

# PROYECTO CHILEANTRUCKS

Sábado 27  
septiembre



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  
Compliance officer

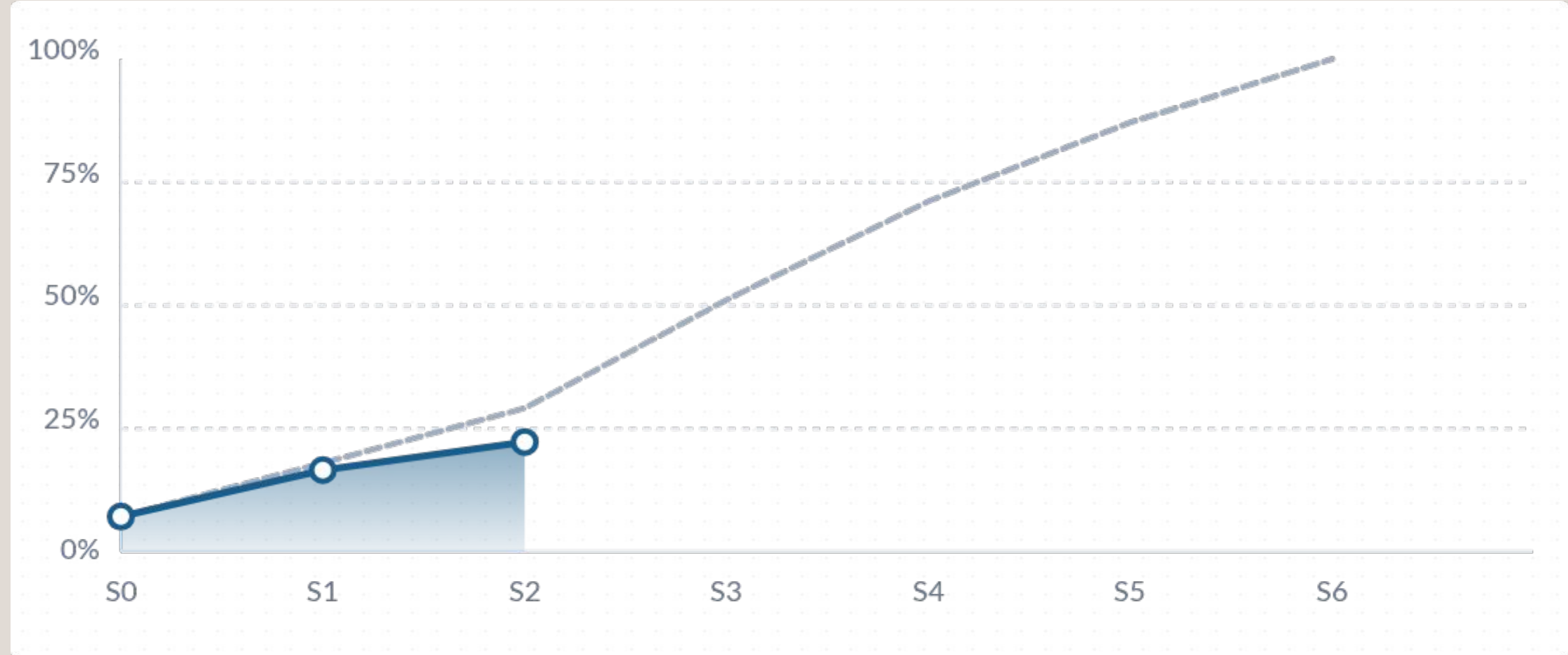
# Agenda

- 1 **Compromisos sesión anterior** (y su estado de avance)
- 2 **Avance del Proyecto** (% Sprint y Curva S)
- 3 **Logros de la semana** (Inicio Sprint 2)
- 4 **Próximas actividades** (Desarrollo Sprint 2)
- 5 **Explicación del desvío** (Atraso o adelanto)
- 6 **Riesgos (Futuro) e Issues (Pasado)**
- 7 **Compromisos** (Para la próxima sesión)

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

Compromiso (de la sesión del 20/Sep)	Estado
[Backend ] Proyecto Django (DRF, CORS) 100% configurado	Finalizado
[BD ] Conexión a PostgreSQL 100% operativa	Finalizado
[BD ] Modelos principales (Usuario, Camion, Carga) implementados en `models.py`	En Progreso (80%)
[Admin ] Configuración básica del Admin de Django (`admin.py`)	En Progreso (50%)
[Arquitectura] Finalizar documentación Vistas 4+1 (Arrastrado S1)	Finalizado

# Avance de Tareas



Avance Sprint 2 (Backend Base): 40% completado (2 de 5 tareas). Avance Total del Proyecto: 22.2% (10 de 45 tareas).

## 3. Logros de la Semana (Inicio Sprint 2)



### Proyecto Django Operativo

El proyecto backend está 100% configurado, incluyendo Django REST Framework, CORS, y variables de entorno (.env).



### PostgreSQL Conectado

La conexión a la base de datos PostgreSQL local está finalizada. Las migraciones iniciales de Django se ejecutaron correctamente.



### Implementación de Modelos

Se implementaron los modelos principales (Usuario, Camión, Carga) en `models.py`, incluyendo un `CustomUser` basado en `AbstractUser`.

## 4. Próximas Actividades (Desarrollo Sprint 2)



### Finalizar Modelos

Completar la implementación de los modelos restantes: Postulación, Reseña y Mensaje, con todas sus relaciones (ForeignKeys).



### Carga Masiva de Comunas

Crear la "data migration" de Django para leer el CSV de comunas y cargarlas a la base de datos automáticamente.



### Completar Admin de Django

Registrar todos los modelos en `admin.py` para facilitar la gestión y prueba de datos durante el desarrollo.

# 5. Desvíos, 6. Issues y Riesgos



## 5. Explicación del Desvío

0

### Días de Desvío

El equipo avanza según la planificación. Las tareas 'En Progreso' (Modelos, Admin) son las actividades principales de la semana y se completarán según el plan del sprint.

## 6. Gestión de Riesgos e Issues

### 🚩 ISSUE

**Issue:** Documentación de Arquitectura (Vistas 4+1) arrastrada del S1.

**Nivel:** Bajo

**Acción (Corregido):** Finalizada en paralelo el Lunes 22/Sep.

**Fecha Comprometida:** 22/Sep (Completado)

### ⚠️ RIESGO

**Riesgo:** La Carga Masiva de Comunas (CSV) puede fallar o tener inconsistencias de encoding.

**Nivel de Riesgo:** Medio

**Plan de Mitigación:** 1. Validar encoding (UTF-8) del CSV. 2. Usar `bulk\_create` de Django para eficiencia. 3. Probar en una rama separada.

**Fecha Comprometida:** 4/Oct (Próxima sesión)



## 7. Compromisos (Para Sábado 4 Oct)

- ✓ **[BD]** Tener el 100% de los modelos (Usuario, Camion, Carga, Postulación, Reseña, Mensaje) implementados en `models.py` y migrados.
- ✓ **[BD]** Finalizar y probar la migración de datos para la Carga Masiva de Comunas.
- ✓ **[Admin]** Tener todos los modelos registrados en `admin.py` y funcionales (poder crear y editar datos desde el panel de admin).