

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

| Resumen de avance proyecto APT | Durante este periodo avanzamos en el **MVP funcional** de Cambioteca, una plataforma para intercambio/venta de libros físicos con perfiles, catálogo, solicitudes de intercambio y coordinación de lugar/fecha. Los principales hitos:   * **Backend (Django):** módulos core y market operativos, conexión estable a **MySQL en AWS RDS**, manejo de usuarios (registro, login, recuperación de contraseña con token), perfiles, regiones/comunas y catálogo de libros con carga de imágenes. Emisión y validación de **JWT**, políticas de contraseña y email único. * **Flujos de intercambio:** creación y gestión de **solicitudes**, propuesta/confirmación de **lugar, fecha y hora**, estados del intercambio (pendiente/aceptado/rechazado/confirmado). Lógica para **calificación** entre usuarios (una sola vez por intercambio) definida y en integración. * **Frontend móvil/web (Ionic + Angular):** pantallas de **Autenticación**, **Registro**, **Catálogo**, **Detalle de libro**, **Solicitudes** y **Detalle de solicitud**. Estilos iniciales mejorados (formularios y tarjetas), navegación y mensajes de validación. * **Gestión del proyecto:** backlog por **épicas** (E01 Autenticación, E02 Publicación, E03 Catálogo, E05 Chat, etc.), **sprints** con seguimiento en Trello y **burndown**. Se planificó alcance por restricciones de tiempo, priorizando funcionalidades críticas del MVP.   **Objetivos específicos cumplidos hasta ahora (y cómo):**   * Autenticación segura y gestión de perfiles → endpoints y validaciones implementadas, pruebas con Postman y sesiones JWT. * Publicación y consulta de libros → CRUD de libros y vistas de catálogo/filtrado operativas. * Gestión de solicitudes de intercambio → creación, estados, coordinación y confirmación con persistencia en BD. * Infraestructura de datos → esquema en MySQL, migraciones y conexión a **AWS RDS** para trabajo colaborativo. |
| --- | --- |
| Objetivos | **Antes**: MVP con intercambio y pago integrado.  **Ahora** (ajuste): el pago integrado se traslada a una Fase 2; el MVP se centra en intercambio directo coordinado entre usuarios (entrega presencial o transferencia entre partes).  **Motivo**: optimizar tiempo y reducir complejidad técnica del sprint, manteniendo valor para el usuario y viabilidad de pruebas. |
| Metodología | ***Metodología:*** *Ágil/Scrum con ciclos de ~2 semanas.*   * ***Planificación:*** *lista de tareas priorizada por valor (backlog) y seguimiento en* ***Trello****.* * ***Coordinación:*** *reuniones diarias breves para ver avances y bloqueos.* * ***Cierre de ciclo:*** *demostración del avance, retroalimentación y ajuste del plan para el siguiente sprint.*  *Ajustes aplicados*  1. ***Menos tareas a la vez y tareas más pequeñas:***  *Para avanzar de forma sostenida, limitamos lo que hacemos en paralelo y dividimos las tareas grandes en pasos claros.* ***Ejemplo (flujo de intercambio):***  *crear solicitud → proponer lugar/fecha → confirmar → calificar.* 2. ***Planificación por prioridad:*** ***Chat*** *y* ***notificaciones*** *se movieron al siguiente sprint para concentrar el esfuerzo en terminar el intercambio completo del MVP.* 3. ***Despliegue por etapas:***  *Primero dejamos operativa la* ***base de datos en la nube (AWS RDS)****. Luego publicamos la* ***API*** *en un* ***servidor de aplicaciones (Railway)*** *—actualmente en progreso— para que la* ***web*** *y el* ***móvil*** *puedan probar en paralelo sobre el mismo backend.* |
| Evidencias de avance | **Link Trello:** [*https://trello.com/invite/b/68b9df08e5f8dca670bababb/ATTI42968d6be2b8c9aafa38cd3fe7844c4856BE3C0C/cambioteca-backlog*](https://trello.com/invite/b/68b9df08e5f8dca670bababb/ATTI42968d6be2b8c9aafa38cd3fe7844c4856BE3C0C/cambioteca-backlog)    ***Link Github:***[*https://github.com/Nicofelipe/cambioteca*](https://github.com/Nicofelipe/cambioteca)    ***Link Documento DAS:***[*DAS\_CAMBIOTECA.docx*](https://docs.google.com/document/d/1Zl-wKn3Eq57MPsMkBi502LKOpemTzINV/edit?usp=sharing&ouid=106324889313771275871&rtpof=true&sd=true)  *Documento DAS Incluye* ***arquitectura 4+1*** *(casos de uso, lógica, procesos, componentes y despliegue), además de contexto, propósito y alcance del sistema.* |
| **2. Monitoreo del Plan de Trabajo** |

| **Plan de Trabajo Proyecto APT** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Competencia / Unidad de Competencia** | **Nombre de Actividad / Tarea** | **Descripción de la Actividad / Tarea** | **Recursos** | **Duración** | **Responsable** | **Observaciones** |
| Gestionar proyectos informáticos | Planificación inicial y distribución de tareas | Definir alcance, cronograma y responsabilidades. | Kick-off, Trello | Semana 1 (22–30 ago) | Ambos | Importante dejar todo claro para avanzar ordenados. |
| Levantamiento y análisis de requerimientos (ágil) | Creación de historias de usuario y criterios de aceptación | Definir backlog inicial (épicas e historias), priorizar y refinar con criterios. | Trello, Plantillas HU | Semanas 2–3 (31 ago – 13 sep) | Ambos | Puede ajustarse con retroalimentación docente. |
| Modelar y administrar bases de datos | Diseño de la base de datos | Elaborar MER, MR y normalización; crear esquema inicial. | MySQL, Workbench, Draw.io | Semanas 4–5 (14 – 27 sep) | Nicolás | Ajustable según evolución del proyecto. |
| Desarrollar aplicaciones móviles | Desarrollo app móvil (UI/UX + frontend) | Pantallas principales: login, registro, catálogo, publicaciones, chat y mapa. | Ionic/Angular, Android Studio | Semanas 6–9 (28 sep – 25 oct) | Nicolás | Priorizar vistas mínimas para mostrar prototipo. |
| Desarrollar aplicaciones web | Desarrollo plataforma web | Implementar web con funcionalidades base (catálogo, perfil, login). | Node.js, Angular/React | Semanas 6–9 (28 sep – 25 oct) | Vicente | Trabajar en paralelo con la app móvil. |
| Programar sistemas de comunicación | Implementación de chat interno | Desarrollar chat básico para usuarios. | Socket.io, backend Django/Node | Semanas 10–11 (26 oct – 8 nov) | Ambos | Complejidad en tiempo real y seguridad. |
| DevOps/ Infra | Despliegue API y configuración | Variables .env, CORS; BD en RDS; API en Railway. | Railway, AWS RDS, GitHub | Semanas 13–14 (9 – 29 nov) | Vicente | Suite Postman en construcción; bugs menores resueltos. |
| Gestionar proyectos informáticos | Pruebas y validación | Ejecutar casos de prueba, pruebas unitarias y validación con usuarios. | Checklist QA, Postman, GitHub | Semanas 14 (23 – 29 nov) | Ambos | Tiempo crítico para detectar errores antes de la entrega. |
| Gestionar proyectos informáticos | Documentación y entrega final | Completar informe final, manual de usuario y presentación. | Word, PowerPoint, GitHub | Semanas 15 (30 nov – 5 dic) | Ambos | La entrega debe cumplir con formato académico. |

**Link de Excel con estado de avance y ajustes:** <https://docs.google.com/spreadsheets/d/1fOPYTwTnCnEPXMbJTbjWhtQz5hiEXjmE/edit?usp=sharing&rtpof=true&sd=true>

| **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:  **Facilitadores**   * **Orden del trabajo:** uso de Trello y un plan claro por etapas. * **Bases sólidas:** diseño de datos y funciones principales estables (registro, catálogo, solicitudes). * **Pruebas tempranas:** casos de prueba básicos que permitieron corregir a tiempo.   **Dificultades y cómo las abordamos**   * **Disponibilidad de tiempo:** coincidió con otras cargas académicas y laborales.  *Qué hicimos:* dividimos las tareas grandes en pasos más pequeños, definimos plazos cortos y realistas, y nos asignamos responsables claros. * **Publicar la API en un servidor externo:** la configuración inicial tomó más de lo esperado.  *Qué hicimos:* armamos una lista única de configuraciones necesarias (variables, permisos, direcciones permitidas) y probamos primero en un ambiente de prueba antes de pasarlo a producción. * **Alcance muy amplio:** había funciones que no eran claves para demostrar el valor del proyecto. *Qué hicimos:* reordenar prioridades y concentramos el esfuerzo en completar el **flujo de intercambio** de principio a fin. |
| --- |

| Actividades ajustadas o eliminadas:  **Pasarela de pago:** **eliminada del alcance actual**.  *Por qué:* requería tiempo y trámites adicionales que no eran necesarios para validar el intercambio de libros.  **Tecnología web:** **unificada** con la misma base que usamos en móvil.  *Por qué:* facilita el desarrollo y reduce la curva de aprendizaje. |
| --- |

| Actividades que no has iniciado o están retrasadas:   * **Pruebas automatizadas (unidad/integración) – No iniciada**  **Motivo:** se concentró el tiempo en construir las funciones principales.  **Cómo avanzaremos:** seleccionar casos críticos (auth, crear solicitud, confirmar intercambio) y automatizarlos primero. * **Google Maps – No iniciada Motivo:** priorizamos completar el flujo de intercambio. **Cómo avanzaremos:** implementar la versión mínima (elegir y guardar punto de encuentro), documentar la clave de API y probar con 2–3 casos reales. |
| --- |