



Plan de trabajo

El siguiente plan de trabajo tiene la finalidad de organizar y dirigir las actividades del equipo HackstreetBoys para el desarrollo de una solución integral de monitoreo de tráfico aéreo en tiempo real con la utilización de Zabbix.

Descripción general

Nombre del equipo: HackstreetBoys Fecha de entrega: Nov 7, 2025

Nombres	Roles	Responsabilidades
Perez Gomez Pamela Jhatdzarhy	Líder Técnico / Zabbix	Configurar Zabbix Cloud, crear hosts, ítems, triggers, automatizaciones, webhooks.
Pérez Medina Juan Manuel	Backend / API y Base de Datos	Consumir OpenSky API, procesar datos, guardarlos en BD, exponer API interna.
Jiménez Valencia Gustavo Efrén	Integraciones y Tiempo Real	Conectar Zabbix ↔ Backend ↔ Frontend. Manejo de WebSockets o actualización periódica.
Galicia Murrieta Miguel Angel	Frontend Web	Crear interfaz web para mostrar vuelos, estadísticas e indicadores.
Ruiz Pablo Cristina Naydelin	Documentación y Presentación	Evidencias, capturas, README, demo final, pitch corto.

Cronograma y tareas

	11:00-12:30	12:30-15:30	15:30-19:00	19:00-22:00
<div><div>Fase 1</div><div>Se llevará a cabo la reunión inicial. Además de llevarse a cabo las configuraciones base.</div></div>	<div>Revisión de proyecto, asignación de roles, creación de repositorio en GitHub</div>	<div>Backend: Crear script para consumir OpenSky API → guardar datos en BD.</div> <div>Zabbix: Crear host, importar plantilla genérica → definir ítems: conteo de vuelos, disponibilidad API.</div> <div>Frontend: Crear estructura y diseño inicial (tablero, mapa o lista).</div> <div>Integraciones: Preparar sockets o endpoint</div> <div>Documentador: Crear README inicial, esquema general.</div>		
<div><div>Fase 2</div><div>Desarrollo principal</div></div>			<div>Backend: Exponer API REST: /vuelos, /estadisticas, /estado_api.</div> <div>Zabbix: Crear triggers: API caída, vuelos > X, limpieza automática BD.</div> <div>Frontend: Mostrar tabla de vuelos + contador + indicador de estado.</div> <div>Integraciones: Conectar frontend ↔ backend (fetch o axios) y preparar recepción de eventos.</div> <div>Documentador: Tomar capturas, registrar comandos, crear guía de despliegue.</div>	
<div><div>Fase 3</div><div>Automatizaciones y Tiempo Real</div></div>				<div>Zabbix envía webhook al backend cuando ocurre un trigger.</div> <div>Backend notifica al frontend (socket o bandera en BD).</div> <div>Frontend actualiza interfaz</div> <div>Revisión de avances, errores y búsqueda de soluciones</div>

	08:00-09:00	09:00-10:00	10:00-12:00
Fase 4 Pruebas y ajustes.	<div>Se simula caída de API (desconectar internet o endpoint falso).</div> <div>Se simula exceso de vuelos (insertar datos de prueba).</div> <div>Ajustar colores, mensajes, recargas y diseño.</div>		
Fase 5 Documentación y Evidencias		<div>● Zabbix Cloud → triggers activos.</div> <div>● Web App funcionando.</div> <div>● Base de datos recibiendo datos.</div>	
Fase 6 Preparación de Presentación Final			<div>Qué se monitorea con Zabbix + por qué</div> <div>Cómo fluye la información (API → BD → Zabbix → Web)</div> <div>Demo + resultados</div>