
Historias de Usuario: Plataforma Web DrivUp (Modulo Transporte Programado)

HU001-Header: Encabezado de Navegación Global

- **Criterios de Aceptación**

1. **Identificación de Marca (Izquierda):**

- [UI] Mostrar el ícono de DrivUp seguido del texto "DrivUp".
- [UI] Los elementos deben estar alineados verticalmente.
- [UI] Hacer clic en el ícono o Nombre navega al usuario a la Página de Inicio (/).

2. **Navegación Principal (Centro):**

- [UI] Mostrar enlaces de navegación horizontalmente: "Inicio" (/), "Servicios" (/services - podría llevar a una página que explique ambos tipos: programado/dinámico), "Acerca de DrivUp", (/acerca-DrivUp)"Contacto" (/contact).

3. **Acciones de Autenticación (Derecha):**

- **Estado Sin Sesión Iniciada:**

- [UI] Mostrar dos botones distintos: "Iniciar Sesión" y "Registrarse".
- [UI] Botón "Iniciar Sesión" Navega a /login.
- [UI] Botón "Registrarse" Navega a /register.

- **Estado Con Sesión Iniciada:**

- [UI] Reemplazar botones de Iniciar Sesión/Registrarse con un indicador de perfil de usuario (p.ej., Avatar de Usuario/Círculo con Iniciales) y potencialmente su nombre o un ícono de menú desplegable.
- [UI] Hacer clic en el indicador de perfil debe abrir un menú desplegable.
- [UI-DROPDOWN] El menú contiene enlaces como: "Mi Panel" (navega a /dashboard/passenger o /dashboard/driver según el rol), "Configuración" (/settings), y "Cerrar Sesión".

- [FUNC] "Cerrar Sesión" dispara el proceso de logout (limpia token, redirige a /).
-

HU002-Footer: Pie de Página con Información Global

- **Criterios de Aceptación:**

1. **Verificación de Contenido:**

- [UI] Asegurar que *todos* los elementos de texto proporcionados en el estén presentes y transcritos correctamente:
 - Nombre/Slogan de Marca: "DrivUp", "Transformando la movilidad urbana con tecnología y comunidad."
 - Secciones: "Plataforma", "Compañía", "Legal", "Licencias".
 - Enlaces bajo **Plataforma**: "Cómo funciona", "Características", "Precios", "FAQ".
 - Enlaces bajo **Compañía**: "Sobre nosotros", "Carreras", "Contacto".
 - Enlaces bajo **Legal**: "Términos de servicio", "Política de privacidad".
 - Texto de **Licencia**: "© 2025 DrivUp. Todos los derechos reservados."

2. **Funcionalidad de Enlaces:**

- [FUNC] Todos los enlaces listados ("Cómo funciona", "Términos de servicio", etc.) deben ser enlaces clicables reales (<a> o Link de React Router).
 - [FUNC] Los enlaces deben navegar a las páginas correspondientes correctas (p.ej., /como-funciona, /terminos, /privacidad, /acerca, /contacto).
-

HU003-01-Home: Sección Hero de la Página de Inicio

- **Criterios de Aceptación:**

1. **Titular y Subtítulo:**

- [UI] Mostrar el titular principal: "Viaja seguro con DrivUp".
- [UI] Mostrar el subtítulo descriptivo: "Conectamos pasajeros y conductores de forma segura, transparente y a precios justos. Únete a nuestra comunidad y transforma tu forma de moverte por la ciudad."

- [STYLING] El titular debe ser grande y prominente (text-4xl o text-5xl, negrita, text-primary). El subtítulo debe ser más pequeño (text-lg o text-xl, text-secondary).

2. Llamadas a la Acción (Botones):

- [UI] Mostrar dos botones prominentes debajo del texto.
- [UI] Texto Botón 1: "Solicitar Viaje". Debería ser un botón de acción primario. Inicialmente, podría enlazar a /login o /registro si se requiere iniciar sesión primero, o directamente al panel del pasajero (/dashboard/pasajero) si ya inició sesión.
- [UI] Texto Botón 2: "Conviértete en Conductor". Enlaza a /register?role=driver o al dashboard/conductor si ya inicio sesión como pasajero

HU003-02-Home: Sección "¿Por qué elegir DrivUp?"

- **Criterios de Aceptación:**

1. Título de Sección:

- [UI] Mostrar un título claro para la sección, p.ej., "¿Por qué elegir DrivUp?". Centrado
- [UI] incluir una frase introductoria corta bajo el título "Nuestra plataforma está diseñada pensando en la seguridad, transparencia y comodidad tanto de pasajeros como de conductores."

2. Bloques/Tarjetas de Beneficios (Mínimo 3):

- [UI] Presentar los beneficios en un formato visualmente distinto (p.ej., tarjetas en fila (mobiles) /cuadrícula (deskop)). Cada bloque debe tener un ícono, un título, una descripción corta
- **Beneficio 1: Seguridad Garantizada:**
 - [UI] Ícono (p.ej., escudo, checkmark).
 - [UI] Título: "Seguridad garantizada".
 - [UI] Descripción: "Verificamos a todos nuestros conductores. Requerimos documentos de identidad, licencia de conducir vigente, seguro vehicular y revisión de antecedentes para asegurar tu tranquilidad en cada viaje."
- **Beneficio 2: Comunidad Confiable:**
 - [UI] Ícono (p.ej., personas, estrellas).

- [UI] Título: "Comunidad confiable".
- [UI] Descripción: "Construimos una comunidad basada en la confianza y el respeto mutuo. Las calificaciones y comentarios bidireccionales ayudan a mantener altos estándares."
- **Beneficio 3: Viajes Eficientes y Optimizados:**
 - [UI] Ícono (p.ej., ruta de mapa, reloj).
 - [UI] Título: "Viajes Eficientes".
 - [UI] Descripción: "Nuestra tecnología (para el módulo dinámico) busca optimizar rutas y agrupar pasajeros inteligentemente, reduciendo tiempos de espera y costos. El módulo programado te da control directo para agendar."

3. Diseño y Estilos:

- [UI] Organizar los bloques de beneficios horizontalmente en pantallas más anchas (p.ej., cuadrícula de 3 columnas) y apilarlos verticalmente en pantallas más pequeñas.

HU003-03-Home: Sección Final de Llamada a la Acción

- **Criterios de Aceptación:**

1. Texto Persuasivo:

- [UI] Mostrar el título: "Únete a DrivUp hoy mismo".
- [UI] Mostrar el texto de apoyo: "Ya sea que necesites un viaje o quieras generar ingresos como conductor, DrivUp es tu mejor opción."
- [STYLING] El texto debe ser claro y centrado o alineado apropiadamente dentro de la sección.

2. Botones de Acción:

- [UI] Mostrar botones de acción relevantes. Tentativamente "Descargar la app":
 - Botón 2: "Conocer más" (Secundario, contorneado o bg-secondary, enlaza quizás a /como-funciona).

3. Diseño y Estilos:

- [UI] Típicamente una sección de ancho completo con contenido centrado.

HU004-01-Autenticacion: Registro de Usuario

- **Criterios de Aceptación:**

1. **Selección de Rol:**

- [UI] El formulario debe presentar claramente una opción para seleccionar el tipo de registro: "Soy Pasajero" o "Soy Conductor".
- [FUNC] Seleccionar un rol debe mostrar dinámicamente los campos de entrada relevantes.

2. **Campos de Pasajero (Mostrados si se selecciona "Soy Pasajero"):**

- [UI] Nombre(s) (Input tipo texto, requerido).
- [UI] Apellido(s) (Input tipo texto, requerido).
- [UI] Número de Documento de Identidad Nacional (p.ej., Cédula de Ciudadanía) (Input tipo texto, requerido).
- [UI] Dirección de Correo Electrónico (Input tipo email, requerido, debe tener formato válido).
- [UI] Número de Teléfono (Input tipo tel, requerido, potencialmente con manejo/validación de código de país).
- [UI] Fecha de Nacimiento (Input tipo date, requerida, para verificación de edad).
- [UI] Contraseña (Input tipo password, requerida, longitud mínima 8 caracteres, se sugiere combinación de letras/números).
- [UI] Confirmar Contraseña (Input tipo password, requerida, debe coincidir con el campo Contraseña).
- [UI] Casilla de verificación para aceptar los Términos de Servicio y la Política de Privacidad (Requerida, con enlaces a las páginas respectivas).

3. **Campos de Conductor (Mostrados si se selecciona "Soy Conductor"):**

- [UI] Todos los Campos de Pasajero
- **Detalles Personales:**
 - [UI] Número de Licencia de Conducir (Input tipo texto, requerido).
 - [UI] Fecha de Vencimiento de la Licencia de Conducir (Input tipo date, requerida).
 - [UI] Foto de Perfil (Input tipo file, opcional pero recomendada, acepta formatos de imagen como JPG, PNG, tamaño máx 2MB).

- **Detalles del Vehículo:**
 - [UI] Marca del Vehículo (p.ej., Renault, Chevrolet)
 - [UI] Modelo del Vehículo
 - [UI] Año del Vehículo (Input tipo number, requerido)
 - [UI] Color del Vehículo (Input tipo texto).
 - [UI] Placa (Matrícula) (Input tipo texto, requerido, restricción única en la base de datos).
 - [UI] Capacidad de Pasajeros (Excluyendo al Conductor) (Input tipo number, requerido, min 1).
- **Carga de Documentos (Crucial para Formalización):**
 - [UI] Documento de Registro Vehicular (Tarjeta de Propiedad) (Input tipo file, requerido, acepta PDF/JPG/PNG, tamaño máx 5MB).
 - [UI] Seguro Vehicular Vigente (SOAT y potencialmente Póliza Responsabilidad Civil) (Input tipo file, requerido, acepta PDF/JPG/PNG, tamaño máx 5MB).
 - [UI] Escaneo/Foto de Licencia de Conducir (Input tipo file, ayuda a la verificación).

4. Envío y Validación del Formulario:

- [UI] Un botón claro "Registrarse", deshabilitado hasta que todos los campos requeridos para el rol seleccionado estén llenados correctamente.
- [FUNC] **Validación Frontend:** Implementar validación en tiempo real donde sea posible (p.ej., formato de email, coincidencia de contraseña, campos requeridos, tipos/tamaños de archivo). Mostrar mensajes de error claros y amigables junto a los campos inválidos (text-error).
- [FUNC] Al enviar, mostrar un indicador de carga.
- [FUNC] En registro exitoso (API responde 201), redirigir al usuario a la página de Login (/login). Mostrar un mensaje de éxito (p.ej., "Registro exitoso. Por favor, inicia sesión."). El conductor podría ver "Registro exitoso. Tu cuenta podría ser revisada para activación."
- [FUNC] En error de API (p.ej., email ya existe, validación falló en servidor), mostrar un mensaje de error general en la parte superior del formulario o errores específicos de campo si los provee la API.

5. Contrato API Backend:

- **Endpoint:** POST /registro

- **Cuerpo de la Solicitud (Content-Type: multipart/form-data debido a subida de archivos):**
 - // Ejemplo para Conductor (enviado como campos de form data)
 - {
 - "role": "driver", // o "passenger"
 - "firstName": "Julián",
 - "lastName": "Pérez",
 - "email": "jperez@email.com",
 - "phoneNumber": "+573001234567",
 - "password": "securePassword123", // Texto plano aquí, hashear en servidor
 - // Específico de conductor:
 - "dateOfBirth": "1990-05-15",
 - "nationalId": "123456789",
 - "driverLicenseNumber": "DL987654",
 - "driverLicenseExpiry": "2028-05-15",
 - "vehicleMake": "Renault",
 - "vehicleModel": "Duster",
 - "vehicleYear": 2020,
 - "vehicleColor": "Gris",
 - "licensePlate": "XYZ789",
 - "vehicleCapacity": 4,
 - "agreedToTerms": true,
 - "consentedBackgroundCheck": true
 - // Archivos adjuntos: profilePicture, registrationDoc, insuranceDoc, licenseScan (opcional)
 - }
-

HU004-02-Authenticacion: Inicio de Sesión de Usuario

- **Criterios de Aceptación:**

1. **Campos del Formulario de Login:**

- [UI] Mostrar un campo de entrada para Dirección de Email (tipo email, requerido).
- [UI] Mostrar un campo de entrada para Contraseña (tipo password, requerido).
- [UI] Incluir un enlace "¿Olvidaste tu contraseña?" (navega a /forgot-password - implementación fuera de esta historia).
- [UI] Un botón claro "Iniciar Sesión" (primario bg-primary).

2. **Envío y Validación del Formulario:**

- [UI] Botón deshabilitado hasta que ambos campos estén llenos.
- [FUNC] Validación frontend para formato de email y contraseña no vacía.
- [FUNC] **Redirigir basado en rol:**
 - Si el rol es 'passenger', redirigir a /dashboard/pasajero.
 - Si el rol es 'driver', redirigir a /dashboard/conductor.

3. **Contrato API Backend:**

- **Endpoint:** POST /api/auth/login
- **Cuerpo de la Solicitud (Content-Type: application/json):**
 - {
 - "email": "jperez@email.com",
 - "password": "securePassword123"
 - }

Lógica Backend (Explicación Dev Junior):

- Recibir la solicitud. Validar que email y password estén presentes.
- **Buscar Usuario:** Consultar la tabla usuarios por un usuario con el email proporcionado.
- **Verificar si Usuario Existe:** Si no se encuentra usuario, devolver 404 Not Found (o ambiguamente 401 Unauthorized).

- **Verificar Contraseña:** Si el usuario existe, usar `bcrypt.compare()` para comparar la password proporcionada con el `password_hash` almacenado.
- **Verificar Coincidencia de Contraseña:** Si las contraseñas no coinciden, devolver 401 Unauthorized.
- **Verificar Estado de Cuenta (Conductores):** Si es un conductor, verificar su `verification_status` en la tabla conductores.
- **Generar JWT:** Si la contraseña coincide y la cuenta está activa, generar un JSON Web Token (JWT) usando una librería como `jsonwebtoken`.
 - **Payload:** Incluir `userId: user.user_id`, `role: user.role` en el payload. *NO incluir datos sensibles.*
- **Respuesta:** Devolver 200 OK con el JWT e información básica del usuario (como rol, quizás nombre, *excluyendo hash de contraseña*) por ejemplo:
 - {
 - "user": { "id": 123, "role": "driver", "firstName": "Julián" }
 - }

HU005-01-lista-conductores: Modulo transporte programado (Vista Pasajero)

- **Historia de Usuario:** Como **Pasajero con sesión iniciada**, quiero **ver una lista de conductores disponibles que ofrecen servicios de transporte programado** para que pueda **explorar sus perfiles y elegir uno para contactar**.
- Esta vista actúa como un directorio o catálogo de conductores que participan en el sistema de "Transporte Compartido Programado". Debe presentar información clave de forma concisa para permitir a los pasajeros comparar y seleccionar conductores.
- **Criterios de Aceptación:**
 1. **Obtención y Muestra de Datos:**
 - [FUNC] Al cargar la página, obtener una lista de conductores *activos* desde la API backend (GET `/api/conductores`).
 - [UI] Mostrar un indicador de carga mientras se obtienen los datos.
 - [UI] Si no se encuentran conductores, mostrar un mensaje amigable (p.ej., "No hay conductores disponibles para transporte programado en este momento.").

2. Lista/Cuadrícula de Conductores:

- [UI] Mostrar los conductores en formato de lista de tarjetas (Una sola columna, varias filas, ocupando gran parte del ancho).
- [UI] Cada entrada de conductor (tarjeta/fila) debe mostrar:
 - **Foto de Perfil**
 - **Nombre Completo**
 - **Calificación Promedio:** Mostrada visualmente usando estrellas (p.ej., 4.5 estrellas de 5).
 - **Capacidad del Vehículo:** Texto como "Capacidad: X pasajeros" (de vehicles.capacity).
 - **Info Vehículo** vehicles.make y vehicles.model.
 - **Fecha de Ingreso:** Texto como "Miembro desde: Mes Año" (de tabla conductores.created_at).
- [UI] Incluir un botón/enlace claro en cada entrada: "Ver Detalles del conductor y el botón de Contactar".

3. Filtrado/Ordenación:

- [UI] Añadir controles para filtrar por capacidad del vehículo u ordenar por calificación del conductor, o por id del conductor o nombre del conductor.

- **Logica Backend:**

1. **Endpoint:** GET /api/conductores

2. **Query Params (Opcional para filtros):**

?sortBy=rating&minCapacity=3&page=1&limit=10

- Consultar la tabla conductor, haciendo join con vehiculos.
- Aplicar filtrado/ordenación basado en query params.
- Implementar paginación.
- Devolver 200 OK con un array de objetos de conductor conteniendo los campos necesarios para el frontend (incluyendo la calificación promedio calculada/almacenada).

- // GET /api/conductores -> Ejemplo de Respuesta

- [

- {

- "driverId": 5,

- "firstName": "Ana",
- "lastName": "Garcia",
- "profilePictureUrl": "/uploads/drivers/ana_garcia.jpg",
- "averageRating": 4.8, // Calculada o almacenada
- "vehicle": {
- "make": "Chevrolet",
- "model": "Onix",
- "capacity": 4
- },
- "joinedDate": "2024-01-15T10:00:00Z"
- },
- // ... más conductores
-]

HU005-02-detalles-conductor: Ver Detalles del Conductor y Reseñas (Vista Pasajero)

- **Historia de Usuario:** Como Pasajero con sesión iniciada explorando conductores, quiero hacer clic en un conductor para ver información más detallada, incluyendo reseñas de otros pasajeros, y poder añadir mi propia reseña para que pueda tomar una decisión informada antes de contactarlo y contribuir con retroalimentación a la comunidad.
- **Criterios de Aceptación:**
 1. **Disparador de Visualización y Diseño:**
 - [UI] Cuando se selecciona un conductor de la lista (HU005-01), esta vista de detalle se hace visible.
 - [UI] Un patrón común es un diseño de dos columnas: la lista de conductores (potencialmente reducida) a la izquierda, y los detalles del conductor seleccionado ocupando la porción derecha de la pantalla.
 - [UI] Incluir una forma de cerrar/descartar la vista de detalle (p.ej., un botón 'X', hacer clic fuera del panel).
 2. **Información Detallada del Conductor:**
 - [FUNC] Obtener datos detallados para el ID de conductor *seleccionado* desde GET /api/drivers/conductores/:id_conductor.

- [UI] Mostrar una versión más grande de la `profile_picture_url`.
- [UI] Mostrar prominentemente `first_name` y `last_name`.
- [UI] Reiterar Calificación Promedio (estrellas y número).
- [UI] Mostrar `joinedDate` ("Miembro desde...").
- [UI] Mostrar detalles completos de vehicles (Marca, Modelo, Año, Color, Capacidad).
- [UI] Opcionalmente mostrar número de viajes completados
- [UI] **Botón Contactar:** Un botón primario "Contactar a [Nombre del Conductor]" debe estar visible. Hacer clic aquí debería probablemente iniciar la interfaz de chat (parte de HU006, pero la *iniciación* ocurre aquí creando una conversación si no existe).

3. Sección de Reseñas:

- [UI] Etiquetar claramente la sección: "Comentarios y Calificaciones".
- [FUNC] Obtener reseñas para este conductor específico desde la API (GET `/api/conductores/:idconductor/reseñas`)
- [UI] Mostrar reseñas en orden cronológico (más nuevas primero).
- [UI] Cada reseña debe mostrar:
 - Nombre del Reseñador (p.ej., "María P." - quizás solo nombre e inicial del apellido por privacidad).
 - Calificación dada (estrellas visuales).
 - Texto del comentario.
 - Fecha de la reseña.
- [UI] Si no existen reseñas, mostrar "Este conductor aún no tiene comentarios."
- [UI] Implementar paginación para reseñas si la lista puede volverse muy larga.

4. Formulario Añadir Reseña:

- [UI] Mostrar una sección/formulario "Deja tu comentario".
- [UI] Incluir una entrada de calificación por estrellas (permitir seleccionar 1-5 estrellas, requerido).
- [UI] Incluir un área de texto para el comentario (requerido, validación de longitud mín/máx útil).

- [UI] Un botón "Enviar Comentario", habilitado solo cuando se proporcionan calificación y comentario.
- [FUNC] Al enviar, mandar los datos de la reseña al backend (POST /api/conductores/:idconductor/reseñas).
- [FUNC] En éxito: Limpiar el formulario, mostrar un mensaje de éxito (p.ej., "Comentario enviado. ¡Gracias!"), y *añadir dinámicamente la nueva reseña al principio de la lista de reseñas mostrada* sin necesidad de refrescar toda la página.

5. API Backend:

▪ Obtener Detalles del Conductor (+ Reseñas Recomendado):

- **Endpoint:** GET /api/conductores/ idconductor
- **Lógica Backend:** Obtener datos del conductor (de tabla conductores), datos del vehículo (de tabla vehiculos), y reseñas asociadas (de tabla reseñas, ordenadas por created_at DESC). Hacer Join reviews con pasajeros (o users) para obtener nombres de reseñadores. Agregar total_viajes. recuperar valor almacenado de promedio de calificacion. Devolver 200 OK con un objeto JSON anidado.
- **Ejemplo Estructura Respuesta:**
- // GET /api/conductores /5
- {
- "driverId": 5,
- "firstName": "Ana", "lastName": "Garcia", /* ... otros campos conductor */
- "averageRating": 4.8,
- "joinedDate": "2024-01-15T10:00:00Z",
- "totalTrips": 52, // Ejemplo si se rastrea
- "vehicle": { /* ... detalles vehículo ... */ },
- "reviews": [
- { "reviewId": 101, "reviewerName": "Carlos R.", "rating": 5, "comment": "Excelente servicio, muy puntual!", "createdAt": "2024-06-10T15:30:00Z" },
- { "reviewId": 95, "reviewerName": "Maria L.", "rating": 4, "comment": "Buen viaje, conductor amable.", "createdAt": "2024-06-08T09:00:00Z" }

- // ... más reseñas (potencialmente paginadas)
-]
- }

HU006-01-mensajes-rutas-programadas: Modulo transporte programado (Vista Conductor)

- **Historia de Usuario:** Como **Conductor con sesión iniciada**, quiero **ver mensajes de pasajeros que me contactaron sobre viajes programados y poder responderles** para que pueda **coordinar detalles, confirmar disponibilidad y acordar el servicio programado**.
- Esta vista sirve como la bandeja de entrada del conductor para el módulo "Transporte Compartido Programado". Facilita la comunicación directa iniciada por pasajeros que encontraron al conductor vía HU005. Debería asemejarse a una interfaz básica de chat o mensajería.
- **Criterios de Aceptación:**
 1. **Diseño de Bandeja de Entrada:**
 - [UI] Típicamente un diseño de dos paneles.
 - **Panel Izquierdo (Lista de Conversaciones):**
 - [FUNC] Obtener la lista de conversaciones de las que el conductor forma parte desde GET /api/conversaciones/id_conductor.
 - [UI] Mostrar cada conversación como un ítem mostrando: Nombre del Pasajero (passenger.firstName + passenger.lastName), fragmento del último mensaje, timestamp del último mensaje.
 - [UI] Resaltar visualmente las conversaciones no leídas (p.ej., texto en negrita, insignia de conteo no leído).
 - [UI] Hacer clic en un ítem de conversación carga sus mensajes en el panel derecho y lo marca como seleccionado/activo.
 - [UI] Si no existen conversaciones, mostrar "No tienes mensajes de pasajeros por ahora."
 - **Panel Derecho (Ventana de Chat):**
 - [UI] Muestra los mensajes para la conversación *actualmente seleccionada*. Mostrar nombre del pasajero arriba.
 - [FUNC] Obtener mensajes para el conversationId seleccionado desde GET /api/conversations/:conversationId/messages.

- [UI] Los mensajes deben mostrarse cronológicamente (más antiguos arriba, más nuevos abajo).
- [UI] Diferenciar visualmente entre mensajes enviados por el conductor y mensajes recibidos del pasajero (p.ej., alineación, color de fondo). Mostrar nombre/avatar del remitente
- [UI] Un área de entrada abajo con un campo de texto para escribir mensajes y un botón "Enviar".

2. Funcionalidad de Mensajes:

- [FUNC] Escribir en el campo de entrada y hacer clic en "Enviar" (o presionar Enter) envía el mensaje vía POST `/api/conversations/:conversationId/messages`.
- [FUNC] El mensaje enviado debe aparecer inmediatamente en la ventana de chat (luego confirmada por backend). Limpiar el campo de entrada después de enviar.
- [FUNC] Deshabilitar botón Enviar si el campo de entrada está vacío.

3. Iniciar Conversaciones (Manejado en Otro Lugar):

- [CONTEXTO] Esta vista principalmente *gestiona* conversaciones existentes. La lógica de iniciación ocurre cuando un pasajero hace clic en "Contactar" en HU005-02. Esa acción debe disparar una llamada backend como POST `/api/conversations/initiate` que verifica si existe una conversación entre el pasajero (del JWT) y el idconductor objetivo; si no, crea un nuevo registro de conversación y registros de participantes.

4. Contratos API Backend:

- **Obtener Mis Conversaciones:**
 - **Endpoint:** GET `/api/conversations/id`
 - **Lógica Backend:** Obtener idconductor. Consultar tabla conversaciones, haciendo join para encontrar conversaciones que incluyen el idconductor. Marcar estado no leído basado en `messages.read_status` y `messages.sender_user_id != driverId`. Devolver 200 OK.
- **Obtener Mensajes para una Conversación:**
 - **Endpoint:** GET `/api/conversaciones/:idconversacion/mensajes`
 - **Lógica Backend:** Verificar que el conductor (del JWT) sea participante en idconversacion. Obtener todos los mensajes para esta conversación, ordenar por `created_at` ASC. Hacer Join con `users` para obtener info del remitente si es necesario.

- **Enviar Mensaje:**
 - **Endpoint:** POST /api/conversacion/:idconversacion/mensajes
 - **Cuerpo de la Solicitud (application/json):** {"messageText": "Hola, sí estoy disponible el viernes."}
 - **Lógica Backend:** Verificar que el conductor (del JWT) sea participante. Validar messageText. Insertar nuevo mensaje (idconveersacion, sender_user_id, message_text, read_status = false).
- **Iniciar Conversación (Ejemplo - Llamado desde HU005-02):**
 - **Endpoint:** POST /api/conversacion/iniciar
 - **Cuerpo Solicitud:** {"targetDriverId": 5}
 - **Lógica Backend:** Obtener idpasajero. Verificar si ya existe una conversación entre idpasajero y idconductor. Si sí, devolver el idconversacion existente. Si no, crear un nuevo registro en conversaciones, crear dos registros en conversation_participants, y devolver el nuevo conversationId

HU006-02-mensajes-rutas-programadas: Modulo transporte programado (Vista Pasajero)

Al hacer clic sobre el botón “Contactar” de la lista de conductores, al usuario tipo pasajero se le desplegara un chat interactivo en donde el usuario podrá contactarse con el conductor, se debe aplicar la misma lógica de iniciación de conversaciones de la historia HU006-02.

