

| 1. Resumen avance Proyecto APT |
| --- |
|  |

| Resumen de avance proyecto APT | Durante el desarrollo del proyecto RutaFit se ha avanzado en la configuración del entorno tecnológico, la construcción de la base de datos y el diseño de las principales interfaces de la aplicación.  Inicialmente, se configuraron los frameworks y dependencias necesarias para el trabajo con React Native y Expo en el front-end móvil, actualizando las versiones de Node, npm y librerías asociadas para asegurar compatibilidad. A nivel de servicios, se crearon las cuentas de Firebase, destinada exclusivamente a la autenticación y gestión de credenciales de los usuarios (registro, login y recuperación de contraseña), y de MongoDB Atlas, que se utiliza como base de datos para almacenar el resto de la información vinculada a usuarios, rutas, eventos, tipos de deporte y niveles de experiencia.  En paralelo, se desarrolló una API REST con Express, desplegada en Vercel, que cumple el rol de intermediaria entre la aplicación móvil y la base de datos. Gracias a esta arquitectura, se han implementado operaciones como el registro de usuarios en MongoDB (posterior a la validación en Firebase), la consulta de catálogos dinámicos y la creación de eventos, asegurando que la comunicación entre front-end y base de datos sea consistente y escalable.  En el ámbito del diseño, se elaboró un mockup en Figma que sirvió como guía visual para el desarrollo de las primeras interfaces. Con base en este diseño, se implementaron la pantalla de inicio de sesión y el registro de usuarios, ambos conectados a la lógica de Firebase para autenticar y proteger la información sensible. También se implementó la recuperación de contraseña mediante el sistema de enlaces seguros proporcionado por Firebase, lo que refuerza la seguridad de la aplicación.  Además, se desarrolló la interfaz de eventos, en la cual un modal despliega el formulario para registrar actividades deportivas. Para complementar, se integró el consumo de datos dinámicos desde la API con el método GET, lo que permite cargar en tiempo real las opciones de tipo de deporte y nivel de experiencia definidas en la base de datos. Esta estrategia evita la carga manual en el front-end y permite que, si se agregan nuevas categorías, estas se reflejen automáticamente en la aplicación.  Un hito clave ha sido la definición completa de la base de datos en MongoDB, estructurada en las siguientes colecciones:   * Usuarios: almacena la información básica (nombre, apellido, correo único, contraseña, género, fecha de nacimiento y nivel de experiencia), además de referencias a las rutas creadas y eventos en los que participa cada usuario. * rutas: guarda los datos de cada recorrido (nombre, descripción, nivel de dificultad, distancia), junto con las coordenadas registradas y las valoraciones de los usuarios. * eventos: contiene el detalle de cada actividad deportiva (nombre, fecha, lugar, estado, máximo de participantes, tipo de deporte vinculado a tipos\_deporte), además de los participantes inscritos y el organizador. Incluye campos de auditoría como createdAt y updatedAt. * tipos\_deporte: define las distintas disciplinas disponibles en la aplicación (ejemplo: ciclismo, running), lo que permite mantener un catálogo dinámico y escalable. * nivel\_experiencia: almacena los distintos niveles de los usuarios (principiante, intermedio, avanzado), utilizados en el registro y en la personalización del perfil.   Estas estructuras se diseñaron con tipos de datos y restricciones claras, como la unicidad del correo en la colección usuarios o la obligatoriedad de campos clave en eventos, lo que asegura consistencia y confiabilidad en la información.  De manera complementaria, se avanzó en la implementación del perfil de usuario. Para ello, se rescata desde el proceso de login el ID del usuario autenticado en Firebase, con el cual se consulta la base de datos en MongoDB para obtener la información completa del perfil. Estos datos se almacenan en caché local, lo que permite mantenerlos visibles en la aplicación aunque no haya conexión inmediata y conservarlos hasta que el usuario cierre sesión o se autentique otra persona, también se agregó una función de actualizar perfil.  Finalmente, en la vista de Home se integró el mapa inicial con la configuración de permisos de ubicación, lo que permite reconocer la posición actual del usuario y establece la base para las futuras funcionalidades de seguimiento de rutas y geolocalización en tiempo real.  Adicionalmente, se elaboraron diagramas de arquitectura y de clases, que documentan la interacción entre los distintos módulos de la aplicación y sirven como guía para la implementación técnica.  En cuanto a los objetivos del proyecto, su evaluación se realizará cuando la aplicación entre en funcionamiento; por ahora no hay datos que permitan afirmar cumplimientos. Los objetivos de retención (60% a 3 meses), creación y compartición de rutas (50% en tres meses), participación en eventos (40% en tres meses) y satisfacción global (70% al cierre del piloto) se medirán con usuarios reales una vez iniciada la operación. El único objetivo que ya muestra avances es el de usabilidad: se cuenta con pantallas principales, un plan sencillo de pruebas con encuesta y se han iniciado pruebas internas para recoger opiniones y ajustar la interfaz. |
| --- | --- |
| Objetivos | -Promover la retención de usuarios, de manera que al menos el 60% de quienes se registren siguen activos después de 3 meses, evaluado mediante su interacción con rutas o eventos.  -Lograr que al menos el 50% de los usuarios activos registren y compartan una ruta deportiva durante sus primeros 3 meses de uso.  -Fomentar la participación en comunidad, alcanzando que el 40% de los usuarios participe en al menos un evento deportivo grupal durante los primeros 3 meses.  -Validar la usabilidad de la interfaz mediante pruebas con al menos 20 usuarios beta, alcanzando un 70% de satisfacción en criterios de facilidad de uso, diseño y comprensión de funciones.  -Evaluar la satisfacción global de la aplicación, alcanzando un 70% de valoración positiva en criterios de diseño, funcionalidad y utilidad, mediante encuestas aplicadas a un grupo de al menos 30 usuarios al final de la fase piloto. |
| Metodología | Para el desarrollo del proyecto RutaFit se continúa implementando la metodología ágil Scrum, sin modificaciones respecto a la planificación inicial. Esta metodología ha demostrado ser la más adecuada, ya que permite trabajar de forma iterativa e incremental, entregando valor de manera constante y adaptando el desarrollo a las necesidades que surgen en el proceso. |
| Evidencias de avance | Documentos Scrum: registran la planificación y seguimiento de tareas, lo que refleja la organización del trabajo en equipo y la aplicación de una metodología ágil. Estos documentos permiten mostrar la trazabilidad del proyecto y la distribución de responsabilidades.  Base de datos en MongoDB Atlas: evidencia la definición de colecciones, campos e índices pensados para el correcto manejo de la información de la aplicación. Esta base demuestra que se ha avanzado en el diseño y la implementación del modelo de datos que sustentará las principales funcionalidades.  Diagrama de arquitectura: ilustra la estructura general del sistema, donde se observa la interacción entre la app móvil en React Native, el backend en Express desplegado en Vercel, Firebase Auth y la base de datos MongoDB Atlas, además de Google Map API. Este diagrama permite entender la lógica de integración entre los distintos componentes. Diagrama de clases: representa la organización interna del proyecto a nivel de entidades y sus relaciones, facilitando la comprensión de la lógica de negocio y garantizando la consistencia del modelo de datos. |
| 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-0) | 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. |
| Desarrollo de software móvil | Definición de arquitectura (Firebase Auth + MongoDB + API en Express + React Native con Expo) | Documentación, diagramas de arquitectura | 1 semana | Equipo completo | Facilitó tener claro desde el inicio el stack tecnológico | Completado |  |
| Diseño de experiencia de usuario | Definir colores, tipografías y diseño base de pantallas | Mockups, Figma, guías de estilo | 1 semana | Integrante de diseño / Equipo | Permitió avanzar rápido en interfaces sin dudas estéticas | Completado | No requiere ajustes |
| Desarrollo frontend móvil | Creación de pantallas de login, registro y home con validaciones básicas | React Native con Expo, Expo Router | 1 semana | Equipo frontend | Se avanzó con validaciones de correo y contraseña | En curso | Se ajustó tiempo por curva de aprendizaje con Expo Router |
| | Desarrollo backend | | --- |  |  | | --- | |  | | Configuración de API en Express, desplegada en Vercel, conexión con MongoDB | Node.js, Express, Vercel, MongoDB Atlas | 1 semanas | Equipo backend | Cluster estuvo pausado y hubo que reactivarlo | En curso | Ajuste en la configuración de índices y fechas de eventos |
| Modelado de base de datos | Creación de colecciones: usuarios, eventos, categorías, niveles de experiencia | MongoDB Atlas, Data Explorer | 2 semana | Equipo backend | Ya se definió esquema base de eventos y usuarios | Completado | No requiere ajustes |
| Gestión ágil de proyecto | Uso de Scrum con sprints de 2-3 semanas, generación de documentos y tableros | Scrum, GitHub Projects, documentación de Scrum | Durante todo el proyecto | Scrum Master / Equipo | Facilitó organización y control de avances | En curso | No requiere ajustes |
| Panel web de estadísticas | Implementación de panel para visualizar métricas de uso | React, Express, Vercel | 4 semanas | Equipo backend/frontend | Aún no se inicia, previsto para siguiente sprint | No iniciado | Pendiente iniciar en próximos ciclos |
| Pruebas de integración y seguridad | Validar flujos críticos: login, registro, creación de eventos | Jest, Postman, pruebas en Expo Go | 2 semanas | Equipo QA | Se están preparando casos de prueba | Con retraso | Se extendió por problemas de dependencias y ajustes de backend |

| 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 abordar 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 Factores que han facilitado y/o dificultado el desarrollo El desarrollo del proyecto se ha visto favorecido por la claridad de la idea inicial, la cual nos permitió orientar todas las decisiones hacia un objetivo común desde el inicio. A esto se suma el uso de tecnologías con amplia documentación y soporte, como React Native, Firebase y MongoDB, que facilitaron el aprendizaje progresivo y la resolución de problemas.  En particular, Firebase ha sido una herramienta fundamental, ya que sus funciones de autenticación simplificaron la gestión del registro, inicio de sesión y recuperación de contraseñas de los usuarios. Gracias a ello, hemos podido garantizar desde las primeras etapas del proyecto un sistema seguro y confiable para manejar datos sensibles, sin necesidad de desarrollar estos procesos desde cero.  Otro aspecto que ha resultado clave es haber definido un diseño base antes de iniciar el desarrollo. Esto incluyó la selección de la paleta de colores, las tipografías oficiales y la línea gráfica general de la aplicación. Contar con estas definiciones previas evitó detenernos en discusiones estéticas durante la programación y nos permitió mantener coherencia visual en todas las pantallas de la app.  Finalmente, el uso de GitHub como repositorio central y la organización del trabajo en ramas nos han facilitado coordinar esfuerzos, registrar avances de forma ordenada y trabajar en paralelo sin perder el control del código.  Si bien al inicio teníamos clara la idea conceptual del proyecto, enfrentamos dificultades para llevarla a la práctica. En este sentido, la orientación de los profesores de capstone ha sido un pilar fundamental, ya que nos brindaron una guía constante para organizarnos de manera más efectiva y para seleccionar las herramientas más adecuadas para el desarrollo.  Las principales dificultades del proyecto han estado marcadas por la falta de experiencia previa en las tecnologías utilizadas, lo que significó enfrentarnos a una curva de aprendizaje pronunciada. Desde un inicio decidimos trabajar con herramientas y frameworks nuevos para nosotros, por lo que gran parte del avance se ha dado en paralelo con la investigación y el aprendizaje de su uso. A esto se sumaron los problemas de compatibilidad entre dependencias y los ajustes constantes en la configuración de MongoDB Atlas, que demandan tiempo extra para resolver.  Otra dificultad importante ha sido la fase de pruebas, ya que el emulador móvil no siempre funciona de manera estable y esto nos ha dificultado comprobar cómo se comporta realmente la aplicación en un dispositivo físico.  Para superar estos obstáculos hemos recurrido a la consulta de documentación oficial, la realización de pruebas constantes y la revisión de la arquitectura del sistema cada vez que fue necesario ajustarla. Aunque el proceso ha sido desafiante, esta dinámica de resolver sobre la marcha nos ha permitido avanzar de manera sostenida y, al mismo tiempo, adquirir competencias técnicas que antes no teníamos. |
| --- |

| Actividades ajustadas o eliminadas:  Durante el desarrollo de RutaFit realizamos ajustes para mantener un alcance realista, seguro y acorde al tiempo disponible: se descartó reportar incidentes porque requería un flujo de seguimiento, triage y almacenamiento adicional que excedía la capacidad del MVP; en compartir rutas, pasamos de publicar todo automáticamente a que el usuario elija qué rutas compartir, priorizando control y privacidad; se eliminó compartir ubicación con personas de confianza por sus implicancias de seguridad y la complejidad de permisos persistentes; se eliminó el módulo de valoraciones y comentarios (tanto comentarios predefinidos como de texto libre), ya que implicaba una carga de moderación constante y riesgo de publicación de contenido ofensivo, además de desviar esfuerzo del núcleo funcional; el sistema de ranking, medallas y logros se dejó fuera por requerir un motor de gamificación y reglas de cálculo que no aportan al MVP; la búsqueda de rutas se simplificó a búsqueda por comuna (en lugar de nombre o cercanía) para reducir la complejidad de indexación, geocodificación y rendimiento; la visualización de usuarios activos en el mapa se descartó por riesgos de privacidad y potencial mal uso; y la función de “match” con usuarios cercanos se eliminó por alinearse más a una red social (con necesidades de moderación, consentimiento y reglas adicionales). Con estos ajustes, el proyecto queda enfocado en lo esencial: registrar/compartir rutas de forma controlada, gestionar eventos y mantener una experiencia segura y viable para el MVP. |
| --- |

| Actividades que no has iniciado o están retrasadas:  Hasta el momento no se registran actividades retrasadas respecto a la planificación inicial de los sprints. Todas las tareas definidas en el plan de trabajo se encuentran dentro de los plazos establecidos y avanzan según lo previsto. No obstante, hemos tenido ciertos retrasos en la realización de algunas ceremonias de Scrum, específicamente las daily meetings, el sprint review y la retrospective. Pese a ello, el progreso del proyecto no se ha visto afectado de manera significativa, ya que el equipo ha mantenido la comunicación por otros medios y el seguimiento de las tareas se ha gestionado de forma adecuada mediante GitHub y la planificación de los sprints. |
| --- |

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-0)