|  |  |
| --- | --- |
|  | 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 |
| ¿Has podido cumplir todas las actividades en los tiempos definidos? ¿Qué factores han facilitado o dificultado el desarrollo de las actividades de tu plan de trabajo?  Hasta la fecha, he logrado cumplir aproximadamente el **85% de las actividades planificadas** según la carta Gantt. La investigación de campo, análisis de requerimientos y diseño completo del sistema se completaron dentro de los plazos establecidos. Sin embargo, el desarrollo de algunos módulos ha tomado más tiempo del estimado inicialmente.  **Factores que han facilitado el desarrollo:**   * **Documentación técnica abundante**: shadcn/ui, React y Supabase cuentan con documentación clara y comunidades activas que resolvieron dudas técnicas rápidamente. * **Tecnologías maduras y estables**: El stack tecnológico elegido (React, PostgreSQL) tiene ecosistemas consolidados con múltiples recursos de aprendizaje. * **Problema bien acotado**: Los requerimientos de las juntas de vecinos están claramente definidos por la Ley 19.418, lo que facilitó el análisis inicial. * **Metodología Kanban**: La flexibilidad de Kanban permitió ajustar prioridades sin afectar el flujo general del proyecto.   Factores que han dificultado el desarrollo:   * Complejidad de la autenticación multi-rol: Implementar el sistema de roles y permisos para diferentes tipos de usuarios (vecino, miembro, directorio) tomó más tiempo del planificado, requiriendo 3 días adicionales. * Integración de servicios externos: La configuración de servicios de notificaciones (email, SMS) presentó desafíos técnicos que extenderán el desarrollo del módulo de comunicaciones en 2 días. * Curva de aprendizaje de Supabase: Aunque es una plataforma potente, comprender a fondo Row Level Security (RLS) y las políticas de acceso requirió tiempo adicional no contemplado inicialmente. * Alcance ambicioso: El sistema incluye muchos módulos interconectados, lo que generó algunas dependencias técnicas que ralentizaron el desarrollo de funcionalidades posteriores.   **Ajustes realizados:**  Para mantener el proyecto en tiempo, priorizaré el desarrollo del MVP con las funcionalidades core (gestión de vecinos, solicitudes básicas, y comunicaciones esenciales), dejando funcionalidades secundarias (reportes avanzados, integraciones adicionales) para iteraciones posteriores. Actualmente estoy en la semana 14 de 18, con el MVP al 80% de completitud y confianza en finalizar las pruebas y documentación en las semanas restantes. |

|  |
| --- |
| 2. |
| ¿De qué manera has enfrentado y/o planeas enfrentar las dificultades que han afectado el desarrollo de tu Proyecto APT?  Estrategias para enfrentar las dificultades  1. Complejidad de autenticación multi-rol:   * Acción tomada: Simplifiqué la arquitectura de permisos usando las capacidades nativas de Supabase RLS en lugar de implementar un sistema custom complejo. * Resultado: Reducir 2 días de desarrollo y obtener mayor seguridad usando funcionalidades probadas de la plataforma.   2. Integración de servicios externos de notificaciones:   * Acción tomada: Para el MVP, implementé únicamente notificaciones por email usando servicio gratuito con buena integración. Las notificaciones SMS y WhatsApp quedaran documentadas como funcionalidad futura. * Resultado: Cumplír con el requerimiento core de comunicaciones sin bloquear el avance del proyecto.   3. Curva de aprendizaje de Supabase:   * Acción tomada: Dediqué 2 días completos a estudiar casos de uso específicos de RLS similares a mi proyecto, y consulté en la comunidad de Supabase para resolver dudas puntuales. * Planificación futura: Implementaré las políticas RLS de manera incremental, testeando cada módulo antes de avanzar al siguiente.   4. Alcance ambicioso del sistema:   * Acción tomada: Reorganicé el tablero Kanban priorizando con el criterio MoSCoW (Must have, Should have, Could have, Won't have). Enfoqué el desarrollo en los "Must have": autenticación, gestión de vecinos, solicitudes básicas, y comunicaciones. * Planificación futura: Las funcionalidades "Should have" (reportes avanzados, gestión completa de proyectos) se desarrollarán si el tiempo lo permite, o quedarán documentadas como roadmap post-entrega. Won't have (No tendrá - por ahora) integración con sistema contable   5. Dependencias técnicas entre módulos:   * Acción tomada: Creé un módulo base de componentes y servicios compartidos (layout, forms, API client) en las primeras semanas de desarrollo, que sirve como fundación para todos los módulos posteriores. * Resultado: Aunque tomó tiempo inicial, ahora el desarrollo de nuevas funcionalidades es más rápido y consistente.   6. Gestión del tiempo restante (4 semanas):   * Planificación:   + Semana 15: Completar módulos core faltantes (actividades comunitarias)   + Semana 16: Pruebas de integración y corrección de bugs críticos   + Semana 17: Pruebas de usabilidad y refinamiento de UI   + Semana 18: Documentación final y preparación de presentación   Lecciones aprendidas aplicables:   * Validar integraciones complejas con prototipos pequeños antes de integrarlas al sistema completo * Mantener comunicación constante con el profesor guía para validar ajustes de alcance * Documentar decisiones técnicas en el momento para facilitar la elaboración del informe final |

|  |
| --- |
| 3. Hasta el momento: |
| ¿Cómo evalúas tu trabajo? ¿Qué destacas y qué podrías hacer para mejorar tu trabajo?  Evaluación de mi trabajo  Aspectos destacables:   * Planificación sólida: La fase de análisis y diseño fue exhaustiva, lo que facilitó el desarrollo posterior y redujo cambios de requerimientos. * Decisiones técnicas acertadas: El stack elegido (React, Supabase, shadcn/ui) demostró ser adecuado para el proyecto, con buena documentación y comunidad activa. * Adaptabilidad: Supe ajustar el alcance usando priorización MoSCoW cuando identifiqué riesgos de tiempo, manteniendo el foco en entregar valor real. * Consistencia: Mantuve disciplina en el uso de Kanban y documentación continua del proceso.   Aspectos a mejorar:   * Estimación de tiempos: Subestimé la complejidad de algunos módulos (autenticación multi-rol, RLS). En futuros proyectos, agregaré un 20-30% extra a tareas con tecnologías nuevas. * Validación temprana: Debí crear prototipos más simples de integraciones complejas antes de integrarlas al sistema completo. * Gestión de dependencias: Identificar mejor las dependencias entre módulos desde el inicio habría optimizado el orden de desarrollo. * Pruenas desde el inicio: Implementar pruebas unitarias desde el principio en lugar de dejarlo para las últimas semanas. |

|  |
| --- |
| 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 y consultas  Inquietudes:   * Priorización entre completar todas las pruebas unitarias vs. agregar funcionalidades "Should have" en las 4 semanas restantes. * Nivel de detalle esperado en la documentación técnica final (¿manual de usuario completo o solo guía básica?).   Preguntas al docente:   * ¿Es aceptable entregar funcionalidades documentadas como roadmap futuro en lugar de implementadas, si cumplo sólidamente con el MVP? * ¿Qué pesa más en la evaluación: amplitud de funcionalidades o profundidad/calidad de las implementadas?   Preguntas a pares:   * ¿Cómo están manejando el testing en proyectos con tiempos ajustados? * ¿Alguien más usó Supabase? ¿Consejos para optimizar las políticas RLS? |

|  |
| --- |
| 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?  **Redistribución de actividades**  Al ser un proyecto individual, no aplica redistribución entre miembros. Sin embargo, he realizado una **repriorización de tareas** en mi tablero Kanban:   * **Reasigné tiempo** de funcionalidades secundarias (reportes avanzados, integraciones múltiples) hacia reforzar testing y documentación. * **Nuevas actividades identificadas**:   + Creación de datos de prueba (seed data) para demostración   + Video tutorial básico del sistema para usuarios finales   Estas actividades no estaban en la planificación inicial pero son necesarias para una entrega completa y profesional. |

|  |
| --- |
| 6. APT grupal |
| ¿Cómo evalúan el trabajo en grupo? ¿Qué aspectos positivos destacan? ¿Qué aspectos podrían mejorar?  **Aspectos positivos:**   * Autonomía total en decisiones técnicas, lo que agilizó la implementación * Consistencia en estilo de código y arquitectura al ser un solo desarrollador * Flexibilidad para ajustar horarios según complejidad de tareas   **Aspectos a mejorar:**   * **Falta de perspectivas diversas**: No contar con feedback inmediato de compañeros para validar decisiones de diseño o soluciones técnicas * **Carga de trabajo**: Asumir todos los roles (análisis, diseño, desarrollo, testing) puede generar puntos ciegos * **Solución aplicada**: Compensé consultando activamente en comunidades online (Supabase, foros de React) y validando decisiones con el profesor guía   En un proyecto futuro, valoraría tener al menos un compañero para code reviews y validación de UX/UI. |