JSONPlaceholder Administrator

# Introducción

El siguiente desafío se plantea en varias etapas sobre el proyecto de código abierto [JSONPlaceholder](https://github.com/typicode/jsonplaceholder), creado y mantenido por [typicode](https://github.com/sponsors/typicode), también creador de [JSON-Server](https://github.com/typicode/json-server). La idea principal es crear una aplicación web que maneje varias entidades que están relacionadas y a lo largo del desarrollo se tomen diversas decisiones de diseño y arquitectura para manejar un aplicación que crece.

Como se puede observar en la documentación de [JSONPlaceholder](https://jsonplaceholder.typicode.com/), tenemos 6 entidades, *users*, *posts*, *comments*, *todos*, *albums* y *photos*. Los users tienen todos, posts y albums, los posts tienen comments y los albums tienen photos. Todas las entidades soportan solicitudes http y https con todos los métodos (GET, POST, PUT, PATCH y DELETE). Por este motivo tenemos una oportunidad ideal para construir una aplicación tan pequeña o grande como queramos.

# Primera etapa

En una primera iteración de la aplicación nos vamos a centrar en listar los usuarios en la hidratación de la página. Una vez listados los usuarios queremos acceder a uno en particular para listar sus *todos* (un ToDo es una tarea). Esto se logra capturando el id de un user y usando ese id para hacer una consulta a /todos?userId=<id> o a la ruta anidada /users/<id>/todos. En este momento queremos permitir un CRUD (Create, Read, Udpate, Delete) o ABM (Alta, Baja, Modificación) de esta entidad.

Los usuarios tienen muchas propiedades y en el espíritu de mantener esta app sencilla no vamos a hacer CRUD de usuarios, en su lugar se propone hacer CRUD de las demás entidades.

Estos requisitos ya plantean varios desafíos a la hora de decidir qué información retener en memoria, qué y cómo cachear datos para no realizar peticiones a través de la red cuando no son necesarias, etc. Muchas de estas decisiones se toman en base a lógica de negocio o conocimiento profundo sobre el servicio (backend) que se está consumiendo pero aquí tenemos rienda suelta a probar diferentes estrategias y en ese proceso acertar o equivocarnos.

# Más allá

En lugar de ir directo a los todos de un usuario, podemos dar la opción de ir a los todos, a los albums o a los posts. Las imágenes plantearán un desafío interesante.

Eventualmente se quiere tener un CRUD de todas las entidades y esto nuevamente plantea la toma de decisiones de diseño y arquitectura de la aplicación para integrar todas las funcionalidades de manera coherente.

# Notas

Las solicitudes POST, PUT, PATCH y DELETE no generan cambios reales en el servidor, pero éste contesta como si así sucediera. Cuando estamos queriendo crear, actualizar o borrar un recurso tendremos que reflejar en memoria esa operación, idealmente cuando el servidor nos da el visto bueno y no simplemente actualizando el recurso en la vista del lado del cliente sin involucrar al servidor.