|  | Esta pauta tiene como objetivo ayudarte a monitorear el desarrollo de tu Proyecto APT, reflexionando sobre tus avances de acuerdo con lo planificado en la fase anterior y recibiendo retroalimentación de tus pares y docentes que te permita hacer los ajustes necesarios para cumplir con los objetivos de tu proyecto. Esta pauta debe ser respondida con tu grupo. |
| --- | --- |

| Puedes completar esta guía y, posteriormente, cargarla en la sección de reflexión de la Fase 2, para retroalimentación de tu docente. |
| --- |

| 1. Mira tu carta Gantt y reflexiona sobre los avances de tu Proyecto APT |
| --- |
| Actualmente me encuentro en la Semana 1 del cronograma (Fase de Setup Inicial) y estoy avanzando según lo planificado, aunque con algunos ajustes menores. He completado aproximadamente el 70% de las actividades programadas para esta primera semana:  Definición completa del stack tecnológico (Next.js 14, TypeScript, PostgreSQL, Prisma)  Diseño de arquitectura del sistema  Modelado de base de datos (9 entidades principales)  Estructura completa de carpetas del proyecto  Documentación de requerimientos y funcionalidades  Análisis de mercado y competencia  Configuración inicial de archivos del proyecto |

| 2. |
| --- |
| ¿De qué manera has enfrentado y/o planeas enfrentar las dificultades que han afectado el desarrollo de tu Proyecto APT?  Estrategias implementadas:   1. Para la curva de aprendizaje:    * Estoy dedicando tiempo específico a documentación oficial    * Creación de documentación propia mientras aprendo    * Enfoque en aprender haciendo (implementar mientras estudio)    * Priorizar características MVP antes que features avanzadas 2. Para problemas técnicos (compatibilidad Windows):    * Adaptación de scripts bash a batch (.bat) para Windows    * Documentación de comandos específicos para Windows    * Uso de PowerShell como alternativa cuando sea necesario    * Considerar uso de WSL (Windows Subsystem for Linux) si persisten problemas 3. Para decisiones técnicas:    * Análisis comparativo documentado (ej: Next.js vs FastAPI)    * Consulta con profesores y pares    * Priorizar soluciones que optimicen tiempo de desarrollo    * Enfoque pragmático: "lo que funcione mejor para el plazo"   Plan para enfrentar dificultades futuras:   * Semana 2-3: Si encuentro dificultades con implementaciones complejas, simplificar el alcance del MVP * Buffer de tiempo: He considerado 1 semana de colchón en mi cronograma para imprevistos * Validación temprana: Probar cada módulo con mi padre apenas esté funcional * Pivotear si es necesario: Si alguna tecnología resulta muy compleja, buscar alternativas (ej: si Google Maps API es muy cara, usar OpenStreetMap) |

| 3. Hasta el momento: |
| --- |
| ¿Cómo evalúas tu trabajo? ¿Qué destacas y qué podrías hacer para mejorar tu trabajo?  Evaluación general: 7.5/10  Aspectos positivos que destaco:  Planificación sólida:   * Carta Gantt realista con hitos claros * Análisis de mercado completo * Arquitectura bien pensada antes de programar   Documentación exhaustiva:   * Todo está documentado desde el inicio * Decisiones técnicas justificadas * Será muy útil para la memoria de título   Enfoque pragmático:   * Priorización de MVP sobre features "nice to have" * Stack elegido pensando en tiempo de desarrollo * Cliente real disponible para validación   Valor comercial claro:   * Problema real identificado * Diferenciación vs competencia definida * Modelo de negocio pensado   Aspectos a mejorar:  Velocidad de ejecución:   * He dedicado mucho tiempo a planificación (necesario pero extenso) * Necesito pasar más rápido a código funcional * Meta: Tener algo visual funcionando para fin de semana 2   Balance teórico-práctico:   * Debo equilibrar mejor tiempo de investigación vs implementación * Aplicar principio "done is better than perfect"   gestión de scope creep:   * Tengo muchas ideas de features, debo mantener foco en MVP * Crear lista de "Fase 2" para features post-MVP   Plan de mejora:   1. Esta semana: Terminar setup y tener el proyecto corriendo localmente 2. Próxima semana: Primera funcionalidad completamente operativa (CRUD vehículos) 3. Implementar regla: "No más de 1 día en investigación sin avance de código" 4. Validación semanal: Mostrar avances a mi padre cada viernes |

| 4. Después de reflexionar sobre el avance de tu Proyecto APT |
| --- |
| ¿Qué inquietudes te quedan sobre cómo proceder? ¿Qué pregunta te gustaría hacerle a tu docente o a tus pares?  Inquietudes técnicas:   1. ¿Es realista implementar tracking GPS en tiempo real en el MVP?    * O debería dejarlo como actualización manual de ubicación inicialmente    * ¿Qué tan complejo es integrar Google Maps API vs alternativas gratuitas? 2. ¿Cómo manejar el tema multi-tenant desde el inicio?    * ¿Debería construir toda la lógica de separación por empresa desde la primera funcionalidad?    * O ¿es mejor empezar single-tenant y migrar después? 3. Autenticación y seguridad:    * ¿NextAuth.js es suficiente o debería considerar algo más robusto?    * ¿Qué nivel de seguridad se espera para un proyecto de título?   Inquietudes de alcance:   1. ¿El MVP planificado es realista para 3 meses?    * Tengo 5 módulos principales (flotas, rutas, entregas, facturación, mantenimiento)    * ¿Es preferible hacer 3 módulos muy bien o 5 módulos básicos? 2. ¿Cuánto peso tiene la "novedad técnica" vs "solución funcional"?    * ¿Es mejor usar tecnologías muy nuevas (Next.js 14) o más estables?    * ¿Se valora más la innovación o la robustez? |

| 5. A partir de esta instancia de monitoreo de su Proyecto APT |
| --- |
| ¿Consideran que las actividades deben ser redistribuidas entre los miembros del grupo? ¿Hay nuevas actividades que deban ser asignadas a algún miembro del grupo?  Este es un proyecto individual, por lo que no aplica redistribución de tareas entre miembros.  Sin embargo, he identificado áreas donde podría buscar colaboración externa:  Posibles colaboraciones:   1. Testing con usuarios:    * Mi padre y sus trabajadores (empresa piloto)    * Otros transportistas para validar generalización del sistema 2. Revisión de código:    * Compañeros con experiencia en tecnologías similares    * Profesores para validación de arquitectura 3. Diseño UX/UI:    * Si el tiempo lo permite, podría consultar con alguien de diseño    * Aunque shadcn/ui ya proporciona componentes bien diseñados 4. Testing de la aplicación:    * Compañeros para probar funcionalidades como beta testers    * Identificar bugs y mejoras de usabilidad |

| 6. APT grupal |
| --- |
| ¿Cómo evalúan el trabajo en grupo? ¿Qué aspectos positivos destacan? ¿Qué aspectos podrían mejorar?  Aunque es un proyecto individual, puedo reflexionar sobre mi autodisciplina y organización:  Aspectos positivos de mi metodología:  Organización:   * Uso de documentación estructurada * Versionado de código (Git) * Cronograma detallado con hitos medibles   Comunicación con stakeholder:   * Contacto regular con mi padre (cliente) * Validación constante de requerimientos * Expectativas alineadas   Proactividad:   * Busco resolver dudas activamente * Investigo soluciones antes de estancarme * Documentación a medida que avanzo   Aspectos a mejorar en mi forma de trabajo:  Gestión del tiempo:   * A veces dedico demasiado tiempo a "perfeccionar" cosas pequeñas * Necesito ser más estricto con timeboxing de tareas   Balance vida-proyecto:   * Debo establecer horarios específicos para no saturarme * Tomar descansos regulares para mantener productividad   Pedir ayuda a tiempo:   * Tendencia a querer resolver todo solo * Debo aprovechar más a profesores y compañeros cuando me estanque |