PROYECTO

CHILEANTRUCKS

Sábado 11 Octubre

EQUIPO DE TRABAJO
BASTIAN MOYA
FELIPE HEREDIA

DOCENTE: FERNANDO GONZALO HERRERA FRANCESCONI

SECCIÓN: 707V

EQUIPO DE TRABAJO

BASTIAN MOYA

D. full stack

Security champion

FELIPE HEREDIA

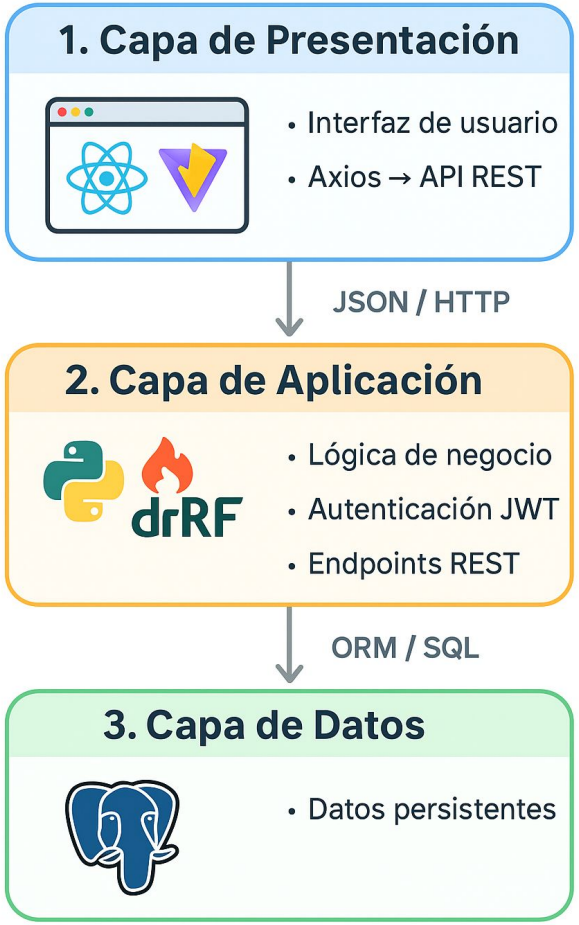
D. full stack

Complience officer

Agenda

- **Arquitectura**
- **Modelo de datos**
- **Casos de uso**
- **Diagramas de secuencia**
- **Compromisos sesión anterior** (y su estado de avance)
- **Avance del Proyecto** (% Sprint y Curva S)
- **Logros de la semana** (Cierre Sprint 2)
- **Próximas actividades** (Kick-off Sprint 3)
- **Explicación del desvío** (Atraso o adelanto)
- **Riesgos (Futuro) e Issues (Pasado)**
- **Compromisos** (Para la próxima sesión)

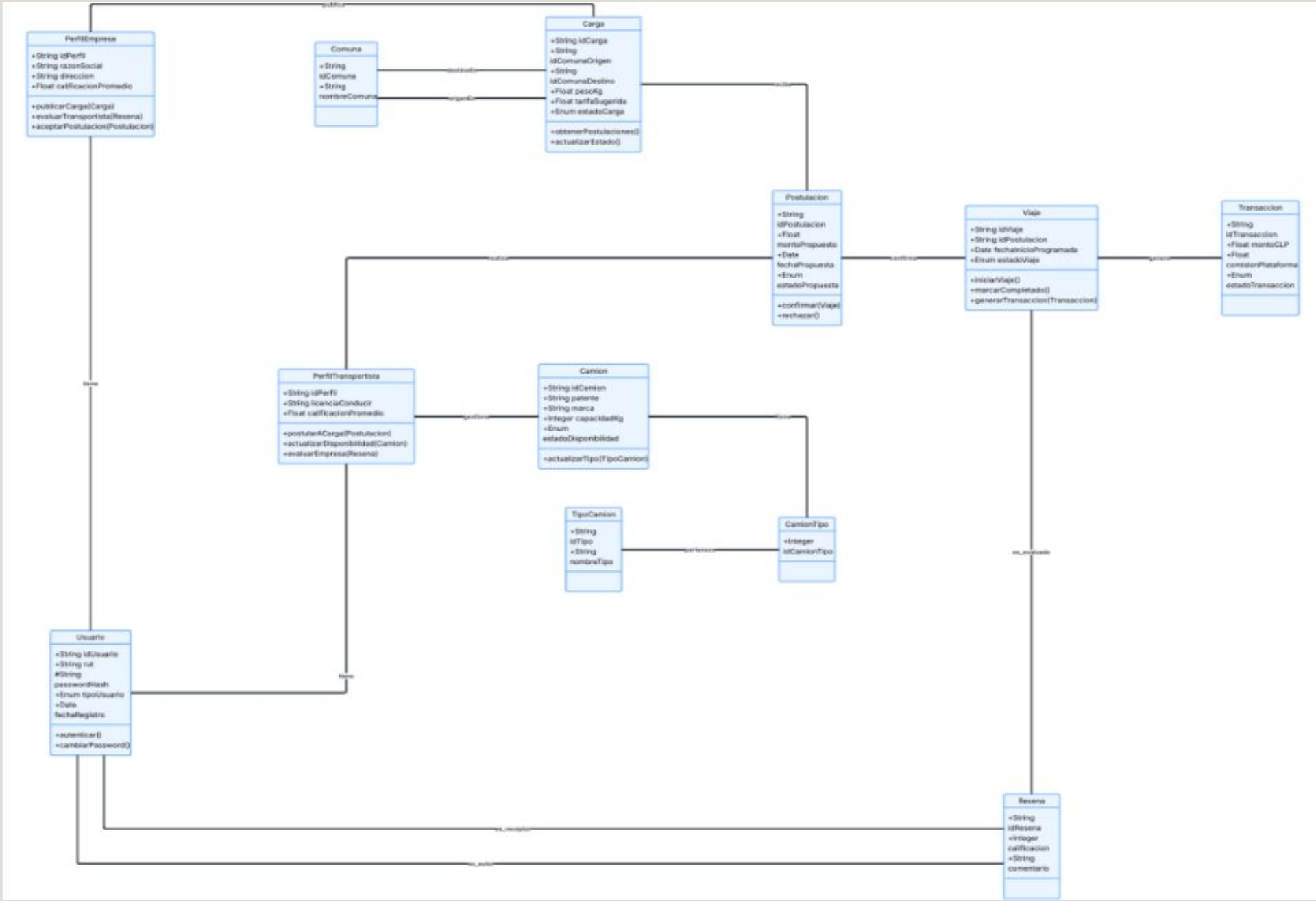
Arquitectura



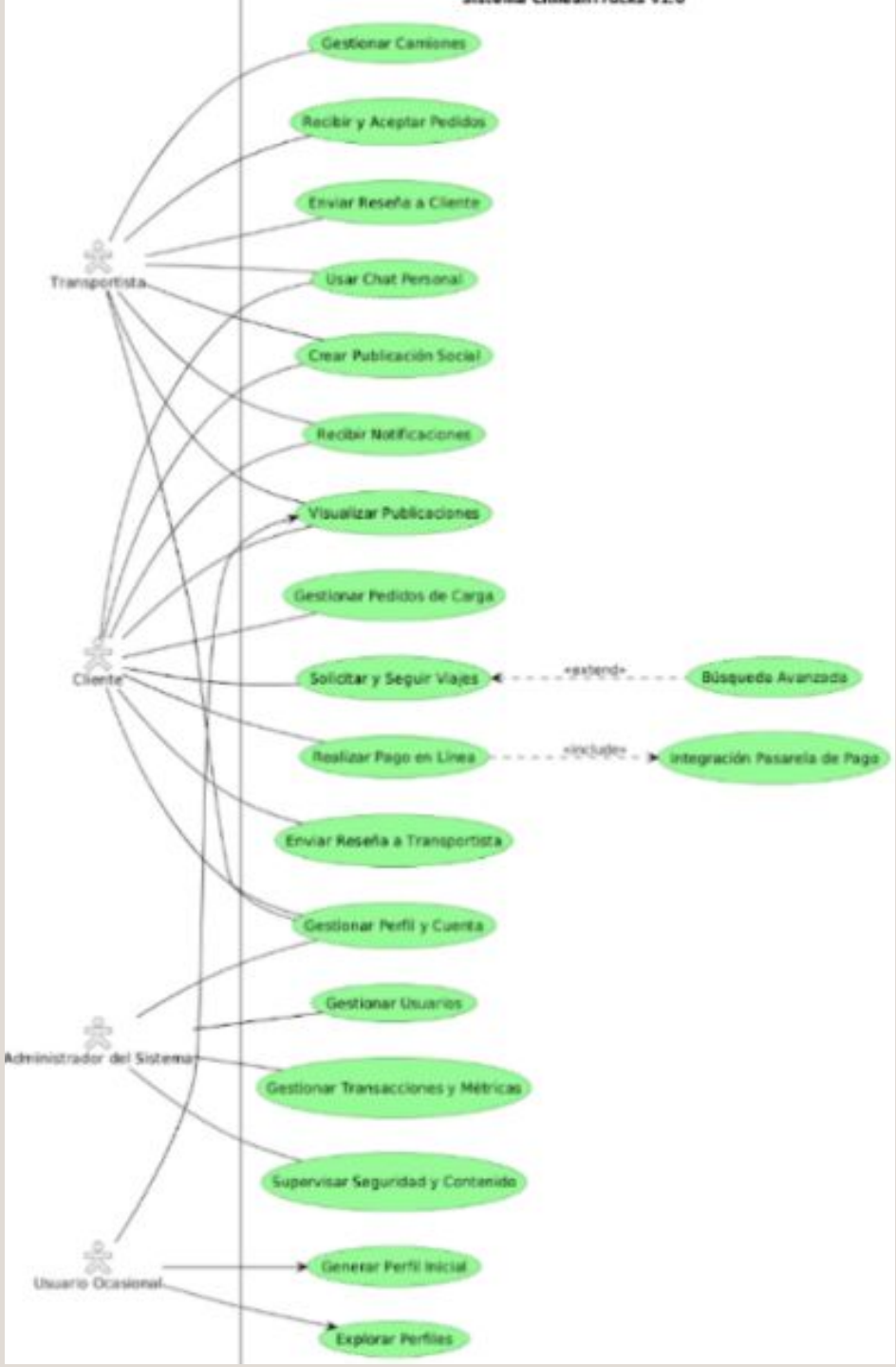
La arquitectura de ChileanTrucks está basada en un modelo de **tres capas desacopladas**. La **capa de presentación** corresponde al frontend desarrollado con **React y Vite**, donde se muestran las interfaces y se gestionan las interacciones del usuario. La **capa de aplicación** está construida con **Django y Django REST Framework**, encargándose de toda la lógica de negocio, validaciones, autenticación mediante **JWT** y exposición de la API REST que consume el frontend. Finalmente, la **capa de datos** está compuesta por una base de datos **PostgreSQL**, donde se almacena toda la información persistente del sistema.

Estas capas se comunican de manera ordenada: el frontend envía solicitudes HTTP (JSON) al backend; este procesa la información, accede a la base de datos mediante el ORM de Django y devuelve respuestas estructuradas al frontend. Este diseño desacoplado permite mantener, escalar y desarrollar cada capa de forma independiente, garantizando un sistema organizado y flexible

Modelo de datos



Casos de uso


Gestionar Camiones
Recibir y Aceptar Pedidos
Enviar Reseña a Cliente
Usar Chat Personal
Crear Publicación Social
Recibir Notificaciones
Visualizar Publicaciones
Gestionar Pedidos de Carga
Solicitar y Seguir Viajes
«extend»
Búsqueda Avanzada
Realizar Pago en Línea
«include»
Integración Pasarela de Pago
Enviar Reseña a Transportista
Gestionar Perfil y Cuenta
Gestionar Usuarios
Gestionar Transacciones y Métricas
Supervisar Seguridad y Contenido
Generar Perfil Inicial
Explorar Perfiles
Transportista
Cliente
Administrador del Sistema
Usuario Ocasional

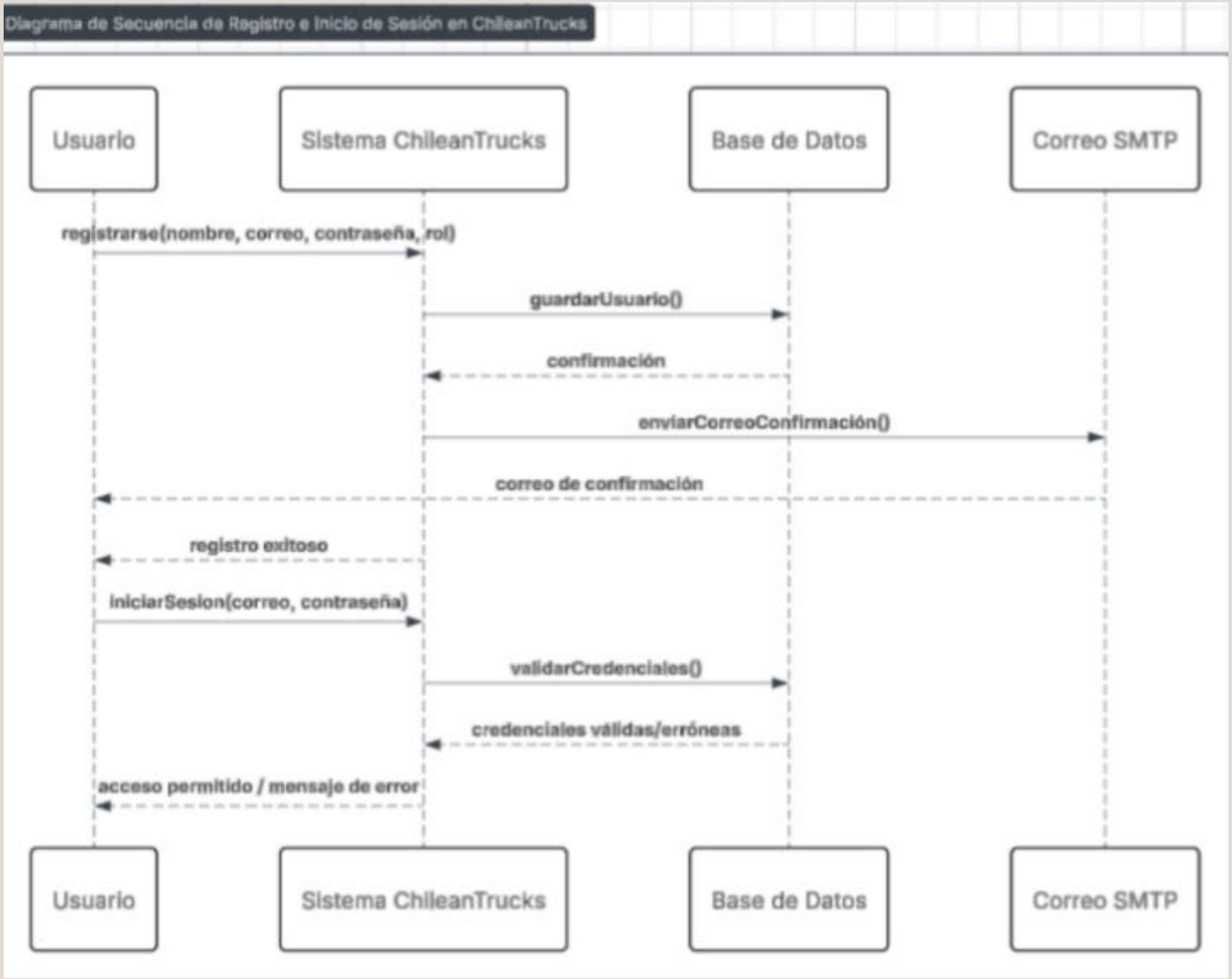
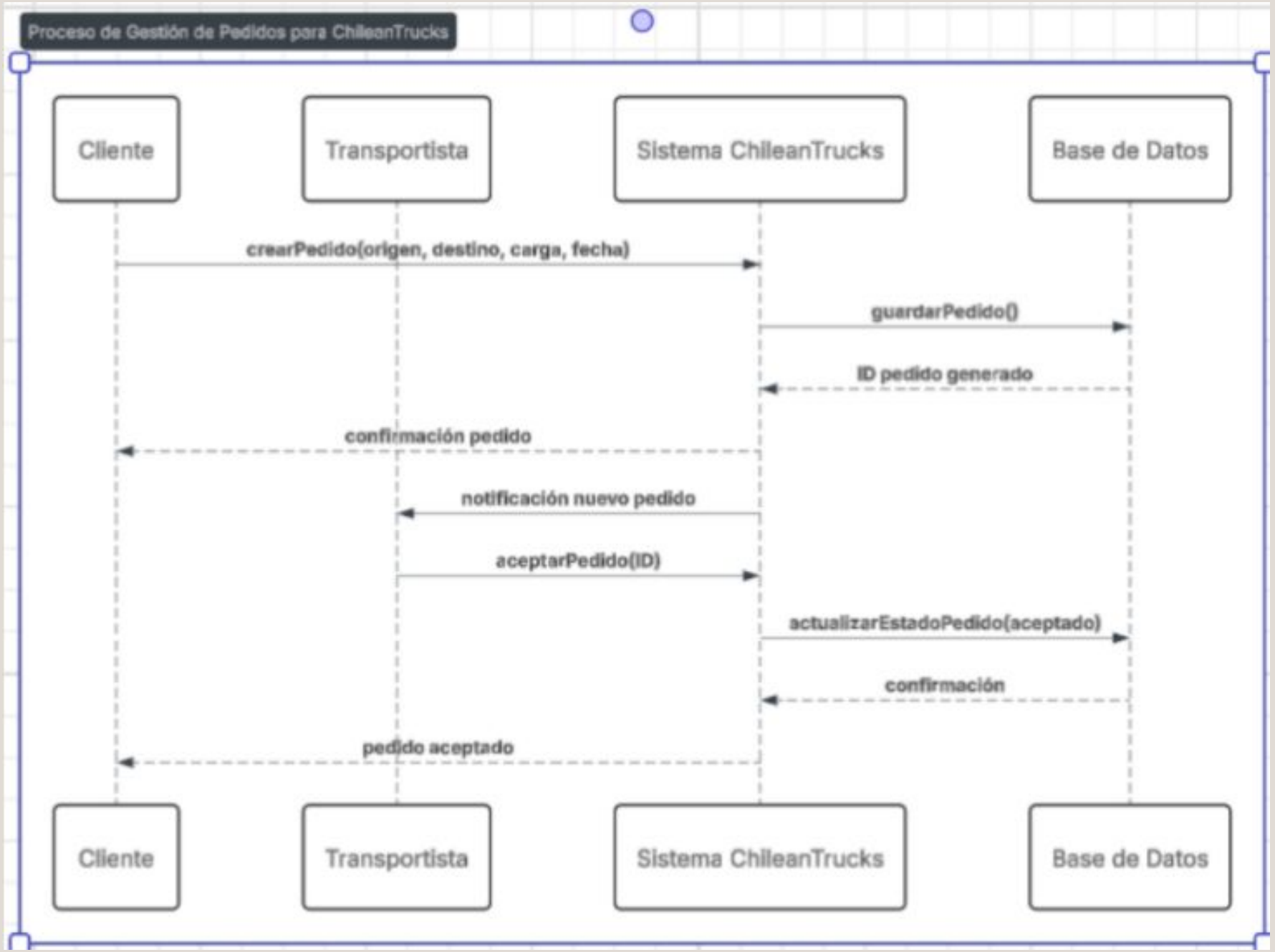


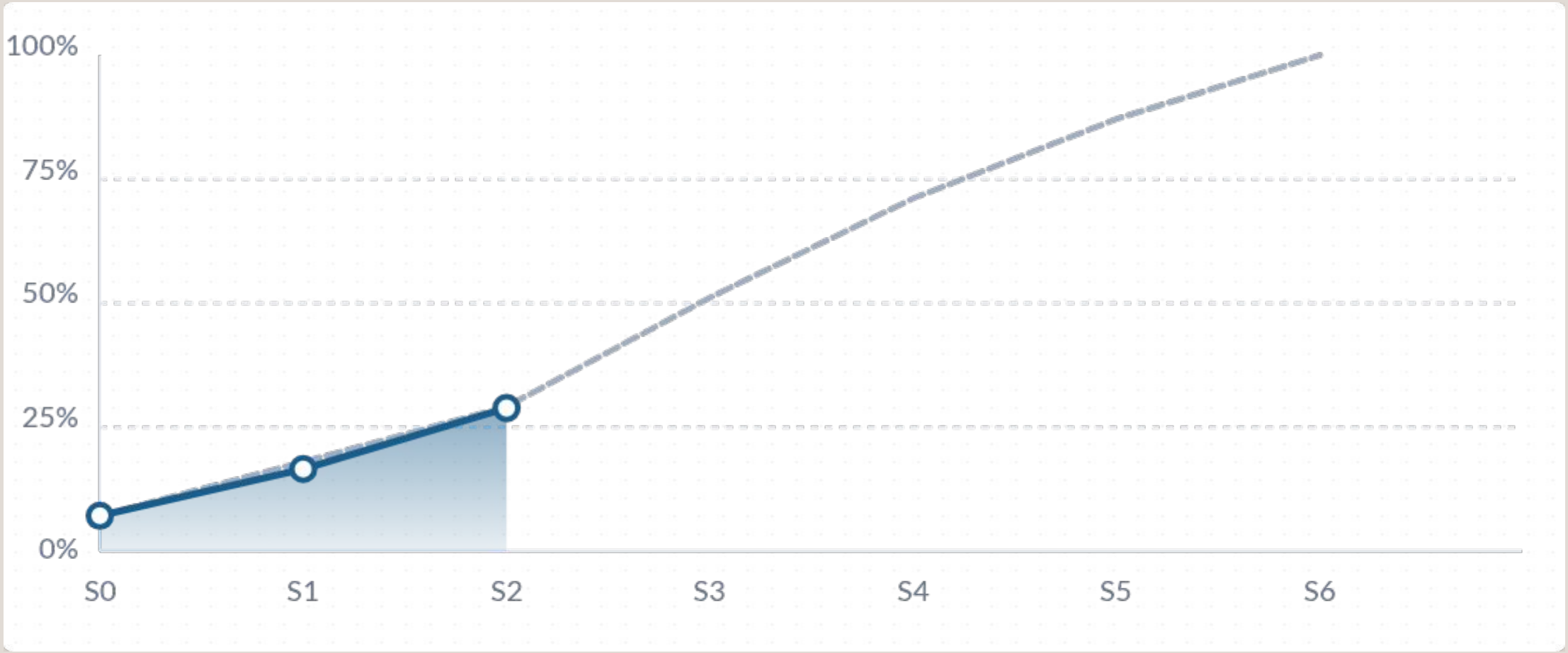
Diagrama de Secuencia de Registro e Inicio de Sesión en ChileanTrucks
Usuario
Sistema ChileanTrucks
Base de Datos
Correo SMTP
registrarse(nombre, correo, contraseña, rol)
guardarUsuario()
confirmación
enviarCorreoConfirmación()
correo de confirmación
registro exitoso
iniciarSesion(correo, contraseña)
validarCredenciales()
credenciales válidas/erróneas
acceso permitido / mensaje de error
Usuario
Sistema ChileanTrucks
Base de Datos
Correo SMTP


Proceso de Gestión de Pedidos para ChileanTrucks
Cliente
Transportista
Sistema ChileanTrucks
Base de Datos
crearPedido(origen, destino, carga, fecha)
guardarPedido()
ID pedido generado
confirmación pedido
notificación nuevo pedido
aceptarPedido(ID)
actualizarEstadoPedido(aceptado)
confirmación
pedido aceptado
Cliente
Transportista
Sistema ChileanTrucks
Base de Datos

Estado de Compromisos (Sprint 2: 23 Sep – 13 Oct)

Compromiso (de la sesión del 4/Oct)	Estado
[Admin] Finalizar el 100% del `admin.py` y Pruebas de Humo	Finalizado
[Investigación] Presentar plan de implementación de JWT	Finalizado
[Investigación] Presentar hallazgos y plan de Stripe	Finalizado
[Planificación] Presentar el Sprint Planning detallado del Sprint 3	Finalizado

Avance de Tareas



Avance Sprint 2 (Backend Base): 100% completado (5 de 5 tareas). Avance Total del Proyecto: 28.9% (13 de 45 tareas).

3. Logros de la Semana (Cierre Sprint 2)

Sprint 2 Finalizado

El Sprint 2 (Backend Base) se completa 100%, 2 días antes de la fecha (13/Oct). La base de datos, modelos y admin están sólidos.

Pruebas de Humo (Admin) OK

Se realizó un Smoke Test manual en el Admin de Django, creando un flujo completo (Usuario -> Camión -> Carga) con éxito.

Riesgos S3 Mitigados

Se completó la investigación y el plan de acción para JWT (Auth) y Stripe (Pagos), reduciendo la incertidumbre técnica del S3.

Chilean Trucks

4. Próximas Actividades (Kick-off Sprint 3: 14 Oct – 27 Oct)

Autenticación y CRUDs

Implementar Simple JWT para Registro y Login. Desarrollar ViewSets y Serializers para CRUD de Camiones y Cargas.

Lógica Core (Postulación)

Implementar la lógica de Postulación (validación de peso y tipo) y el Ciclo de Vida del Viaje (iniciar, finalizar, aprobar).

Integraciones (Stripe, GPS)

Implementar el endpoint de pago con Stripe (Payment Intent) y el endpoint que recibe la geolocalización (GPS) del transportista.

5. Desvíos, 6. Issues y Riesgos

5. Explicación del Desvío

+2

Días (Adelantado)

El Sprint 2 se cierra hoy (11/Oct), 2 días antes de la fecha límite (13/Oct).
El equipo utilizó esta holgura para mitigar riesgos del Sprint 3.

6. Gestión de Riesgos e Issues

ISSUE (Cerrado)

Issue: (Riesgo S2) El Sprint 3 era muy denso y con alta incertidumbre técnica.
Nivel: Alto (Cerrado)
Acción (Corregido): Se investigó y prototipó JWT y Stripe antes de iniciar el S3. El riesgo se reduce de "desconocido" a "planificado".
Fecha Comprometida: 11/Oct (Completado)

RIESGO (Futuro)

Riesgo: La **Lógica de Postulación** (validar peso/tipo) y el **Ciclo de Vida del Viaje** (múltiples estados) es la lógica de negocio más compleja de la API.
Nivel de Riesgo: Medio
Plan de Mitigación: Priorizar el desarrollo de pruebas unitarias (`tests.py`) para estas funciones antes de exponerlas en los ViewSets.
Fecha Comprometida: 18/Oct (Próxima sesión)

7. Compromisos (Para Sábado 18 Oct)

- **[Backend]** Implementar Autenticación (JWT) 100%: Endpoints de Registro, Login y Logout funcionales.
- **[Backend]** Finalizar el CRUD (ViewSet y Serializer) de Camiones (Transportista).
- **[Backend]** Finalizar el CRUD (ViewSet y Serializer) de Cargas (Empresa).
- **[QA]** Probar con Postman los endpoints de Auth, Camiones y Cargas, y presentar la colección de Postman.