## **Asigna a cada tarea quién del equipo (rol) la realiza.**

1. **Analista de sistema → Dani**
   * Identificar los requisitos funcionales y no funcionales de la aplicación en una reunión con el cliente.
   * Detallar los flujos de búsqueda de libros y sus requisitos.
2. **Diseñador de software → Lili**
   * Crear los bocetos y maquetas visuales de la interfaz de la aplicación para el diseño de la interfaz de usuario.
   * Diseñar la estructura general del sistema y la interacción entre los componentes.
3. **Analista programador → Darío**
   * Implementar la funcionalidad para que los usuarios se registren en la aplicación que se está desarrollando.
   * Analizar y optimizar las consultas a la base de datos mejorando el rendimiento.
   * Revisar el código, identificar errores y depurarlos para mejorar la calidad.
4. **Programador → Juanma**
   * En la creación del back-end para la gestión de libros, el desarrollo de sistemas de dicha gestión, incluyendo la base de datos.
5. **Arquitecto de software → Lili**
   * Diseñar la estructura general del sistema y la interacción entre los componentes. (Esta tarea también puede incluirse en el diseñador de software dependiendo del proyecto).
6. **Jefe de proyecto → Juanma**
   * Crear documentación técnica y de usuario para el proyecto.
   * Realizar pruebas de usabilidad con usuarios para identificar posibles problemas de diseño. (Esta tarea también puede estar bajo el analista de sistemas, dependiendo de la estructura del equipo) .

**Esquema**

Aplicación para un gimnasio en la que se puedan registrar diferentes datos. Cada usuario deberá registrarse con un nombre de usuario y contraseña.

**1. Analista de sistema (Dani)**

* **Identificar requisitos funcionales y no funcionales**:
  + **Funcionales**: Son las características que la aplicación debe tener para cumplir con su propósito. Ejemplo: registro de nuevos usuarios, acceso al historial de entrenamientos.
* **Requisitos funcionales:**
  + **No funcionales**: Requisitos de rendimiento, seguridad, y usabilidad. Ejemplo: la app debe ser rápida, proteger datos personales y ser intuitiva en dispositivos móviles.
* **Ejemplo**: En una reunión con el cliente, Dani pregunta sobre el tiempo máximo de respuesta aceptable para una consulta de reservas de clases y si se requiere una capa adicional de seguridad para los pagos.
* **Detallar flujos de búsqueda de libros y sus requisitos**:
  + Aquí se entiende que se busca por "libros" en lugar de entrenamientos, reservas de clases, etc., que podría ser un término erróneo o de un proyecto similar.
  + **Ejemplo**: Un flujo en la app podría ser "búsqueda de entrenadores o clases" por disponibilidad, popularidad o tipo de entrenamiento.

**2. Diseñador de software (Lili)**

* **Crear bocetos y maquetas visuales**:
  + Estos son los primeros modelos visuales de la app, donde se estructura la disposición de cada pantalla.
  + **Ejemplo**: Crear un boceto de la pantalla de inicio mostrando el menú, secciones de entrenamiento, progreso, y accesos rápidos a reservas y perfil.
* **Diseñar la estructura general e interacción**:
  + Decidir cómo interactúan las diferentes partes de la aplicación, asegurando la usabilidad y funcionalidad del sistema.
  + **Ejemplo**: Establecer la interacción entre la base de datos y el sistema de reservas, asegurando que los usuarios puedan ver y reservar en tiempo real.

**3. Analista programador (Darío)**

* **Implementar el registro de usuarios**:
  + Esto implica crear la funcionalidad completa para que los usuarios se registren y creen un perfil.
  + **Ejemplo**: Crear un formulario de registro que incluya correo, nombre, y nivel de actividad física.
* **Optimizar las consultas de la base de datos**:
  + Analizar y mejorar el tiempo de respuesta de las consultas a la base de datos, especialmente en funciones como búsqueda de entrenadores.
  + **Ejemplo**: Revisar las consultas SQL para la búsqueda de entrenadores y asegurarse de que los datos se entreguen en el menor tiempo posible.
* **Revisar y depurar código**:
  + Asegurarse de que el código es eficiente y libre de errores, mejorando así la calidad general del software.
  + **Ejemplo**: Depurar la función de inicio de sesión cuando algún usuario no pueda acceder a su cuenta.

**4. Programador (Juanma)**

* **Crear el back-end para la gestión**:
  + Desarrollar el sistema para gestionar datos clave de la app como usuarios, clases, entrenadores, y reservas.
  + **Ejemplo**: Crear la base de datos de usuarios y permitir actualizaciones de su perfil de salud y progreso físico.

**5. Arquitecto de software (Lili)**

* **Diseñar la estructura e interacción de componentes**:
  + Desarrollar un sistema donde cada parte funcione correctamente con las demás, asegurando estabilidad y seguridad.
  + **Ejemplo**: Decidir entre almacenar imágenes de los entrenadores en un servidor externo o en la misma base de datos para optimizar recursos.

**6. Jefe de proyecto (Juanma)**

* **Crear documentación técnica y de usuario**:
  + Documentar las instrucciones y detalles técnicos para programadores y crear guías de uso para los usuarios finales.
  + **Ejemplo**: Crear una guía del usuario que explique cómo reservar clases, cambiar su perfil y ver su progreso.
* **Realizar pruebas de usabilidad**:
  + Hacer que los usuarios prueben la app para identificar cualquier dificultad de navegación o diseño.
  + **Ejemplo**: Organizar una sesión de prueba en la que varios usuarios realicen tareas comunes y así detectar puntos problemáticos.