|  |  |
| --- | --- |
|  | 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?  No hemos podido cumplir todas las actividades en los tiempos definidos. Según nuestra carta Gantt, teníamos planificado completar varias actividades documentales y técnicas durante las primeras semanas, pero hemos experimentado retrasos significativos.  Factores que han dificultado:  - Subestimación del tiempo para EDT y MDI: Requirió horas extras al final del sprint debido a nuestra falta de experiencia en descomposición de estructura de trabajo  - Indefinición inicial del alcance del MVP: Generó retrabajos en el documento de alcance y dificultad para priorizar funcionalidades  - Validación tardía de Google Vision OCR: Descubrimos restricciones técnicas al final del sprint que debimos validar tempranamente  - Retraso en endpoint de carga de recibos: Problemas con CORS y límites de archivo requirieron 2 días adicionales de debugging  - No se inició el desarrollo del frontend: El backend tomó más tiempo del estimado, dejando a Mario bloqueado  Factores que han facilitado:  - Arquitectura hexagonal bien aplicada: Permitió un desarrollo ordenado y facilita testing  - Mejora en comunicación del equipo: Daily standups de 15 minutos y tablero Kanban actualizado diariamente  - Stack tecnológico maduro: Go + Fiber, Next.js, PostgreSQL han demostrado ser herramientas robustas  - Configuración técnica sólida: PostgreSQL optimizado y AWS S3 integrado correctamente |

|  |
| --- |
| 2. |
| ¿De qué manera has enfrentado y/o planeas enfrentar las dificultades que han afectado el desarrollo de tu Proyecto APT?  Dificultades ya enfrentadas  - Gestión de credenciales AWS insegura: Resolvimos el problema reonfigurando completamente las variables de entorno y eliminando credenciales hardcodeadas  - Problemas de CORS en carga de recibos: Dedicamos 2 días de debugging para configurar correctamente el servidor  - Falta de claridad en documentación: Mario tuvo que leer el código fuente para entender payloads, lo que nos hizo  conscientes de la necesidad de documentación  Estrategias planificadas para próximos sprints:  1. Implementar TDD (Test-Driven Development): Escribir tests antes del código de producción con cobertura mínima del 80%  2. Establecer "Definition of Done": Código revisado por peer review, pruebas pasando, documentación actualizada, logs estructurados  3. Setup técnico al inicio del sprint: Dedicar las primeras 2 horas a configuración de entorno y resolver bloqueos  técnicos  4. Pair programming entre backend y frontend: Sesiones de 2 horas al inicio de cada módulo para alinear contratos de API  5. Automatizar documentación Swagger: Usar anotaciones en código Go para generar documentación automáticamente  6. Crear archivo .env.example: Documentar todas las variables de entorno con valores de ejemplo |

|  |
| --- |
| 3. Hasta el momento: |
| ¿Cómo evalúas tu trabajo? ¿Qué destacas y qué podrías hacer para mejorar tu trabajo?  valúo mi trabajo como satisfactorio pero con oportunidades de mejora importantes. Hemos logrado avances técnicos significativos, pero la falta de planificación detallada y la subestimación de tiempos ha generado estrés y retrabajos.  Aspectos que destaco:  - Aplicación correcta de arquitectura hexagonal que facilitará el mantenimiento futuro  - CRUD de gastos funcional con validaciones de negocio apropiadas  - Integración exitosa con AWS S3 para almacenamiento de recibos  - Mejora notable en comunicación del equipo (daily standups efectivos)  - Migraciones de base de datos implementadas con seeds iniciales  Aspectos a mejorar:  - Estimación de tiempos: Necesito ser más realista con las estimaciones, considerando mi nivel de experiencia y posibles imprevistos  - Pruebas unitarias: He priorizado features sobre calidad, acumulando deuda técnica que debo abordar  - Documentación continua: Debo documentar mientras desarrollo, no al final del sprint  - Validación temprana: Debo validar limitaciones técnicas (como cuotas de APIs externas) antes de comprometer features  - Gestión de seguridad: Debo ser más cuidadoso con credenciales y configuraciones desde el inicio |

|  |
| --- |
| 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:  1. Balance entre calidad y velocidad: ¿Cómo priorizamos entre completar features planificadas vs. abordar la deuda técnica acumulada (pruebas unitarias)?  2. Integración de OCR: Descubrimos limitaciones de cuota en Google Vision OCR. ¿Debemos replantear esta  funcionalidad o buscar alternativas (Tesseract, AWS Textract)?  3. Sincronización frontend-backend: Mario ha estado bloqueado esperando endpoints. ¿Cómo podemos trabajar en paralelo de manera más efectiva?  4. Alcance del MVP: ¿Estamos siendo demasiado ambiciosos con las funcionalidades? ¿Deberíamos reducir el alcance para asegurar un MVP funcional y bien probado?  Preguntas para el docente:  - ¿Es aceptable reducir el alcance del MVP si esto garantiza mayor calidad y cumplimiento de plazos?  - ¿Qué porcentaje de tiempo del sprint debería dedicarse a pruebas y documentación vs. desarrollo de features?  - ¿Cómo se evaluará el balance entre funcionalidades implementadas y calidad del código (cobertura de tests,  documentación)?  - ¿Existe alguna herramienta o práctica recomendada para mejorar nuestras estimaciones de tiempo? |

|  |
| --- |
| 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?  Sí, consideramos necesario redistribuir algunas actividades para lograr mayor balance y eficiencia:  Ajustes en responsabilidades:  1. Documentación Swagger:  - Antes: Daniel (backend)  - Propuesta: Ambos - Daniel implementa anotaciones en código Go, Mario valida desde perspectiva de consumidor frontend  2. Pruebas de Integración:  - Antes: Ambos  - Propuesta: Mario lidera - Como desarrollador frontend, puede definir mejor los contratos de API necesarios  3. Definición de contratos de API (DTOs):  - Nueva actividad - Ambos al inicio de cada sprint - Sesión de pair programming para alinear expectativas  4. Setup de entorno y configuración inicial:  - Nueva actividad - Daniel - Crear .env.example y documentación de setup para facilitar onboarding  5. Validación de servicios externos:  - Nueva actividad - Daniel - Validar limitaciones técnicas (cuotas, restricciones) antes de sprint planning  Actividades que se mantienen:  - Daniel: Backend, arquitectura, base de datos, seguridad, integraciones externas (OCR, AWS S3)  - Mario: Frontend, UX/UI, dashboard, consumo de APIs  Beneficios esperados:  - Reducir bloqueos de Mario esperando endpoints  - Mejor alineación en contratos de API  - Mayor visibilidad compartida sobre el progreso técnico |

|  |
| --- |
| 6. APT grupal |
| ¿Cómo evalúan el trabajo en grupo? ¿Qué aspectos positivos destacan? ¿Qué aspectos podrían mejorar?  En general, evaluamos el trabajo en grupo como positivo y en constante mejora. La comunicación ha mejorado significativamente y hemos logrado establecer rutinas de trabajo efectivas.  Aspectos positivos que destacamos:  1. Comunicación efectiva:  - Daily standups de 15 minutos funcionando correctamente  - Bloqueos resueltos con mayor rapidez  - Tablero Kanban en GitHub Projects actualizado diariamente  2. Coordinación en planificación:  - Reuniones de sprint planning con participación activa de ambos  - Coordinación fluida durante la fase de planificación  - Reuniones de sincronización semanales establecidas  3. Complementariedad de habilidades:  - División clara de responsabilidades (backend/frontend)  - Cada uno aporta desde su expertise  4. Compromiso compartido:  - Ambos participamos en gestión del proyecto y validación de entregables  - Ambos asumimos responsabilidad sobre documentación técnica  Aspectos a mejorar:  1. Trabajo en paralelo:  - Mario ha estado bloqueado esperando endpoints del backend  - Necesitamos definir contratos de API tempranamente para trabajar en paralelo  2. Peer review:  - No hemos implementado revisiones de código cruzadas  - Falta validación mutua antes de mergear cambios  3. Sesiones de pair programming:  - Solo trabajamos juntos en planificación, no en desarrollo  - Necesitamos sesiones de 2 horas al inicio de cada módulo complejo  4. Visibilidad compartida:  - La documentación Swagger incompleta dificultó la comunicación sobre APIs  - Necesitamos documentación continua, no al final  5. Estimaciones conjuntas:  - Subestimamos tiempos en EDT/MDI  - Debemos estimar juntos considerando dependencias entre frontend y backend  Acciones concretas para mejorar el trabajo en grupo:  - Implementar sesiones de pair programming semanales  - Establecer Definition of Done que incluya peer review  - Alinear contratos de API al inicio de cada sprint  - Usar mock servers o stubs para que frontend avance sin esperar backend real |