

Requisito		Estimación	Argumento	Caso de uso (GitHub)
RF_1	RF_1: La aplicación deberá ser instalable y ejecutable tanto en Android como en iOS.	5	React Native permite desarrollo multiplataforma, pero requiere configuración en ambos entornos. No hay experiencia previa por parte de los desarrolladores.	
MUST				
RF_2	RF_2: Si el usuario abre la aplicación por primera vez y no se ha iniciado sesión aún, deberá mostrar pantalla de bienvenida con opciones de registro o inicio de sesión	5	Es funcionalidad estándar; puede implementarse con React Navigation y AsyncStorage. La mayoría de los desarrolladores no han hecho un Login antes.	
MUST				
RF_3	RF_3: La aplicación deberá permitir al usuario registrarse con Google o Facebook.	5	Integrar el workflow OAuth de dichas plataformas y el debido manejo de tokens; requiere configuración de APIs externas.	
SHOULD				
RF_4	RF_4: La aplicación deberá permitir al usuario registrarse manualmente.	5	Se necesita conectar y sincronizar las base de datos cada vez que se registre un usuario.	<a href="https://github.com/Yareaj/ingesoft-i/blob/main/Documentacion/Casos%20de%20uso/CU_2_angomezp.pdf">https://github.com/Yareaj/ingesoft-i/blob/main/Documentacion/Casos%20de%20uso/CU_2_angomezp.pdf</a>
MUST				
RF_5	RF_5: La aplicación deberá permitir al usuario iniciar sesión con username y contraseña.	5	Se debiera hacer operaciones CRUD a la base de datos y generar un token de autenticación para el usuario	Jauri
MUST				
RF_6	RF_6: La aplicación deberá permitir al usuario iniciar sesión con Google o Facebook.	5	Integrar el workflow OAuth de dichas plataformas y el debido manejo de tokens; requiere configuración de APIs externas.	
SHOULD				
RF_7	RF_7: Cuando el usuario complete el proceso de inicio de sesión, la aplicación deberá mostrar la pantalla principal del usuario (dashboard) que consiste en un mapa con ubicación actual, si ya se han otorgado los permisos de GPS, y los botones para acceder a las pestañas de "Feed", "Profile", "Saved Routes", "Historial", "Start training".	2	Requiere navegación y obtención de datos del usuario autenticado.	Pablo
MUST				
RF_8	RF_8: Si el usuario intenta usar funciones basadas en ubicación o inicia sesión por primera vez en la aplicación, deberá solicitar permiso para GPS (Si no se ha otorgado anteriormente) y explicar por qué se solicita.	2	Puede hacerse con expo-location o APIs nativas. Manejo de permisos por plataforma a través de librería en React Native	<a href="https://github.com/Yareaj/ingesoft-i/blob/main/Documentacion/Casos%20de%20uso/CU_1_angomezp.pdf">https://github.com/Yareaj/ingesoft-i/blob/main/Documentacion/Casos%20de%20uso/CU_1_angomezp.pdf</a>
MUST				
RF_9	RF_9: Cuando el usuario otorgue permiso de GPS y los servicios de ubicación estén activos, la aplicación deberá mostrar la ubicación actual del usuario sobre un mapa en la pantalla principal.	3	Integración con react-native-maps. Requiere marcadores dinámicos.	
MUST				
RF_10	RF_10: Si el usuario deniega permisos críticos y el usuario intenta usar una funcionalidad que los necesite, la app deberá bloquear la funcionalidad requerida hasta que estos permisos se otorguen.	2	Configuración para pedir permisos de GPS, se puede hacer con la biblioteca Geolocation.	
MUST				

RF_11				
SHOULD	RF_11: La app deberá pedir permiso para enviar notificaciones push y permitir configurar su periodicidad.	5	Requiere entender como funciona el NotificationManager.	
RF_12				
MUST	RF_12: Cuando el usuario oprima 'inicio de actividad', la app deberá rastrear ruta (GPS) y tiempo.	5	Requiere cronómetro y estado en tiempo real. Integración con react-native-maps. Requiere marcadores dinámicos.	Jauri
RF_13	RF_13: Mientras la actividad esté en curso, la aplicación deberá registrar cada segundo (o la frecuencia necesaria según se vea en las pruebas de software) la distancia recorrida, el tiempo transcurrido, las calorías quemadas y calcular la velocidad promedio, grabando los puntos de ubicación de GPS para tal objetivo.	5	Cálculos y almacenamiento en tiempo real.	Diego
MUST				
RF_14	RF_14: Si la funcionalidad "Run against my ghost" está activa, la app deberá emitir reporte de audio cada 1 km o cada minuto indicando diferencia, según la configuración establecida por el usuario en la actividad	5	Control de flujo de rastreo. Sincronización de actividad con el ghost.	
SHOULD				
RF_15	RF_15: Si la funcionalidad "Run against my ghost" esta activa, la app deberá mostrar en el mapa la posición actual del deportista y la de su ghost con el fin de tener en tiempo real la comparativa.	5	Control y visualización de flujo de rastreo.	Pablo
MUST				
RF_16	RF_16: Si la funcionalidad "Run against my ghost" está activada y se alcanza la distancia del entrenamiento compartido, entonces la aplicación deberá detener automáticamente la grabación del entrenamiento actual.	2	Teniendo ya las funcionalidades de grabacion, pausa y finalizacion de entrenamiento este proceso sale como una combinacion de estas.	
MUST				
RF_17	RF_17: Mientras se graba un entrenamiento, la aplicación deberá tener la opción de "pausa". Cuando la actividad esta en pausa, el usuario tiene las dos opciones (botones) de "reanudar" y "finalizar" el entrenamiento, sin alterar en ningún caso la información recopilada hasta el momento.	3	Persistencia temporal y control de estado.	Diego
MUST				
RF_18	RF_18: La aplicación deberá permitir al usuario guardar un trayecto (ruta GPS) con un nombre y metadata (distancia, desnivel aproximado) para uso futuro.	5	CRUD local o en la nube vinculado al usuario.	<a href="https://github.com/Yareaj/ingesoft-i/blob/main/Documentacion/Casos%20de%20uso/CU_3_angomezp.pdf">https://github.com/Yareaj/ingesoft-i/blob/main/Documentacion/Casos%20de%20uso/CU_3_angomezp.pdf</a>
MUST				
RF_19	RF_19: Cuando ocurra un evento relevante (ej. alguien compite contra tu ghost, actividad finalizada, <b>permiso denegado</b> ), la aplicación deberá enviar la notificación push correspondiente según las preferencias de privacidad y notificación del usuario.	5	Integración con Firebase o Expo Notifications.	
COULD				
RF_20	RF_20: Si el usuario solicita comparar dos entrenamientos (por ejemplo, su entrenamiento vs su ghost), entonces la aplicación deberá generar un reporte comparativo que incluya cadencia promedio por km, velocidad por km y diferencias temporales, el cual podrá ser visualizado, compartido o descargado en formato imagen.	5	Procesamiento y visualización de datos	
COULD				

RF_21	RF_21: El sistema deberá tener la opción de finalizar el entrenamiento y al finalizarlo, la aplicación deberá mostrar un resumen técnico que incluya tiempo de actividad, distancia recorrida y velocidad promedio.	3	Vista simple con datos calculados.	Jauri
MUST				
RF_22	RF_22: La aplicación deberá almacenar el historial de entrenamientos del usuario y permitirle consultarlo como lista ordenada por fecha.	3	Consulta en base de datos local y presentarlo en la aplicación.	Pablo
MUST				
RF_23	RF_22: La aplicación deberá incluir una funcionalidad de configuración de privacidad donde el usuario pueda especificar: quién puede ver sus trayectos, quién puede usar sus ghosts y de quién recibir notificaciones o retroalimentación.	5	Requiere modelos de usuario y control de permisos. No hay experiencia de los desarrolladores en manejo de privacidad de elementos.	
SHOULD				
RF_24	RF_24: El usuario podrá consultar su perfil para el cual se desplegará una pestaña con el contenido de su foto de perfil, descripción de perfil, número de seguidos, número de seguidores, ver perfil completo	5	Requiere diseño de interfaz, manejo de estado y carga dinámica de datos del perfil.	Diego
MUST				
RF_25	RF_25: La app deberá mantener y dar visualización a una lista de "seguidos" y "seguidores" en el perfil de cada usuario	5	Implica gestionar relaciones entre usuarios y sincronización en tiempo real de listas.	
COULD				
RF_26	RF_26: Habrá una pestaña de "Feed" en la que se alojarán las publicaciones de los usuarios seguidos y las del propio usuario.	5	Demanda integración del feed con base de datos, renderizado dinámico y optimización.	
COULD				
RF_27	RF_27: En la pestaña "Feed" habrá una barra de búsqueda para encontrar a otros usuarios	3	Involucra desarrollo de un buscador con filtrado y consultas eficientes en tiempo real.	
COULD				
RF_28	RF_28: Un usuario podrá visualizar la información pública del perfil de otros usuarios. Además, podrá acceder a la información seleccionada como de acceso a solo seguidos (entrenamientos, estadísticas) de los usuarios a los que siga.	5	Relaciones entre usuarios y sincronización.	
COULD				
RF_29	RF_29: La app deberá tener un apartado de 'Rutas guardadas' para agregar o compartir rutas que quiera destacar de su historial de entrenamientos.	3	CRUD de rutas con almacenamiento.	
SHOULD				
RNF_1	RNF_1: El sistema deberá estar desarrollado en React Native con TypeScript.	5	Define el stack principal multiplataforma.	
MUST				
RNF_2	RNF_2: La aplicación deberá cumplir con tiempos de muestreo GPS tales que el rastreo permita reconstruir la ruta con precisión suficiente para calcular distancia y pace (p. ej. muestreo $\leq 5$ s en movimiento).	5	Determina calidad de experiencia; depende del hardware.	
MUST				

RNF_3				
MUST	RNF_3: La app deberá implementar autenticación segura con JWT y HTTPS.	5	Evita vulnerabilidades de datos personales.	
RNF_4				
SHOULD	RNF_4: La interfaz deberá responder a interacciones en menos de 1 s.	5	Impacta la UX; requiere optimización.	
RNF_5				
COULD	RNF_5: El código deberá ser modular y documentado.	5	Mejora la escalabilidad.	
RNF_6				
COULD	RNF_6: Las funcionalidades del código deben estar testeadas un 65% con pruebas unitarias.	5	Mejora la escalabilidad.	
RNF_6				
COULD	RNF_6: El sistema deberá funcionar offline y sincronizar al reconectarse.	5	Requiere sincronización de datos local-nube.	