**Guía2. Desarrollo Proyecto APT**

**Asignatura Capstone**

|  |
| --- |
| **1. Resumen avance Proyecto APT** |
| A continuación, encontrarás distintos campos que deberás completar con la información solicitada. |

|  |  |
| --- | --- |
| Resumen de avance proyecto APT | Se definió y validó con el sponsor el problema y alcance del proyecto (ingreso a talleres y trazabilidad de la flota). Se aprobó el Acta, se levantaron y priorizaron requerimientos y se consolidó el **SRS v1.1**. Se diseñó la **arquitectura** objetivo (API Django + Postgres + Docker; frontend web responsivo) y el **MER**. Se elaboraron **mockups** de las pantallas clave y **casos de uso** primarios (ingreso a taller, creación de OT, estados/pausas, adjuntos, reportes). Se creó el repositorio y el **esqueleto del backend** (apps users y workorders, autenticación/roles y endpoints base en local). A nivel de gestión, se completaron **EDT + Diccionario**, **Gantt**, **RACI** y **Matriz de Adquisiciones**. Presentación de Fase 1 entregada. |
| Objetivos | Se mantienen el objetivo general y los específicos. Ajustes menores:   * Incorporación de actor **Guardia** y **Bodega**. * Énfasis en **trazabilidad en tiempo real** de estados/pausas. * Meta cuantitativa: **reducir ≥40%** el tiempo de registro y disminuir errores de duplicidad. * Inclusión de **reportes operacionales** (tiempos y SLA). |
| Metodología | **Modelo en Cascada** con hitos semanales: Análisis → Diseño → Implementación → Pruebas → Despliegue. Validaciones tempranas con prototipos; control mediante minutas y **RACI**; seguimiento con Gantt e issues en Git. **QA**: plan de pruebas inicial y criterios de aceptación por caso de uso. |
| Evidencias de avance | **Acta de Constitución**: formaliza el inicio y alcance.  **EDT + Diccionario / Gantt / RACI**: planificación, roles y control.  **SRS v1.1**: requerimientos funcionales y no funcionales validados.  **Arquitectura + MER**: diseño técnico y estructura de datos.  **Mockups + Casos de Uso**: validación temprana de UX y flujo.  **Repositorio y código backend inicial**: estructura del proyecto, modelos/endpoint base y autenticación.  **Plan de Pruebas Inicial**: estrategia y criterios de aceptación.  **Presentación Fase 1**: síntesis del avance ante la comisión. |

|  |
| --- |
| **2. Monitoreo del Plan de Trabajo** |
| Examina cuidadosamente tu plan de trabajo, enfocándote especialmente en la columna de estado de avance y ajustes. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Plan de Trabajo | | | | | | | |
| Competencia o unidades de competencias | Actividades | Recursos | Duración de la actividad | Responsable[[1]](#footnote-1) | Observaciones | Estado de avance | Ajustes |
| *Gestión de proyectos.*  *Diseño de Software.*  *Arquitectura/DevOps.*  *Desarrollo backend y frontend.*  *Calidad.* | *Toma de requerimientos.*  *Alcances.*  *Acta de constitucion.*  *Matrices de organización.*  *Gestion de actividades.* | *Plantilla de reunión con sponsor.*  *Plantilla srs*  *Diagramas*  *Plan de actividades y pruebas.* | *1 día*  *3 días.*  *1 semana.*  *3 días.* | *Diego*  *Diego*  *Diego/Luis*  *Diego/Luis* | *Dificultades técnicas relacionas al funcionamiento de la aplicación y facilitadores correspondientes a programas que nos permitan solucionar la problematica* | *Completado*  *Completado*  *Ajustado*  *ajustado* | *Se ajustaron las plantillas srs para que quedaran acorde al plan de trabajo y también se ajustó el plan de actividades (Gantt) para dar tiempo a más tareas importantes.* |

|  |
| --- |
| **3. Ajustes a partir del monitoreo** |
| Profundiza en las observaciones de tu plan de trabajo. Analiza las actividades planificadas y señala qué aspectos facilitaron u obstaculizaron la ejecución del plan. Plantea cómo abordaste y/o abordarás los obstáculos. Por último, señala los ajustes que realizaste al plan de trabajo a partir de este análisis. |

|  |
| --- |
| Factores que han facilitado y/o dificultado el desarrollo de mi plan de trabajo:  validaciones rápidas de alcance.  Plantillas/guías del curso (EDT, RACI, SRS) que ordenaron el trabajo.  Base técnica conocida (Django + Postgres + Docker) y repositorio configurado desde el inicio.  Definición temprana de casos de uso y mockups que alinearon expectativas.  Dificultades:  **Temas técnicos** (errores de imports/migraciones) → estandaricé entorno con Docker Compose, checklist de makemigrations/migrate y CI local.  **Datos reales/ejemplos** para reportes → definí dataset de prueba y plantilla de carga para Fase 2. |

|  |
| --- |
| Actividades ajustadas o eliminadas:  **Flujo “pausas/estados”**: ajustado tras feedback; se añadieron pre/postcondiciones y métricas (tiempos por estado).  **Arquitectura**: endpoints versionados **/api/v1**; **Redis** queda planificado para caché/colas en Fase 2.  **Adjuntos/evidencias**: **S3 y generación de PDF (Celery)** se movieron a “opcional”/Fase 2 para no bloquear Fase 1.  **Gantt**: corrimiento menor de 3–4 días por validaciones; no afecta el hito de cierre de Fase 1.  **Matriz de Adquisiciones**: simplificada a **LPI/LPN** según lineamientos de la cátedra. |

|  |
| --- |
| Actividades que no has iniciado o están retrasadas:  **Frontend web** (UI funcional sobre los endpoints): retrasado por priorizar SRS + backend base. Estrategia: dividir en vistas mínimas (login, lista OT, detalle) y usar componentes estándar.  **Pruebas automatizadas** (unitarias/integración): iniciadas parcialmente. Estrategia: añadir cobertura mínima por UC crítico y regla de PR con tests. |

1. En caso de que el Proyecto APT sea grupal, en esta columna deben indicar el nombre de los responsables de cada tarea o actividad. Esto posteriormente permitirá diferenciar la evaluación por cada integrante. [↑](#footnote-ref-1)