# B. Planificación:

## Sprint 0:

El primer sprint lo dedicamos principalmente para la preparación de Diagramas, mockups, primeras comunicaciones entre el equipo y el planteamiento de la idea.

Este sprint lo creamos poco antes de empezar el proyecto por lo que las tareas las creamos en el Basecamp.

Por lo que el primer Sprint lo podíamos dividir en tres partes:

- Crear Mockups de las pantallas de la app.
- Saber los requisitos técnicos generales.
- Que tecnologías utilizar y por qué.

Tareas realizadas:

- Mockup de la pantalla del ICOBlog
- Buscar documentación y testear módulos e iconos de React Native
- Mockup de la pantalla para ver la ubicación de los hospitales
- Mockup de la pantalla de contactos hospitalarios
- Mockup de la pantalla para ver el perfil del paciente
- Mockup de la pantalla de ver una cita
- Mockup Pantalla para seleccionar idioma
- Mockup de la pantalla de receta
- Mockup de la pantalla de Medicación
- Mockup de la pantalla del Calendario
- Mockup de la pantalla de Home
- Diagrama de casos de uso

## Sprint 1:

El primer sprint creado en el Jira, lo primero que hicimos fue hablar con el equipo sobre que queríamos incluir en el backlog oficial.

Además lo dedicamos para pulir las tareas realizadas en el sprint 0 de preparación.

**Tareas realizadas:**

- Rematar mockups (Usar todos la misma fuente, mismos títulos, etc)
- Arreglar diagrama
- Diagrama E-R
- Pantalla principal de la app (Main) -> Primer código real de la app
- Pantalla cambio de idioma sin funcionalidad
- Crear variables de texto que cambien el idioma
- Desplegar Beta-API (labs.iam vs AWS) -> Crear el esqueleto de la Api e intentar desplegarlo
- Documentar pasos de desplegar app

- Documentar código de las clases -> Tener presente lo de ir comentando el código

En el primer sprint pudimos completar todas las tareas que creamos.

## Sprint 2:

En este sprint la idea era crear una pequeña Api de prueba y desplegarla, y por la parte de la aplicación que esta comenzase a coger forma.

**Tareas realizadas:**

- Pantalla perfil
    - Crear frontend
    - Crear petición a la API de datos del usuario
- Añadir la gestión de las String y variables globales
- Crear componente de título de pantalla

**Tareas pendientes:**

- Pantalla contactar
    - Crear parte visual
    - Crear la llamada a la api
- Pantalla de medicación
    - Gestionar alarmas
    - Crear parte visual
    - Crear las llamadas a la API
- BackEnd-API
    - Crear Base de datos
    - Crear un perfil con informacion para testear
    - API -> Get para enviar informacion del usuario (perfil)

Empezamos a encontrar problemas con el desplegamiento de la Api, se intentó desplegar en el servidor del instituto (labs.iam.cat), después de crear diversas veces una pequeña Api y crear diversas base de datos, vimos que los puts y los deletes no nos funcionaban. Otro problema también fue que los gets que contenían objetos json tampoco iban.

Estos errores del backend hicieron que la primera versión de la pantalla de perfil cargase contra una api pública de perfiles aleatorios.

Las pantallas de medicación y contactar se empezaron pero no se pudieron acabar por tiempo.

## Sprint 3:

En este sprint debíamos tener comunicación entre la app y el servidor lista y las llamadas rest controladas, además de funciones asincronas como sería la carga dinámica de los datos y las notificaciones locales de la app .

**Tareas realizadas:**

- Modal de detalle de medicación
- Crear REEDME
- BackEnd-API
    - Crear Base de datos
    - Crear un perfil con informacion para testear
    - API -> Get para enviar informacion del usuario (perfil)

**Tareas pendienets:**

- Pantalla contactar
- Pantalla como llegar/ubicación hospitales
- Pantalla de medicación
- Métodos de la API GET i PUT

Al final creamos el reedme.md para ir haciendo la documentación, la api se estaba estancando porque habían funcionalidades que no iban, y la aplicación ya necesitaba tener la comunicación con el servidor. En la pantalla de la medicación se cambió la forma de hacer las alarmas porque para hacer una alarma “normal” necesitábamos acceder a ficheros de Android e IOS y nos estábamos quedando sin tiempo.

## Sprint 4:

Había que tener desplegado y funcionando el servidor de manera correcta porque era el último sprint, acabar las pantallas que faltaban y comprobar que la Api devolviese los datos de manera correcta.

**Tareas realizadas:**

- Pantalla ICOBlog
    - Crear frontend con el recicled view de react
    - Sub pantalla de respuestas
    - Peticiones API REST
- Pantalla del calendario
    - Añadir componente del calendario
    - Crear el listado de citas próximas
    - Controlar notificaciones
- Pantalla de medicación
    - Gestionar alarmas
    - Crear parte visual
    - Crear las llamadas a la API Rest
- API
    - añadir hospital en el GET de Perfil
    - crear una petición GET preview de Citas
    - crear un GET medicamento preview
    - añadir propiedad hora en cita Subtarea
    - añadir propiedad tipo de medicamento y dosis
    - Petición GET de preguntas y respuestas de IcoBlog
- Unión Backend con app

**Tareas pendientes:**

- Actualizar hospitales
- Actualizar medicación

Al final completamos el 90% de la app y gran parte de las funcionalidades, pero nos quedó una parte importante que por tiempo y por problemas a la hora de desplegar el servidor no la pudimos hacer. Era que la app funcionase con usuarios logeados, pero que estos usuarios solo pudiesen iniciar sesión. Pero finalmente la app se ha quedado sin pantalla de logIn.

## Conclusiones:

Aun empezando antes de tiempo quisimos abarcar más de lo que podíamos hacer, encontramos diversos problemas a la hora de desplegar que desconocíamos como solucionarlos. Por la parte de la app nos ralentizaba el ser una hecho de tratar con una tecnología que no habíamos tocado antes y no sabíamos que se podía hacer y que no de manera nativa.

A nivel de grupo la comunicación era buena, ya que hacíamos un par de reuniones a la semana, pero por los diferentes horarios de cada uno, se complicaba a veces tener las tareas sincronizadas.

Finalmente quedaron solo un par de tareas en el backlog aunque este no se actualizó correctamente.