**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 | *Durante esta segunda fase del proyecto se avanzó en el desarrollo técnico y validación funcional del sistema Transervis, centrado en digitalizar la gestión de repuestos, optimizar la carga de datos y simplificar el proceso de publicación.*  *Ya tenemos implementado el backend Django con conexión estable a Oracle Autonomous Database (Always Free) y una VM en Oracle Cloud que aloja toda la parte lógica del sistema.*  *En esta etapa se concretaron los módulos principales:*   * *CRUD de repuestos con validaciones y conexión real a la base de datos.* * *Carga masiva CSV/XLSX, que permite ingresar lotes grandes de piezas desde planillas.* * *Validación y previsualización de registros, para evitar datos repetidos o mal cargados.* * *Exportación de ficha + imágenes para publicación manual en Facebook.* * *Roles y autenticación (Administrador, Bodega y Ventas).* * *Se añadieron reportes básicos (rotación e inventario general).*   *También se ajustó la arquitectura del proyecto, adoptando una metodología cascada en lugar de Scrum, ya que somos dos personas trabajando y el flujo secuencial se adapta mejor a la carga académica. Esto nos permitió tener entregables más claros por fase y mantener la trazabilidad sin perder orden.* |
| Objetivos | * Contar con un sistema funcional y estable en Oracle Cloud. * Mejorar la eficiencia en el ingreso de repuestos (voz o CSV). * Habilitar la exportación rápida de publicaciones para redes sociales. * Simplificar la administración interna de inventario. * Reducir errores humanos al cargar datos. |
| Metodología | El proyecto ahora sigue un modelo en cascada, con fases definidas:  Requerimientos -> Diseño -> Construcción -> Pruebas -> Despliegue -> Cierre.  Cada fase se completa antes de pasar a la siguiente, con revisiones entre medio.  Los roles dentro del equipo están distribuidos de forma simple:   * Uno se encarga del desarrollo, pruebas y despliegue. * Otro realiza la documentación, validación y pruebas UAT con usuarios reales. |
| Evidencias de avance | * Diagramas completos (arquitectura, componentes, flujo de usuario, ER, etc.) creados en Lucidchart. * Script DDL en Oracle SQL Developer (base de datos funcional y vinculada a Django). * Sistema web en pruebas internas (CRUD operativo y carga CSV funcionando). * Presentación actualizada según los cambios en el proyecto con todos los ajustes de alcance. * Estas evidencias demuestran el avance real del proyecto y muestran que ya pasamos de una etapa conceptual a una de desarrollo activo y validación. |

|  |
| --- |
| **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 |
| *Nombra las competencias o unidades de competencias que se relacionan con las diferentes actividades requeridas para el desarrollo del proyecto APT.* | *Nombra las actividades que se necesitan para desarrollar el proyecto APT.* | *Nombra los recursos necesarios para llevar a cabo las actividades definidas.* | *Señala la duración de cada actividad.* | *Escribe el nombre del integrante del equipo responsable de cada actividad.* | *Señala las dificultades o facilitadores que se podrían presentar durante la ejecución de cada una de las actividades propuestas.* | *Describe el estado de avance de cada actividad.*  *Tipos de estado:*  *En curso/ Con retraso/ No iniciado/ Completado/ Ajustada* | *Señala los ajustes o reformulaciones que has realizado.* |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Plan de Trabajo Proyecto APT | | | | | | | | |
| Competencia o unidades de competencias | Nombre de Actividades/Tareas | Recursos | Duración de la actividad | Descripción Actividades/Tareas | Responsable | Observaciones | Estado de avance | Ajustes |
| Gestionar proyectos informáticos, ofreciendo alternativas para la toma de decisiones de acuerdo a los requerimientos de la organización. | Kick-off y Acta de Inicio/Alcance | Alineación con Transervis y profesor. Definir problema, objetivos, alcance, stakeholders, riesgos y criterios de éxito. | WhatsApp; Google Docs; Plantilla de Acta; Calendario. | 1 semana | Equipo de trabajo | Facilitadores: acceso directo a empresa y profesor. Dificultades: conflictos de agenda en actualidad y futuro. | Completado | Cambiamos a cascada. Corregí el nombre a Transervis. dejamos el cronograma S1–S18. |
| Gestionar proyectos informáticos, ofreciendo alternativas para la toma de decisiones de acuerdo a los requerimientos de la organización. | Levantamiento de requerimientos y priorización | Entrevistas y observación en bodega y área de ventas; historias de usuario y criterios de aceptación; priorización de requerimientos. | WhatsApp; Google Sheets; Miro/Lucidchart o draw.io. | 2 semanas | Equipo de trabajo | Facilitadores: casos reales y disposición al cambio. Dificultades: baja disponibilidad de usuarios. | Completado | Sacamos el TTS y reconocimiento de imágenes. Dejamos STT como opcional. La publicación queda con export manual. |
| Implementar soluciones sistémicas integrales para automatizar u optimizar procesos de negocio de acuerdo a las necesidades de la organización. | Setup de infraestructura Always Free | Provisionar Oracle Autonomous Database (Always Free) y VM Always Free; repositorio y pipeline simple de despliegue. | Oracle Cloud (ADB + VM); GitHub; VS Code; SSH. | 1 semana | Equipo de trabajo | Facilitadores: costo cero (Always Free). Dificultades: cuotas/límites de servicio y performance. | Completado | El backend vive en la VM. La base está aparte en ADB. STT apagado por defecto. Conexión Django <->ADB probada. |
| Construir el modelo arquitectónico de una solución sistémica que soporte los procesos de negocio de acuerdo a los requerimientos de la organización y estándares de la industria. | Arquitectura 3 capas y Modelo de Datos (ERD) | Definir arquitectura de 3 capas (presentación Django templates, lógica de negocio/servicios, datos Oracle ADB) y ERD: pieza, vehículo, ubicación, categoría, estado, publicación, usuario/rol. | draw.io/Lucidchart; Oracle SQL Developer y Data Modeler; Documentación. | 2 semanas | Equipo de trabajo | Facilitadores: claridad de dominio e iteración temprana. Dificultades: sobre diseño o normalización excesiva. | Completado | Actualizamos los diagramas. La VM aloja el backend y ADB va por separado. Dejamos los roles Admin/Bodega/Ventas. Ajustamos el ERD al alcance real. |
| Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos. | Prototipo de Interfaz y Flujo | Wireframes de ingreso, inventario, búsqueda, publicaciones y dashboard; validación rápida con usuarios. | Figma / Prototipo HTML (Tailwind); WhatsApp (feedback). | 2 semanas | Equipo de trabajo | Facilitadores: feedback temprano y pruebas en dispositivos reales. Dificultades: al ser prototipo, no refleja datos reales. | Completado | Sumamos pantallas de Validar/Previsualizar y Exportar Ficha. Navegación por rol (Admin, Bodega, Ventas). |
| Realizar pruebas de certificación tanto de los productos como de los procesos utilizando buenas prácticas definidas por la industria. | Plan de Pruebas y Métricas Base | Casos de prueba, planilla de bugs y métricas (burn down, throughput, lead time). | Google Sheets; Postman; Capturas. | 1 semana | Equipo de trabajo | Facilitadores: cultura de calidad desde el inicio. Dificultades: falta de tiempo para ejecutar pruebas. | En curso | Priorizamos casos de CSV y CRUD. UAT en sesiones cortas. Sacamos la API de Facebook del plan de pruebas. |
| Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos. | Sprint 1: Inventario base (CRUD) | Modelos, vistas y templates en Django para piezas/categorías/ubicaciones/estados; formularios; endpoints; validaciones. | Django; Oracle ADB; VM; VS Code; Postman. | 1 semana | Equipo de trabajo | Facilitadores: primer módulo usable para validar. Dificultades: cambios de esquema por feedback. | Completado | Validaciones mínimas listas. Auditoría básica activa. Aplicamos los primeros índices. |
| Programar consultas o rutinas para manipular información de una base de datos de acuerdo a los requerimientos de la organización. | Sprint 2: Ingesta masiva (CSV/Excel) | Carga por lotes con validaciones y normalización; vista de previsualización y registro de errores. | Django; librería CSV/XLSX); Oracle ADB. | 1 semana | Equipo de trabajo | Facilitadores: acelera carga histórica. Dificultades: datos sucios o duplicados. | Completado | Previsualización lista. Bitácora de errores por línea. Plantillas CSV validadas. |
| Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos. | Sprint 3: Ingreso por voz (MVP) | Integración de Web Speech API en front; mapeo a campos; confirmación/edición antes de guardar en Django. | Web Speech API (navegador); Django; Oracle ADB. | 1 semana | Equipo de trabajo | Facilitadores: innovación sin costo y alta usabilidad. Dificultades: ruido ambiental y dependencia del dispositivo. | En curso | Lo dejamos como opcional. Fallback manual activo por el ruido. Sin servidor STT dedicado. |
| Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos. | Sprint 4: Búsqueda y filtros | Búsqueda por texto y filtros por categoría/estado/ubicación/vehículo; consultas optimizadas; paginación. | Django; índices BD; templates con filtros. | 1 semana | Equipo de trabajo | Facilitadores: mejora de usabilidad y rapidez. Dificultades: lentitud si faltan índices o hay muchos datos. | En curso | Priorizamos filtros clave y paginación. Dejamos el plan de índices para los campos más consultados. |
| Implementar soluciones sistémicas integrales para automatizar u optimizar procesos de negocio de acuerdo a las necesidades de la organización. | Sprint 5: Notificaciones y recordatorios | Reglas (stock bajo, sin publicar, tickets); jobs programados; alertas internas. | cron/DBMS\_SCHEDULER; Django management commands; Oracle ADB. | 1 semana | Equipo de trabajo | Facilitadores: visibilidad operativa continua. Dificultades: fatiga por exceso de alertas. | Pendiente | Dejamos en alertas internas simples para no meter ruido. |
| Implementar soluciones sistémicas integrales para automatizar u optimizar procesos de negocio de acuerdo a las necesidades de la organización. | Sprint 6: Publicación semiautomática | Generar fichas y export CSV/JSON + imágenes para carga manual a ML/FB. | Plantillas; exportadores CSV/JSON en Django; almacenamiento de imágenes. | 1 semana | Equipo de trabajo | Facilitadores: evita bloqueo por APIs externas. Dificultades: paso manual adicional en la operación. | Pendiente | Sacamos la integración con la API. Mantenemos export manual con imágenes. |
| Desarrollar la transformación de grandes volúmenes de datos para la obtención de información y conocimiento de la organización a fin de apoyar la toma de decisiones y la mejora de los procesos de negocio, de acuerdo a las necesidades de la organización. | Sprint 7: Dashboard ventas y rotación | KPIs: rotación de stock, piezas lentas, ventas por categoría; gráficos en frontend. | Consultas SQL (Oracle); Django views; librería de gráficos (Chart.js/Plotly). | 1 semana | Equipo de trabajo | Facilitadores: decisiones informadas. Dificultades: calidad/definición inconsistente de datos. | Pendiente | Priorizamos KPIs de rotación y piezas lentas. Las métricas avanzadas van después. |
| Resolver las vulnerabilidades sistémicas para asegurar que el software construido cumple las normas de seguridad exigidas por la industria. | Sprint 8: Roles y accesos | Autenticación Django auth, autorización por grupos/roles (admin/bodega/vendedor) y permisos por vista/endpoint. | Django auth; grupos y permisos; hash (PBKDF2/bcrypt/argon2). | 1 semana | Equipo de trabajo | Facilitadores: seguridad mínima indispensable. Dificultades: modelo simple limita escalabilidad inmediata. | Pendiente | Roles definidos (Admin/Bodega/Ventas) mapeados a vistas. Sumamos auditoría básica. |
| Realizar pruebas de certificación tanto de los productos como de los procesos utilizando buenas prácticas definidas por la industria. | Sprint 9: Pruebas integrales y rendimiento | Casos E2E (Postman), pruebas de carga livianas y tuning (índices/consultas/conn pool). | Postman/Newman; JMeter (opcional); EXPLAIN PLAN Oracle. | 1 semana | Equipo de trabajo | Facilitadores: estabilidad y performance. Dificultades: falta de datos realistas para pruebas. | Pendiente | Las cargas son limitadas por Always Free. |
| Realizar pruebas de certificación tanto de los productos como de los procesos utilizando buenas prácticas definidas por la industria. | Sprint 10: UAT con usuarios | Sesiones guiadas, registro de hallazgos y priorización; acta de resultados. | Checklist UAT; capturas/video; planilla hallazgos. | 1 semana | Equipo de trabajo | Facilitadores: validación basada en uso real. Dificultades: baja asistencia de usuarios. | Pendiente | hacemos sesiones cortas, ojalá en fin de semana. Plantillas UAT simples para acelerar feedback. |
| Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos. | Sprint 11: Ajustes post-UAT y hardening | Correcciones, mejoras de UX, revisión de seguridad y respaldos. | GitHub issues; scripts backup ADB; checklist. | 1 semana | Equipo de trabajo | Facilitadores: sistema listo para cerrar. Dificultades: crecimiento del alcance (scope creep). | Pendiente | Le damos prioridad a las correcciones críticas. Incluimos la prueba de restore aquí. |
| Gestionar proyectos informáticos, ofreciendo alternativas para la toma de decisiones de acuerdo a los requerimientos de la organización. | Documentación y Manual de Usuario | Guías breves: ingreso por voz, búsqueda, exportación a marketplaces. | Docs/Word; capturas. | 1 semana | Equipo de trabajo | Facilitadores: guías claras y reutilizables. Dificultades: baja adopción si el material es extenso. | Pendiente | Docs simples y al grano, orientados a tarea. Capturas directas del sistema. |
| Implementar soluciones sistémicas integrales para automatizar u optimizar procesos de negocio de acuerdo a las necesidades de la organización. | Despliegue final y capacitación | Release en VM; variables de entorno; respaldo; sesión de uso con equipo. | Oracle Cloud VM Always Free; SSH; scripts; WhatsApp. | 2 semanas | Equipo de trabajo | Facilitadores: plan de rollback y ambiente controlado. Dificultades: regresiones o caídas en productivo. | Pendiente | Checklist de despliegue y rollback listo. Restore verificado en prueba. |
| Gestionar proyectos informáticos, ofreciendo alternativas para la toma de decisiones de acuerdo a los requerimientos de la organización. | Demo final e Informe de Cierre | Video/demo completo; logros vs plan; mejoras futuras y continuidad. | Grabador de pantalla; plantilla de informe. | 1 semana | Equipo de trabajo | Facilitadores: foco en innovación (voz) y resultados. Dificultades: fallas en vivo. | Pendiente | Énfasis en beneficios prácticos: ingesta rápida y export. STT queda como opcional. |

|  |
| --- |
| **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. |

|  |
| --- |
| Facilitadores:   * Tener Oracle Cloud Always Free permitió desarrollar sin costos. * La buena comunicación dentro del equipo (por WhatsApp) hizo que las revisiones fueran rápidas. * El cambio a metodología en cascada ordenó mucho mejor el flujo y la entrega por fases.   Dificultades:   * Los límites del entorno gratuito de Oracle (CPU y RAM) hicieron lentas las pruebas. * La integración inicial con el módulo de voz (STT) no era tan precisa por el ruido del entorno, así que decidimos dejarlo opcional. |

|  |
| --- |
| Actividades ajustadas o eliminadas:  Eliminada: integración directa con API de Facebook (se reemplazó por exportación manual).  Eliminada: reconocimiento de imágenes (no era viable para esta fase).  Ajustada: el módulo de voz pasó a ser opcional, para activarse más adelante si hay recursos.  Estas decisiones se tomaron para asegurar que el sistema quedara operativo dentro del tiempo disponible. |

|  |
| --- |
| Actividades que no has iniciado o están retrasadas:  Pruebas UAT: aún no comienzan porque estamos cerrando la parte de interfaz.  Despliegue final: programado para el cierre de la fase, una vez que esté todo validado.  Para no afectar el proyecto, se planificó extender los test a un fin de semana completo y realizar sesiones rápidas con usuarios de bodega y ventas. |

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)