Historia de usuario #2

Anexo de Documentos Relacionados:

• APIDog. (s.f.). *REST API URL best practices* & examples. APIDog. https://apidog.com/blog/rest-api-url-best-practices-examples/

Descripción conceptual

Módulo			Gestión de Viajes.
Descripción de requerida(s):	la(s)	funcionalidad(es)	Permitir a los conductores publicar viajes con detalles como origen, destino, horario, precio y número de asientos.

Backend

URL	Método	Código html
https://localhost:8080/api/trips	POST	201 (Creado), 400 (Solicitud Incorrecta)

Caso de uso tecnico

Este endpoint permite a los conductores publicar viajes. Si los datos son válidos, retorna 201 con el ID del viaje creado. Si hay errores, retorna 404 con un cuerpo vacío.

```
Datos de entrada
                                                             Datos de salida
                                     201:
"origin": "Universidad Nacional",
                                       "status": "success",
 "destination": "Calle 45",
 "schedule":
                                       "data": {
"2025-02-01T08:00:00",
                                        "trip id": 1,
                                        "origin": "Universidad Nacional",
"seats": 3,
 "price": 10000
                                        "destination": "Calle 45",
                                        "seats": 3
                                      }
                                     404:
                                       "status": "error",
                                       "message": "Missing origin or destination"
```

Frontend

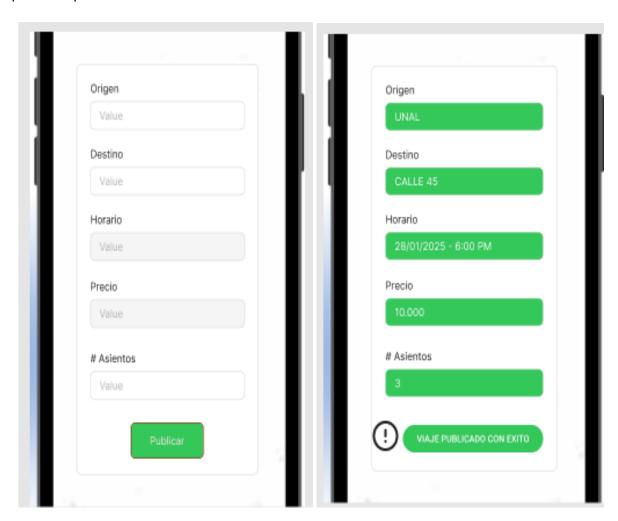
El formulario para publicar un viaje debe permitir al conductor ingresar los detalles necesarios de manera clara y organizada. Debe incluir campos para especificar el origen y destino del viaje, un selector de fecha y hora para el horario, un campo numérico para el precio, y un control deslizante o selector para el número de asientos, limitado entre 1 y 5. Además, el formulario debe mostrar validaciones dinámicas en tiempo real, resaltando en rojo cualquier campo con errores o vacío.

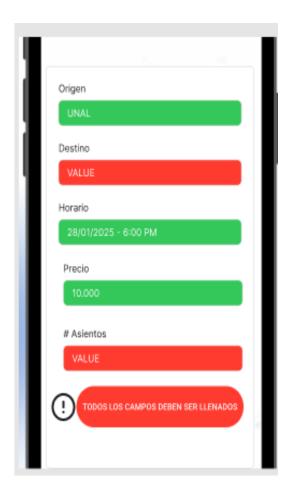
El botón "Publicar" debe permanecer deshabilitado hasta que todos los campos estén correctamente llenados, cambiando su estado a viaje publicado con exito cuando la información sea válida. De lo contrario se mostraran en rojo los campos que faltan por completar

Interacción esperada:

- Formulario para que el conductor ingrese los detalles del viaje.
- Botón "Publicar" para enviar los datos al servidor.
- Visualización del mensaje de estado para indicar el estado final de la publicación.

Mockups/Prototipos:





Flujo visual y eventos:

- 1. El conductor llena los campos del formulario y presiona "Publicar".
- 2. Si la publicación es exitosa, aparece un mensaje de éxito.
- 3. Si hay errores, se resaltan los campos incorrectos.