

**Plan de Proyecto “Panorama”**

# Índice

[**1. Índice 2**](#_heading=h.l550ubmt7cgl)

[**2. Roles del equipo de desarrollo 3**](#_heading=h.db4feth1jtk4)

[**3. Metodología del proyecto 3**](#_heading=h.1l50os2mm20p)

[**4. Cronograma 4**](#_heading=h.nc6vqykqcwmi)

[**5. Recursos 6**](#_heading=h.t3wtpjkli02d)

[**6. Riesgos Identificados 6**](#_heading=h.c64k2hklx2dt)

[**7. Posibles mitigaciones 7**](#_heading=h.3282fyi22scm)

[**8. Anexos 7**](#_heading=h.pa21s0o3pvd)

# Roles del equipo de desarrollo

Javier Vega: Desarrollador - Jefe de proyecto

Moises Figueroa: Desarrollador - Tester - Jefe de proyecto

Zenón Jara: Desarrollador - Tester

# Metodología del proyecto

En este proyecto, hemos elegido la metodología en cascada como enfoque principal para el desarrollo de la plataforma Panorama. Esta decisión se basa en las siguientes características clave del proyecto:

* **Requisitos bien definidos:**
  + Contamos con una Especificación de Requerimientos (ERS) clara desde el inicio, lo que permite un flujo estructurado sin cambios frecuentes.
* **Entregables documentados en cada fase:**
  + Cada etapa (análisis, diseño, implementación, pruebas y despliegue) genera documentos técnicos concretos (diagramas UML, modelo de BD, manuales), lo que facilita la trazabilidad y el control de calidad.
* **Proyecto con alcance acotado:**
  + Al ser una plataforma web con funcionalidades específicas (gestión de eventos, usuarios y búsquedas), la cascada evita distracciones por iteraciones innecesarias.

**¿Por qué no Ágil?**

Si bien metodologías como Scrum son populares, la cascada es más adecuada para este caso porque:

* No hay stakeholders externos requiriendo cambios continuos.
* El equipo es pequeño (3 personas) y trabaja con objetivos fijos.
* Los hitos son documentales (ej: ERS validado, diseño de BD aprobado), no funcionales.

**Adaptaciones realizadas**

Para asegurar flexibilidad, incorporamos:

* Revisiones técnicas semanales: Para detectar riesgos tempranos (ej: retrasos en el diseño de BD).

# 

# Cronograma

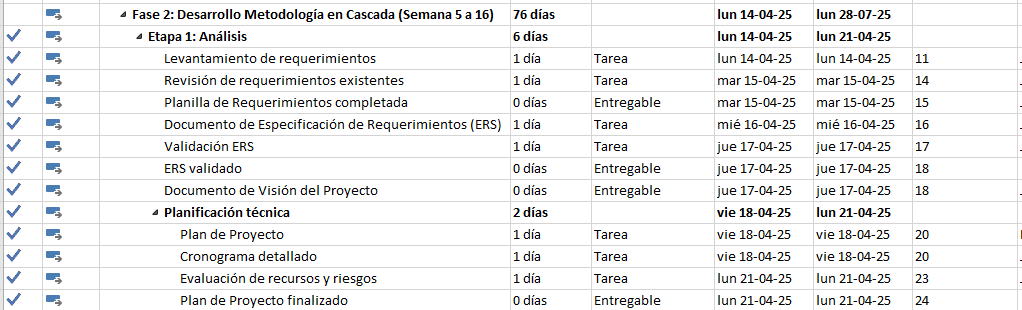
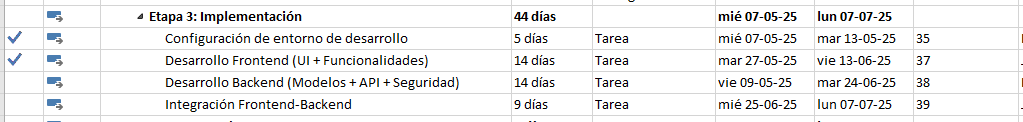
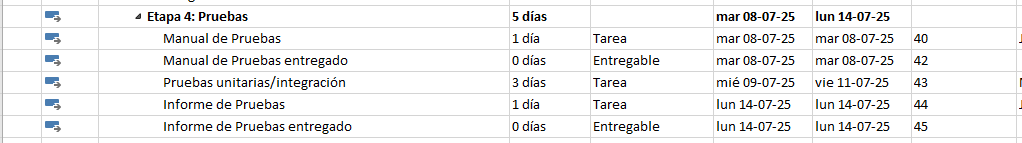
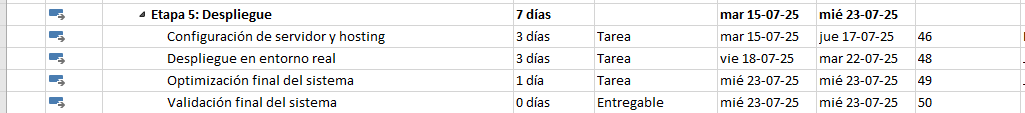
* **Fase 1: Planificación**

En esta etapa se organiza el equipo, se define la idea principal del proyecto, se realiza la propuesta inicial y se construye la planificación general (como la carta Gantt). También se presenta el proyecto y se entregan los primeros informes de avance.



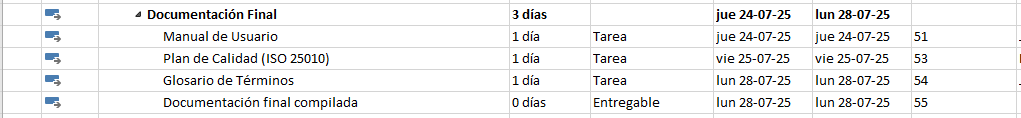
* **Fase 2: Desarrollo**

Esta es la parte más larga y central del proyecto, y está dividida en cinco etapas, siguiendo la metodología en cascada:

1. **Análisis**Se levantan y documentan los requisitos del sistema (ERS), se validan con el equipo y se deja como base oficial del desarrollo.  
   
2. **Diseño** Se crean los diagramas UML, vistas del sistema, modelo de datos y arquitectura general. Aquí se diseña cómo funcionará el sistema antes de programar.  
   
3. **Implementación** Se configura el entorno, se programa el backend y frontend, y luego se integran ambos. Es la etapa de desarrollo activo del software.  
   
4. **Pruebas** Se documentan las pruebas, se prueban todas las funciones del sistema y se entrega un informe con los resultados.  
   
5. **Despliegue** Se configura el servidor, se publica el sistema en línea, se optimiza y se valida su funcionamiento final.  
   

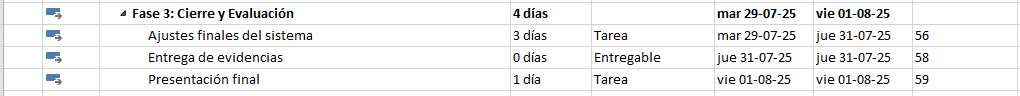
* **Documentación Final**

Se preparan los documentos clave como el Manual de Usuario, Plan de Calidad y Glosario. Se compilan todos los documentos para su entrega formal.



* **Fase 3: Cierre y Evaluación**

Se realizan los últimos ajustes, se entrega evidencia del trabajo y se expone el proyecto en la presentación final.



# 

# Recursos

Para el desarrollo del proyecto “Panorama”, se consideran los siguientes recursos:

**Recursos Humanos:**

* Javier Vega – Parte del equipo de desarrollo.
* Moises Figueroa – Parte del equipo de desarrollo.
* Zenón Jara – Parte del equipo de desarrollo.
* Profesores – Validadores y guía de todo el proceso.

**Recursos Técnicos:**

* Computadores personales con conexión a internet.
* Repositorio en GitHub para control de versiones.
* Herramientas de desarrollo:
  + Visual Studio Code (editor de código)
  + Datamodeler (modelo ER y lógico)
  + Figma (diseño de interfaz)
  + Postman y Swagger (pruebas de API)
  + Bun.js, Next.Js y SupaBase(PostgresQL).

# Riesgos Identificados

Para la evaluación de riesgos, se utiliza una escala cualitativa basada en el nivel de impacto y la probabilidad de ocurrencia:

**Crítico (Alto)**

* Bloquea el avance o genera costos significativos.
* Ej: Base de datos mal diseñada → Rediseño total, 2 semanas perdidas.

**Significativo (Medio-Alto)**

* Afecta plazos o calidad, pero es controlable.
* Ej: Bugs en API → Retraso de 3 días en integración.

**Moderado (Medio)**

* Impacto limitado, requiere ajustes menores.
* Ej: Cambios en diseño de UI → 1 día de retrabajo.

**Menor (Bajo)**

* Mínima afectación, se puede resolver con recursos actuales.
* Ej: Error ortográfico en el manual → Corrección rápida.

**Posibles riesgos identificados:**

* Retrasos en el diseño de la base de datos.
* Bugs críticos durante la integración del sistema.
* Cambios inesperados en los requisitos del proyecto.
* Falta o baja disponibilidad de recursos humanos.
* Fallos en el servidor de hosting.
* Baja adopción del sistema por parte de los usuarios.

# Posibles mitigaciones

Las posibles mitigaciones encontradas son las siguientes:

* Validar el modelo entidad-relación en la primera semana con el equipo.
* Usar herramientas visuales como MySQL Workbench para prototipado rápido.
* Implementar pruebas unitarias desde el inicio (ej. con Jest).
* Usar integración continua (GitHub Actions) para detectar errores tempranamente.
* Establecer congelación (no realizar cambios) del ERS después de su validación formal.
* Canalizar cambios mediante solicitudes oficiales con análisis de impacto.
* Priorizar tareas críticas.
* Configurar respaldos automáticos periódicos (cada 2 o 3 días) del servidor.
* Hacer pruebas de usabilidad tempranas con usuarios.

# Anexos

* [Cronograma](https://drive.google.com/file/d/1aN6FyQSmyyOspndox2gVoIeuw-rjWGLc/view?usp=sharing)