



# PRESENTACIÓN TÉCNICA

PROYECTO: ICOAPP

GRUPO:2

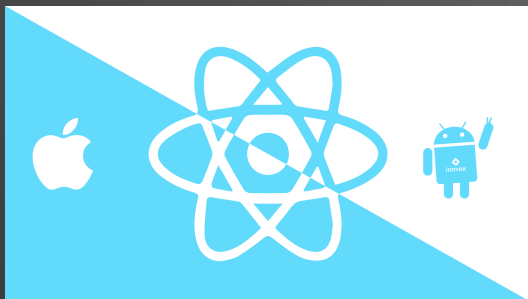
CURSO:2DAW-MATI

JAVIER MARTÍN

YAHYA EL KAJOUAI

MANEL FERRER

# RECURSOS



Symfony



ubuntu



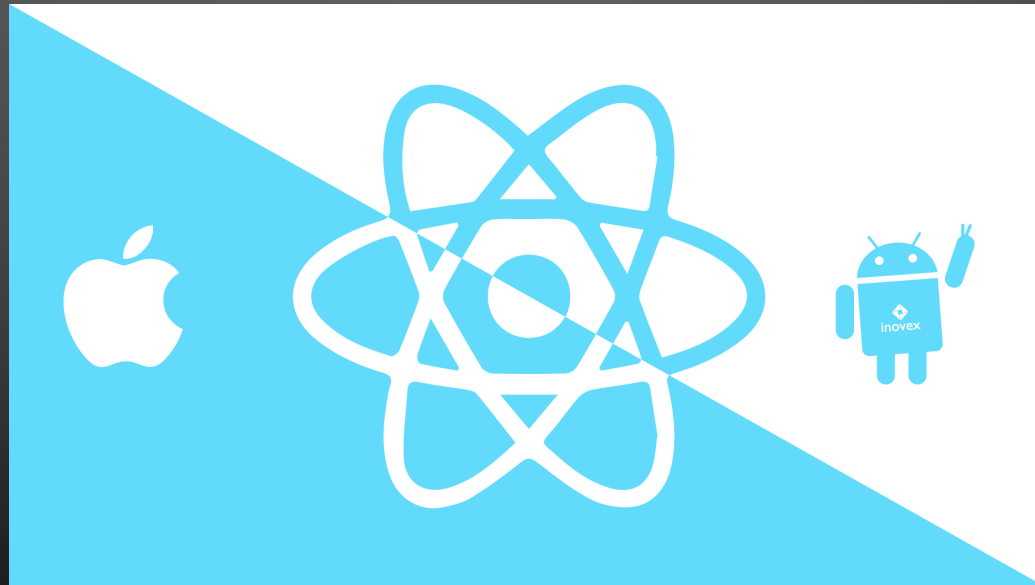
# INTRODUCCIÓN

El proyecto ICOApp es una aplicación móvil multiplataforma que ayuda a los pacientes del Institut d'Oncologia de Catalunya a gestionar y consultar sus datos. El proyecto consiste en 3 partes:

- **FrontEnd (App móvil** ➡ **React Native).**
- **BackEnd (API** ➡ **Symfony).**
- **Despliegue ( Apache, MySQL y PHPMySQL** ➡ **Labs y AWS).**

# FRONTEND

Hemos apostado por **React Native** por ser multiplataforma, por ser un **framework** reconocido y la facilidad de crear apps nativas con un lenguaje de marcas, y JS.



# BACKEND

Hemos apostado por **Symfony** por la **robustez** y **seguridad** que tiene a la hora de crear **APIs** o aplicaciones web.

En este proyecto se ha usado como motor de API el cual gestiona una base de datos **MySQL** y para administrar de manera grafica hacemos uso **PHPMysql**.

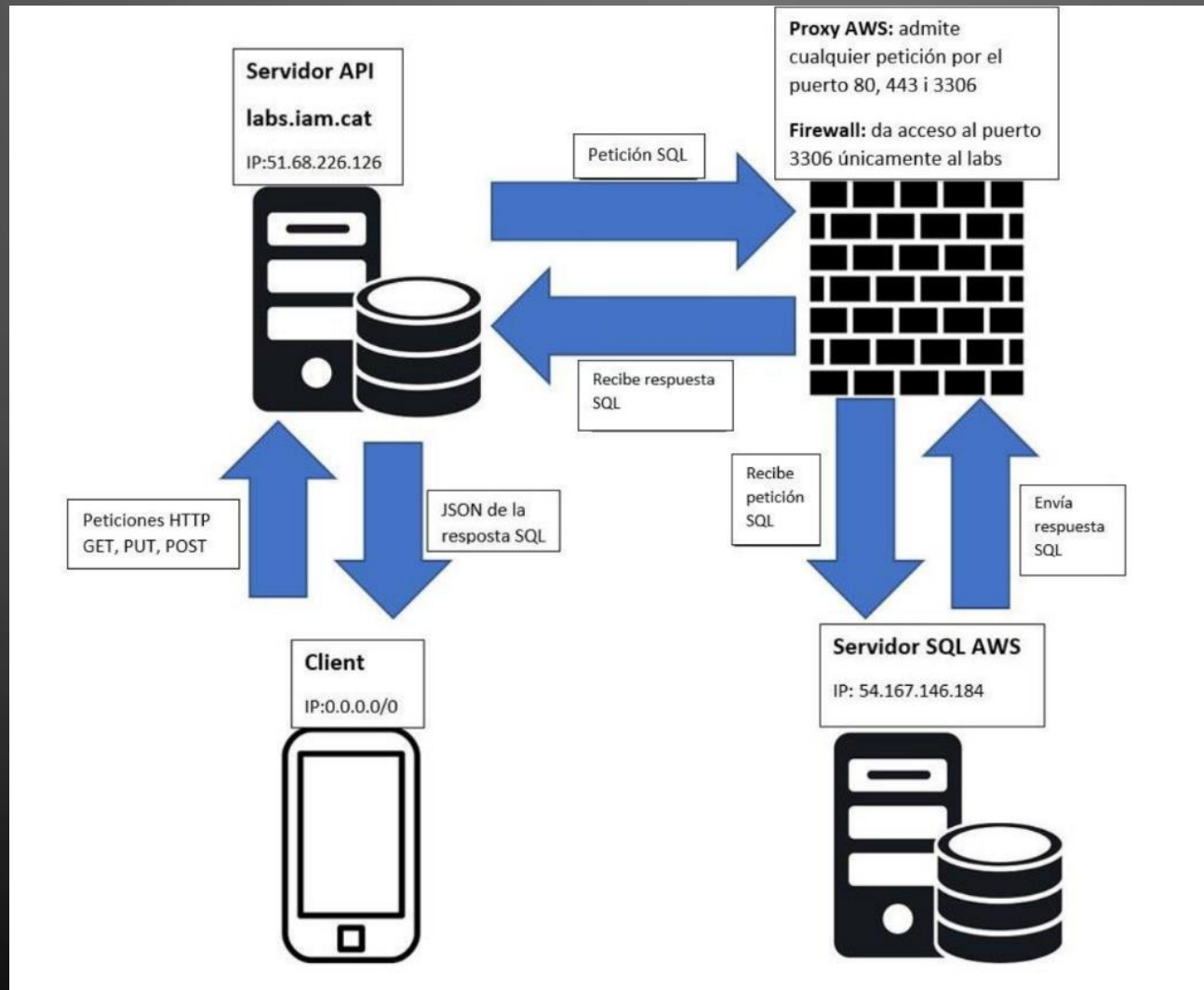


# DESPLIEGUE

Usamos dos servidores **Ubuntu**, uno de los servidores hace las peticiones **API** ([labs.iam.cat](http://labs.iam.cat)), el otro servidor es un servidor **MySQL** (máquina en **AWS**).



# ESTRUCTURA DE RED

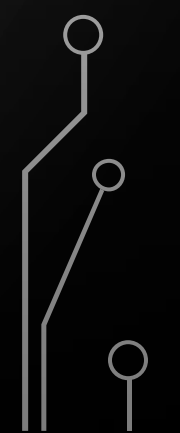





# EVOLUCIÓN DE LAS FUNCIONALIDADES AL LARGO DE LOS SPRINTS.

- 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.





# EVOLUCIÓN DE LAS FUNCIONALIDADES AL LARGO DE LOS SPRINTS.

- **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.

# EVOLUCIÓN DE LAS FUNCIONALIDADES AL LARGO DE LOS SPRINTS.

- **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.

- **Sprint 3:**

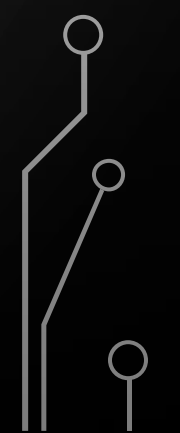

En este sprint debíamos tener comunicación entre la app y el servidor lista y las llamadas rest controladas, además de funciones asíncronas como sería la carga dinámica de los datos y las notificaciones locales de la app .



# EVOLUCIÓN DE LAS FUNCIONALIDADES AL LARGO DE LOS SPRINTS.

- **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.



# CONTRATIEMPO Y SOLUCIÓN

Contratiempo	Solución
El servidor inicial (labs) la base de datos no soporta JSON, al desplegar todo el proyecto en AWS Symfony no resolvía las rutas.	Estructura de red del backend mostrada en la anterior dispositiva.
Para crear alarmas en React Native se necesita hacerlo nativo, tocando código Android e iOS	Utilizar notificaciones locales y otros métodos para simular una alarma
La IP de AWS era dinámica y requería facilitar cada vez que se encendía la máquina	Configurar una IP elástica de AWS