vista de Inicio de Sesión.○ Se navega hasta la vista principal (vista de Rutas).○ Se realiza una petición GET hacia la API. La cabecera cuenta con un TOKEN. Además, la petición posee el ID del usuario.○ La API responde con los datos del usuario.○ Se procede a recorrer el campo “rutas_pendientes” del usuario.○ Cada ítem corresponde a un ID de una ruta en estado “Pendiente”. Por cada ítem, se realiza una petición GET hacia la API.○ La API responde con los datos de cada ruta. Estos datos se almacenan en una lista.○ Se crea una lista de rutas pendientes.● Fallo:<ul style="list-style-type: none">○ Si el usuario no posee un TOKEN válido, se navega hasta la pantalla de Inicio de sesión.○ Si no se cuenta con conexión a internet, se le notifica al usuario.○ Si no se encuentra habilitada la opción de conectividad (datos o wi-fi) en el dispositivo, se le notifica al usuario que debe encenderla.				



- Si el conductor no posee rutas pendientes, la lista queda vacía.

Observaciones:

- Se debe tomar en cuenta que el formato de fecha del sistema es UTC, por lo que al mostrar fechas al usuario se debe cambiar el formato a UTC-5 (zona horaria de Ecuador).
- Solo los usuarios con rol “conductor” pueden iniciar sesión en la aplicación móvil.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GC-02	MÓVIL	MODERADA	MODERADO	NORMAL

Descripción:

Cuando el usuario ingresa por primera vez a la aplicación o cuando haya cerrado sesión se debe mostrar la vista de Inicio de Sesión. En ella se pide el correo y contraseña. Al rellenar ambos campos se puede iniciar sesión.

Criterios de Aceptación:

- **Éxito:**
 - El conductor llena el formulario de inicio de sesión.
 - El conductor pulsa el botón “Iniciar Sesión”.
 - Se realiza una petición PATCH hacia la API. Esta petición contiene el correo y la contraseña.
 - La API retorna el ID del usuario y un TOKEN válido.
 - Se almacena localmente (memoria interna del dispositivo) el ID y el TOKEN.
 - Se cierra la vista de Inicio de Sesión.
 - Se navega hasta la vista principal (vista de Rutas).
 - Se realiza una petición GET hacia la API. La cabecera cuenta con un TOKEN. Además, la petición posee el ID del usuario.
 - La API responde con los datos del usuario.
 - Se procede a recorrer el campo “rutas_pendientes” del usuario.
 - Cada ítem corresponde a un ID de una ruta en estado “Pendiente”. Por cada ítem, se realiza una petición GET hacia la API.
 - La API responde con los datos de cada ruta. Estos datos se almacenan en una lista.
 - Se crea una lista de rutas pendientes.
- **Fallo:**
 - Si el usuario no posee un TOKEN válido, se navega hasta la pantalla de Inicio de sesión.
 - Si no se cuenta con conexión a internet, se le notifica al usuario.
 - Si no se encuentra habilitada la opción de conectividad (datos o wi-fi) en el dispositivo, se le notifica al usuario que debe encenderla.
 - Si el conductor no tiene rutas pendientes, la lista queda vacía.

Observaciones:

- El formato de la petición es JSON.
- Se debe tomar en cuenta que el formato de fecha del sistema es UTC, por lo que al mostrar fechas al usuario se debe cambiar el formato a UTC-5 (zona horaria de Ecuador).



ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GC-03	MÓVIL	MODERADA	MODERADO	NORMAL
Descripción: El usuario debe poder Cerrar Sesión. Para ello deberá ejecutar la acción correspondiente. Esto implica que se debe borrar el ID y el TOKEN almacenados. Posteriormente, se debe navegar hasta la ventana de Inicio de Sesión.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito:<ul style="list-style-type: none">○ El conductor pulsa el botón “Cerrar Sesión”.○ Se procede a borrar las credenciales del usuario: ID y TOKEN.○ Se navega hasta la vista de Inicio de Sesión.• Fallo:<ul style="list-style-type: none">○ El dispositivo no navega hacia la pantalla de Inicio de Sesión, se reinicia la aplicación.				
Observaciones: <ul style="list-style-type: none">• Al cerrar sesión se podría liberar memoria mediante de la eliminación de la lista de rutas pendientes.				

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GC-04	MÓVIL	MODERADA	MODERADO	NORMAL
Descripción: El usuario debe poder ver la información registrada al momento de darse de alta en el sistema. Para ello cuenta con una vista en la que se muestra información relevante sobre su cuenta.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito:<ul style="list-style-type: none">○ El conductor accede a la vista “Perfil”.○ Se realiza una petición GET hacia la API. La cabecera de la petición posee un TOKEN válido. La petición incluye el ID del usuario.○ La API responde exitosamente con la información del usuario.○ Se listan todos los datos del usuario:<ul style="list-style-type: none">▪ Avatar.▪ Nombre.▪ Correo.▪ Vehículo.▪ Rol.▪ Fecha de registro.• Fallo:<ul style="list-style-type: none">○ El dispositivo no navega hacia la pantalla correspondiente, se muestra un mensaje para que el usuario reintente la acción.○ Si el usuario no posee un TOKEN válido, se navega hasta la pantalla de Inicio de sesión.○ Si no se cuenta con conexión a internet, se le notifica al usuario.				



- Si no se encuentra habilitada la opción de conectividad (datos o wi-fi) en el dispositivo, se le notifica al usuario que debe encenderla.
- Si no se obtiene una respuesta satisfactoria de la API, no se muestra la vista.

Observaciones:

- Si no es posible cargar el avatar, se mostrará un por defecto.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GC-05	MÓVIL	MODERADA	MODERADO	NORMAL

Descripción:

Adicionalmente se mostrará información sobre la versión de la aplicación, sobre las instituciones involucradas y sobre el objetivo del proyecto.

Criterios de Aceptación:

- **Éxito:**
 - El conductor accede a la vista “Información”.
 - Se muestra información estática sobre:
 - Versión de la aplicación.
 - Universidad Nacional de Loja – Carrera de Ingeniería en Sistemas.
 - Unidad Municipal de Tránsito.
 - Objetivo del proyecto.
 - Autor del proyecto.
 - Director del proyecto.
- **Fallo:**
 - El dispositivo no navega hacia la pantalla correspondiente, se muestra un mensaje para que el usuario reintente la acción.

Observaciones:

- La información es breve. En el caso de las instituciones solo se mostrará el nombre y su rol de colaborador.



FluxLane Web

Las historias de usuario están divididas en distintos módulos. Éstos se definen a continuación:

MODULO	ID	COLOR
GESTIÓN DE USUARIOS	GU	
GESTIÓN DE RUTAS	GR	

Las historias de usuario son construidas de acuerdo al formato indicado por la metodóloga Mobile-D con modificaciones para el caso de estudio, para ello en la siguiente figura mostramos la nomenclatura usada.

NOMENCLATURA			
TIPO	Web	Arreglo	Mejora
DIFICULTAD	Fácil	Moderada	Difícil
PRIORIDAD	Baja	Normal	Alta
ESFUERZO	Bajo	Normal	Alto

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GU-01	Web	MODERADA	MODERADO	NORMAL
Descripción: El administrador accede al enlace www.fluxlane.com . Allí puede iniciar sesión en el sistema web mediante el ingreso de correo y contraseña.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito: Inicio de sesión.<ul style="list-style-type: none">○ El administrador ingresa en el formulario su correo y contraseña.○ Se hace una petición PATCH hacia la API. La cabecera no tiene que contener un TOKEN. En el cuerpo de la petición consta el correo y contraseña.○ La API retorna un mensaje exitoso con el ID del administrador y un TOKEN válido.○ Se almacena localmente el ID y el TOKEN.○ Se procede a navegar a la vista inicial del sistema web.• Fallo:<ul style="list-style-type: none">○ Mensajes de error por cada dato mal ingresado en el formulario.○ Mensaje de error por usuario no registrado en el sistema.○ Mensaje de error en caso de que no haya respuesta del servidor.				
Observaciones: <ul style="list-style-type: none">• Solo los usuarios con rol “administrador” pueden iniciar sesión en el sistema web.• El sistema web no es responsive.• Contraseña mínimo 6 caracteres.				



ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GU-02	Web	MODERADA	MODERADO	NORMAL
Descripción: El administrador tendrá disponible una tabla con todos los usuarios registrados en el sistema. En la tabla se deberá mostrar el nombre, rol, estado y rutas pendientes del usuario “conductor”. También, contará con un conjunto de opciones para: <ul style="list-style-type: none">• Visualizar la información del usuario.• Cambiar la contraseña.• Cambiar el estado del usuario.• Editar la información.• Eliminar al usuario. Adicionalmente, se debe colocar un formulario para realizar búsquedas específicas en la base de datos. El resultado de esta búsqueda debe ser mostrado en la tabla. Además, se debe colocar un campo para filtrar el contenido de la tabla.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito:<ul style="list-style-type: none">○ El administrador selecciona la sección “Usuarios”.○ Se muestran dos opciones:<ul style="list-style-type: none">▪ Tabla de usuarios.▪ Crear nuevo usuario.○ El administrador selecciona “Tabla de usuarios”.○ Se carga en la vista el componente “Tabla de usuarios”.• Fallo:<ul style="list-style-type: none">○ No se puede recuperar la información de la base de datos, se muestra un mensaje de error.○ No carga la vista, se debe repetir la acción.				
Observaciones: <ul style="list-style-type: none">• Los usuarios con rutas pendientes tendrán el botón de “Eliminar” deshabilitado.• La tabla debe tener un paginador.• El formulario de búsqueda en la base de datos admite palabras con una longitud igual o superior a 4 caracteres.• El filtrado de resultados debe ser dinámico.				

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GU-03	Web	MODERADA	MODERADO	NORMAL
Descripción: En la sección “Opciones” de la tabla, el administrador puede elegir “Visualizar” la información del usuario. Al hacer clic se mostrará un modal con la siguiente información: <ul style="list-style-type: none">• Foto/Avatar.• Nombre.• ID.• Correo.• Rol.• Estado.				



- Vehículo.
- Fecha de alta.
- Fecha de último inicio de sesión.
- Rutas pendientes.

La información no puede ser modificada desde esta vista. Solo debe mostrar un botón de cerrar ventana.

Criterios de Aceptación:

- **Éxito:**
 - El administrador selecciona “Visualizar”.
 - Se muestra un modal con información:
 - Foto/Avatar.
 - Nombre.
 - ID.
 - Correo.
 - Rol.
 - Estado.
 - Vehículo.
 - Fecha de alta.
 - Fecha de último inicio de sesión.
 - Rutas pendientes.
 - El administrador hace clic en el botón “Cerrar”.
- **Fallo:**
 - No se puede recuperar la información de la base de datos, se muestra un mensaje de error.
 - No carga la vista, se debe repetir la acción.
 - El avatar no carga, se muestra un avatar genérico.

Observaciones:

- Se debe tomar en cuenta que el formato de fecha del sistema es UTC, por lo que al mostrar fechas al usuario se debe cambiar el formato a UTC-5 (zona horaria de Ecuador).
- Se puede mostrar un número entero que represente la cantidad total de rutas pendientes.
- En el campo Rutas pendientes se puede mostrar el ID de las rutas.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GU-04	Web	MODERADA	MODERADO	NORMAL

Descripción:

En la sección “Opciones” de la tabla, el administrador puede cambiar la contraseña del usuario. Para ello selecciona la opción “Cambiar contraseña”. Esto mostrará un modal con un formulario de cambio de contraseña. El formulario tendrá la opción de poder visualizar la contraseña.

Criterios de Aceptación:

- **Éxito:**
 - El administrador selecciona “Cambiar contraseña”.
 - Se muestra un modal con un formulario con los siguientes campos:
 - Nueva contraseña.



<ul style="list-style-type: none">▪ Confirmar contraseña.<ul style="list-style-type: none">○ El administrador rellena los dos campos.○ Selecciona “Actualizar” para guardar los cambios.○ Recibe un mensaje exitoso.○ Cierra la vista/modal.• Fallo:<ul style="list-style-type: none">○ Las contraseñas no coinciden, se muestra un mensaje de error.○ La contraseña tiene una longitud inferior a 6 caracteres, se muestra un mensaje de error.○ Se obtiene un mensaje de error al procesar la petición, se muestra el mensaje.
Observaciones: <ul style="list-style-type: none">• La contraseña debe tener una longitud igual o superior a 6 caracteres.• La vista debe tener un botón “Cancelar”.• La vista debe tener un botón “Cerrar”.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GU-05	Web	MODERADA	MODERADO	NORMAL
Descripción: <p>En ocasiones se deberá deshabilitar el acceso al sistema web o a la aplicación móvil de un usuario. Esto no implica que deba ser borrado de la base de datos. Por lo que debe existir una opción para cambiar el estado del usuario de “Activo” a “No Activo”.</p> <p>Para ello, en la sección “Opciones” de la tabla, el administrador selecciona la opción “Cambiar estado”. Esto mostrará un modal con un formulario de cambio de estado.</p>				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito:<ul style="list-style-type: none">○ El administrador selecciona “Cambiar estado”.○ Se muestra un modal con un formulario con un combo box:<ul style="list-style-type: none">▪ Activo.▪ No Activo.○ El administrador escoge una opción.○ Selecciona “Actualizar” para guardar los cambios.○ Recibe un mensaje exitoso.○ Cierra la vista/modal.• Fallo:<ul style="list-style-type: none">○ Se obtiene un mensaje de error al procesar la petición, se muestra el mensaje.				
Observaciones: <ul style="list-style-type: none">• No se debe tomar en cuenta el estado actual del usuario al momento de mostrar las opciones “Activo” y “No Activo”.				



ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GU-06	Web	MODERADA	MODERADO	NORMAL
Descripción: El administrador puede editar la información de un usuario. Para ello debe seleccionar en la opción “Editar”. Posterior a esto, se mostrará una nueva vista/pantalla con un formulario similar al formulario de registro. La diferencia radica en que todos los campos están rellenos y que algunos, como el correo y contraseña, están deshabilitados.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito:<ul style="list-style-type: none">○ El administrador selecciona “Editar”.○ Se muestra una nueva vista con un formulario lleno de:<ul style="list-style-type: none">▪ Nombre (Habilitado).▪ Correo (Deshabilitado).▪ Contraseña (Deshabilitado).▪ Repetir Contraseña (Deshabilitado).▪ Avatar (Habilitado).▪ Rol (Habilitado).▪ Tipo de vehículo (Habilitado).○ El administrador modifica uno o varios campos habilitados.○ Selecciona “Guardar” para guardar los cambios.○ Se muestra un modal para confirmación de cambios.○ Se selecciona “Guardar” y se realiza una petición PATCH hacia la API. La cabecera posee un TOKEN válido. El cuerpo posee los datos modificados del usuario.○ Recibe un mensaje exitoso desde la API.○ Se cierra la vista/pantalla.• Fallo:<ul style="list-style-type: none">○ Se obtiene un mensaje de error al procesar la petición, se muestra el mensaje.○ No es posible subir la imagen/avatar del usuario, se establece una por defecto.				
Observaciones: <ul style="list-style-type: none">• Si se escoge la opción “Cancelar” no se realizará cambio alguno en la información del usuario.• Al cambiar de avatar, el antiguo no se borrará del servidor.				



ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GU-07	Web	MODERADA	MODERADO	NORMAL
Descripción: Para eliminar un usuario del sistema, el administrador tiene que seleccionar la opción “Eliminar”. Se abrirá un modal/vista pidiendo confirmación. Una vez se haya realizado la petición a la API no se podrá deshacer la acción. Sin embargo, se conservarán las rutas asignadas al usuario.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito:<ul style="list-style-type: none">○ El administrador selecciona “Eliminar”.○ Se muestra un modal pidiendo confirmación.○ El usuario confirma el borrado.○ Se realiza una petición DELETE hacia la API. La cabecera posee un TOKEN válido. La petición posee el ID del usuario a eliminar.○ Recibe un mensaje exitoso desde la API.○ Se cierra la vista/pantalla.• Fallo:<ul style="list-style-type: none">○ Se obtiene un mensaje de error al procesar la petición, se muestra el mensaje.				
Observaciones: <ul style="list-style-type: none">• Las rutas asociadas al usuario eliminado se conservarán.• La imagen/avatar del usuario eliminado se conservará.				

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GU-08	Web	MODERADA	MODERADO	NORMAL
Descripción: Para la creación de un nuevo usuario se debe seleccionar la opción “Crear Usuario”. Una vez hecho esto, se abrirá una vista con un formulario.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito:<ul style="list-style-type: none">○ El administrador selecciona “Crear Usuario”.○ Se muestra una nueva vista con un formulario:<ul style="list-style-type: none">▪ Nombre.▪ Correo.▪ Contraseña.▪ Repetir Contraseña.▪ Avatar.▪ Rol.▪ Tipo de vehículo.○ El administrador llena todos los campos○ Selecciona “Guardar” para guardar los cambios.○ Se muestra un modal para confirmación de cambios.○ Se selecciona “Guardar” y se realiza una petición POST hacia la API. La cabecera posee un TOKEN válido. El cuerpo posee los datos modificados del usuario.				



<ul style="list-style-type: none">○ Recibe un mensaje exitoso desde la API.○ Se cierra la vista/pantalla.● Fallo:<ul style="list-style-type: none">○ Se obtiene un mensaje de error al procesar la petición, se muestra el mensaje.○ No es posible subir la imagen/avatar del usuario, se establece una por defecto.○ En caso de que algún campo no se encuentre llene al momento de guardar la información, se le avisará al usuario del inconveniente.○ Si las contraseñas suministradas no coinciden se le notificará al usuario del inconveniente.○ Si las contraseñas suministradas no poseen una longitud igual o superior a 6 caracteres, se notificará al usuario del inconveniente.
Observaciones: <ul style="list-style-type: none">● Todos los campos son obligatorios.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GR-01	Web	MODERADA	MODERADO	NORMAL
Descripción: <p>Se deberá crear una sección llamada “Editor”. En ella se realizarán todas las operaciones de visualización, creación, modificación y eliminación de rutas. Dentro de esta sección hay dos opciones: Tabla de Rutas y Crear Nueva Ruta.</p> <p>El administrador tendrá disponible una tabla con todas las rutas creadas. En la tabla se deberá mostrar el ID, nombre y estado. También, contará con un conjunto de opciones para:</p> <ul style="list-style-type: none">● Visualizar la información de la ruta.● Editar la ruta.● Eliminar la ruta. <p>Adicionalmente, se debe colocar un formulario para realizar búsquedas específicas en la base de datos. El resultado de esta búsqueda debe ser mostrado en la tabla. Además, se debe colocar un campo para filtrar el contenido de la tabla.</p>				
Criterios de Aceptación: <ul style="list-style-type: none">● Éxito:<ul style="list-style-type: none">○ El administrador selecciona la sección “Editor”.○ Se muestran dos opciones:<ul style="list-style-type: none">▪ Tabla de rutas.▪ Crear nueva ruta.○ El administrador selecciona “Tabla de rutas”.○ Se carga en la vista el componente “Tabla de rutas”.● Fallo:<ul style="list-style-type: none">○ No se puede recuperar la información de la base de datos, se muestra un mensaje de error.○ No carga la vista, se debe repetir la acción.				
Observaciones: <ul style="list-style-type: none">● La tabla debe tener un paginador.				



- El formulario de búsqueda en la base de datos admite palabras con una longitud igual o superior a 4 caracteres.
- El filtrado de resultados debe ser dinámico.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GR-02	Web	MODERADA	MODERADO	NORMAL
Descripción: <p>En la sección “Opciones” de la tabla, el administrador puede elegir “Visualizar” la información de la ruta. Al hacer clic se mostrará un modal con dos secciones. La primera sección “Info” cuenta con la siguiente información:</p> <ul style="list-style-type: none">• Nombre.• Descripción.• Fecha de captura.• Hora de. captura.• Clima.• Intervalo de captura.• Usuario asignado. <p>La segunda sección “Ruta” muestra un mapa con la ruta.</p> <p>La información no puede ser modificada desde esta vista. Solo debe mostrar un botón de cerrar ventana.</p>				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito:<ul style="list-style-type: none">○ El administrador selecciona “Visualizar”.○ Se muestra un modal con dos secciones:<ul style="list-style-type: none">▪ Info: Información de la ruta.▪ Ruta: un mapa con la ruta.○ El administrador interactúa con las secciones.○ El administrador hace clic en el botón “Cerrar”.• Fallo:<ul style="list-style-type: none">○ No se puede recuperar la información de la base de datos, se muestra un mensaje de error.				
Observaciones: <ul style="list-style-type: none">• Se debe tomar en cuenta que el formato de fecha del sistema es UTC, por lo que al mostrar fechas al usuario se debe cambiar el formato a UTC-5 (zona horaria de Ecuador).				

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GR-03	Web	MODERADA	MODERADO	NORMAL
Descripción: <p>El administrador puede editar la información de una ruta. Para ello debe seleccionar en la opción “Editar”. Posterior a esto, se mostrará una nueva vista/pantalla con un</p>				



formulario similar al formulario de “Crear Nueva Ruta”. La diferencia radica en que todos los campos están rellenos.

Criterios de Aceptación:

- **Éxito:**
 - El administrador selecciona “Editar”.
 - Se muestra una nueva vista con tres secciones (se detalla en GR-05, GR-06 y GR-07):
 - Información: formulario con campos referentes a detalles de la ruta (nombre, descripción, fecha, intervalo, clima y conductor).
 - Mapa: mapa en el que se dibuja la ruta a seguir. Se deben colocar puntos de referencia para trazar la ruta.
 - Resumen: vista en la que se muestra la información de la ruta y los puntos geográficos (latitud y longitud) por donde pasa la ruta.
 - En la sección “Resumen”, el administrador hace clic en “Guardar”.
 - Se muestra un modal para confirmación.
 - Se selecciona “Guardar” y se realiza una petición POST hacia la API. La cabecera posee un TOKEN válido. El cuerpo posee los datos de la ruta.
 - Recibe un mensaje exitoso desde la API.
 - Se cierra la vista/pantalla.
- **Fallo:**
 - Se obtiene un mensaje de error al procesar la petición, se muestra el mensaje.
 - En caso de que algún campo del formulario esté incompleto se notificará al usuario.

Observaciones:

- Todos los campos son requeridos.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GR-04	Web	MODERADA	MODERADO	NORMAL
Descripción: Para eliminar una ruta del sistema, el administrador tiene que seleccionar la opción “Eliminar”. Se abrirá un modal/vista pidiendo confirmación. Una vez se haya realizado la petición a la API no se podrá deshacer la acción.				
Criterios de Aceptación: <ul style="list-style-type: none">• Éxito:<ul style="list-style-type: none">○ El administrador selecciona “Eliminar”.○ Se muestra un modal pidiendo confirmación.○ El usuario confirma el borrado.○ Se realiza una petición DELETE hacia la API. La cabecera posee un TOKEN válido. La petición posee el ID de la ruta a eliminar.○ Recibe un mensaje exitoso desde la API.○ Se borra el ID de la ruta del campo “rutas_pendientes” del usuario asignado. Se realiza una petición PATCH hacia la API. La petición posee el ID de la ruta eliminada.				



<ul style="list-style-type: none">○ En caso de existir, se borran, también, los subpuntos que pertenezcan a esa ruta. Se realiza una petición GET hacia la API, para obtener los ID de todos los subpuntos asociados a la ruta.○ Se recibe un mensaje de éxito y con los ID de todos los subpuntos asociados.○ Se realiza una petición DELETE por cada subpunto.○ Se recibe un mensaje de éxito.○ Se cierra la vista/pantalla. <ul style="list-style-type: none">● Fallo:<ul style="list-style-type: none">○ Se obtiene un mensaje de error al procesar la petición, se muestra el mensaje.○ En caso de no poder borrar el ID del campo “rutas_pendientes” se notifica al usuario. El borrado debe ser manual.○ En caso de no poder borrar un subpunto se notifica al usuario. El borrado debe ser manual.
Observaciones: <ul style="list-style-type: none">● El orden de borrado de una ruta debe ser el siguiente: borrar la ruta, borrar el ID del campo “rutas_pendientes” del usuario asignado, y, por último, borrar todos los subpuntos asociados a la ruta. No se debe alterar el orden.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GR-05	Web	MODERADA	ALTO	NORMAL
Descripción: <p>Se deberá crear una sección llamada “Editor”. En ella se realizarán todas las operaciones de visualización, creación, modificación y eliminación de rutas. Dentro de esta sección hay dos opciones: Tabla de Rutas y Crear Nueva Ruta.</p> <p>Al seleccionar Crear Nueva Ruta se abre una vista con 3 secciones:</p> <ul style="list-style-type: none">● Información.● Mapa.● Resumen. <p>En esta historia de usuario se detallará el funcionamiento de la primera sección.</p>				
Criterios de Aceptación: <ul style="list-style-type: none">● Éxito:<ul style="list-style-type: none">○ El administrador selecciona “Crear Nueva Ruta”.○ Se muestra la primera sección “Información”.○ Se llenan los campos del formulario:<ul style="list-style-type: none">▪ Nombre.▪ Descripción.▪ Fecha.▪ Hora.▪ Clima.▪ Intervalo de captura.▪ Conductor asignado.● Fallo:				



- Si el nombre de ruta tiene una longitud inferior a 5 caracteres, se muestra un mensaje de error.
- Si algún campo queda vacío, se notifica del inconveniente.
- Si el intervalo no tiene un valor entre 1 y 10 segundos, se muestra un error.
- Si no se muestra un conductor, es debido a que aún no se ha registrado ninguno.
- Si la descripción del clima tiene una longitud inferior a 5 caracteres, se muestra un mensaje de error.

Observaciones:

- Todos los campos son obligatorios.
- La fecha mínima de selección debe ser igual a la fecha actual del sistema.
- La hora por defecto debe ser 08:00 am.
- El valor por defecto del campo intervalo debe ser 1.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GR-06	Web	MODERADA	ALTO	NORMAL
<p>Descripción:</p> <p>Se deberá crear una sección llamada “Editor”. En ella se realizarán todas las operaciones de visualización, creación, modificación y eliminación de rutas. Dentro de esta sección hay dos opciones: Tabla de Rutas y Crear Nueva Ruta.</p> <p>Al seleccionar Crear Nueva Ruta se abre una vista con 3 secciones:</p> <ul style="list-style-type: none">• Información.• Mapa.• Resumen. <p>En esta historia de usuario se detallará el funcionamiento de la segunda sección.</p>				
<p>Criterios de Aceptación:</p> <ul style="list-style-type: none">• Éxito:<ul style="list-style-type: none">○ El administrador selecciona la segunda sección: “Mapa”.○ Se muestra una vista con dos partes:<ul style="list-style-type: none">▪ Mapa. Mapa de la ciudad de Loja.▪ Puntos: Contenedor donde se irán agregando/quitando dinámicamente los puntos de referencia que se coloquen en el mapa. Puntos que sirve para trazar la ruta.○ En la parte del mapa se propone dos opciones:<ul style="list-style-type: none">▪ Asistido: permite crear una ruta en un solo trazo. Es decir, se colocan todos los puntos de referencia en una sola acción. La ruta se genera automáticamente. El usuario puede borrar algún punto si lo desea.▪ Manual: permite crear puntos de referencia. Al crear más de uno se traza la ruta de forma automática. El usuario puede borrar algún punto si lo desea.○ En el contenedor de puntos se agregan/quitan dinámicamente los puntos. Estos poseen un nombre y coordenadas (latitud y longitud).				



- Es posible cambiar el nombre de cada punto. Basta con seleccionar el nombre y editarlo.
- Se habilita el botón de guardado. Al hacer clic se guardan los puntos de referencia de la ruta. Se pueden crear tantos puntos como sea necesarios.
- **Fallo:**
 - Si no se genera la ruta a partir de los puntos, se debe revisar la conexión a internet y a la API del proveedor del mapa.
 - En caso de que no se borre algún punto se deberá cerrar la vista y comenzar de nuevo.
 - El nombre del punto no puede quedar vacío. En caso de que el usuario lo deje vacío, automáticamente se le asignará un nombre genérico.

Observaciones:

- El servicio de mapas propuesto es Maps Box, en su versión gratuita. Teniendo esto en cuenta, se debe limitar la consulta a una ruta con un máximo de 24 puntos/referencias por petición.

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GR-07	Web	MODERADA	ALTO	NORMAL

Descripción:

Se deberá crear una sección llamada “Editor”. En ella se realizarán todas las operaciones de visualización, creación, modificación y eliminación de rutas. Dentro de esta sección hay dos opciones: Tabla de Rutas y Crear Nueva Ruta.

Al seleccionar Crear Nueva Ruta se abre una vista con 3 secciones:

- Información.
- Mapa.
- **Resumen.**

En esta historia de usuario se detallará el funcionamiento de la tercera sección.

Criterios de Aceptación:

- **Éxito:**
 - El administrador selecciona la tercera sección: “Resumen”.
 - Se muestra una vista con dos partes:
 - Información. Resumen del formulario de la primera sección.
 - Puntos: Contenedor de puntos/referencias de la ruta trazada en el mapa. Estos puntos se usarán para dibujar/construir la ruta.
 - El administrador hace clic en guardar.
 - Se muestra un diálogo de confirmación. Se hace clic en “Guardar”.
 - Se hace una petición POST hacia la API. La cabecera posee un TOKEN válido. El cuerpo contiene la información de la ruta.
 - Se obtiene un mensaje exitoso y el ID de la nueva ruta.
 - Se hace una petición PATCH hacia la API. La cabecera posee un TOKEN válido. El cuerpo contiene el ID de la nueva ruta. Esta petición pretende modificar el campo “rutas_pendientes” del usuario asignado. Agregándole el ID de la nueva ruta.



<ul style="list-style-type: none">○ Se obtiene un mensaje exitoso. Se notifica al usuario administrador. <ul style="list-style-type: none">● Fallo:<ul style="list-style-type: none">○ En caso de que algún campo del resumen esté vacío, no se permite guardar la ruta.○ Si se obtiene un mensaje no exitoso desde la API, se notifica al administrador.○ En caso de no poder crear la nueva ruta, no se altera el estado del campo "rutas_pendientes" del usuario.
Observaciones: <ul style="list-style-type: none">● Cancelar la operación de guardado en el modal de confirmación, no implica que deba limpiarse los campos del formulario.● Una vez finalizada toda la operación de creación de una nueva ruta, se cierra la vista "Crear Nueva Ruta".

ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GR-08	Web	MODERADA	ALTO	NORMAL
Descripción: <p>Se deberá crear una sección llamada "Rutas". Esta sección está destinada al tratamiento de las rutas que cuenten con datos recolectados. Es decir, aquellas rutas que posean información obtenida desde la aplicación móvil.</p> <p>El administrador tendrá disponible una tabla. En la tabla se deberá mostrar el nombre, estado y fecha de captura. También, contará con un botón de opciones que permitirá visualizar la información recolectada por la aplicación móvil.</p> <p>Adicionalmente, se debe colocar un formulario para realizar búsquedas específicas en la base de datos. El resultado de esta búsqueda debe ser mostrado en la tabla. Además, se debe colocar un campo para filtrar el contenido de la tabla.</p>				
Criterios de Aceptación: <ul style="list-style-type: none">● Éxito:<ul style="list-style-type: none">○ El administrador abre la sección "Rutas".○ Se realiza una petición GET hacia la API. La cabecera posee un TOKEN válido.○ Se obtiene todas las rutas cuyo estado sea "Pendiente".○ Se crea la vista de la tabla con el resultado de la consulta.● Fallo:<ul style="list-style-type: none">○ Si se obtiene un mensaje no exitoso desde la API, se notifica al administrador.				
Observaciones: <ul style="list-style-type: none">● Al iniciar la vista se debe realizar una búsqueda automática para obtener rutas con información.● Si no existiesen rutas con información se debe mostrar el mensaje correspondiente.				



ID	TIPO	DIFICULTAD	ESFUERZO	PRIORIDAD
GR-09	Web	MODERADA	ALTO	NORMAL
<p>Descripción:</p> <p>Se deberá crear una sección llamada “Rutas”. Esta sección está destinada al tratamiento de las rutas que cuenten con datos recolectados. Es decir, aquellas rutas que posean información obtenida desde la aplicación móvil.</p> <p>Cuando el administrador acceda a las opciones se mostrará otra vista con:</p> <ul style="list-style-type: none">• Información: cuenta con información de la ruta.• Ruta (mapa): muestra en un mapa dos rutas.<ul style="list-style-type: none">○ Ruta creada. Esta ruta se genera a partir de los puntos/referencias establecidas en la creación de la ruta.○ Ruta obtenida. Esta ruta se genera con la información recogida por el dispositivo móvil.• Subpuntos. Se muestran la siguiente información.<ul style="list-style-type: none">○ Subpuntos. Latitud, longitud y velocidad de cada subpunto.○ Velocidad promedio.○ Total de subpuntos.○ Hora de inicio de captura.○ Hora de fin de captura.○ Duración de viaje.○ Velocidad máxima.○ Velocidad mínima.• Opciones. En esta sección se muestran las opciones para cambiar de estado la ruta. El administrador deberá escoger alguna de las opciones y proceder a guardar los cambios.<ul style="list-style-type: none">○ Aprobar. Los datos recolectados son válidos para su publicación.○ Revisar. Los datos se han revisado, pero no se han aprobado.○ Rechazar. Los datos no son válidos.				
<p>Criterios de Aceptación:</p> <ul style="list-style-type: none">• Éxito:<ul style="list-style-type: none">○ El administrador selecciona el botón “Opciones” de una ruta.○ Se realizar una petición GET hacia la API para obtener toda la información de la ruta.○ Se recibe la información.○ Se obtiene los ID de los subpuntos. Se realiza una petición GET hacia la API para obtener la información de cada subpunto.○ Se recibe la información.○ Se realiza una petición GET hacia la API para consultar el usuario encargado de la ruta.○ Se recibe la información.○ Se procede a mostrar la vista y carga la información.• Fallo:<ul style="list-style-type: none">○ Si se obtiene un mensaje no exitoso desde la API, se notifica al administrador.				



- Es posible que el usuario decida buscar, desde esta vista, rutas con un estado diferente a “pendiente”, en ese caso no se debe cargar la vista de “Opciones”.

Observaciones:

- En la sección Ruta (mapa), la ruta creada debe tener un color distinto de la ruta obtenida.
- Es conveniente que en la sección “Subpuntos” se habilite un botón para eliminar los subpuntos con una velocidad igual a 0.