

# ADI\_2223

---

## Práctica 1: Diseño e implementación de APIs REST

---

### Fase 1: diseño y especificación del API

---

El objetivo de la práctica 1 es aprender a diseñar e implementar correctamente APIs REST. Cada uno podéis elegir el dominio que queráis para la aplicación: una tienda, una red social, un sitio educativo para cursos *online*, ...

Importante: por el momento vamos a diseñar **solamente el API**, no el sitio web. Este se desarrollará en la práctica siguiente. Eso quiere decir que por el momento **no tendremos interfaz de usuario**.

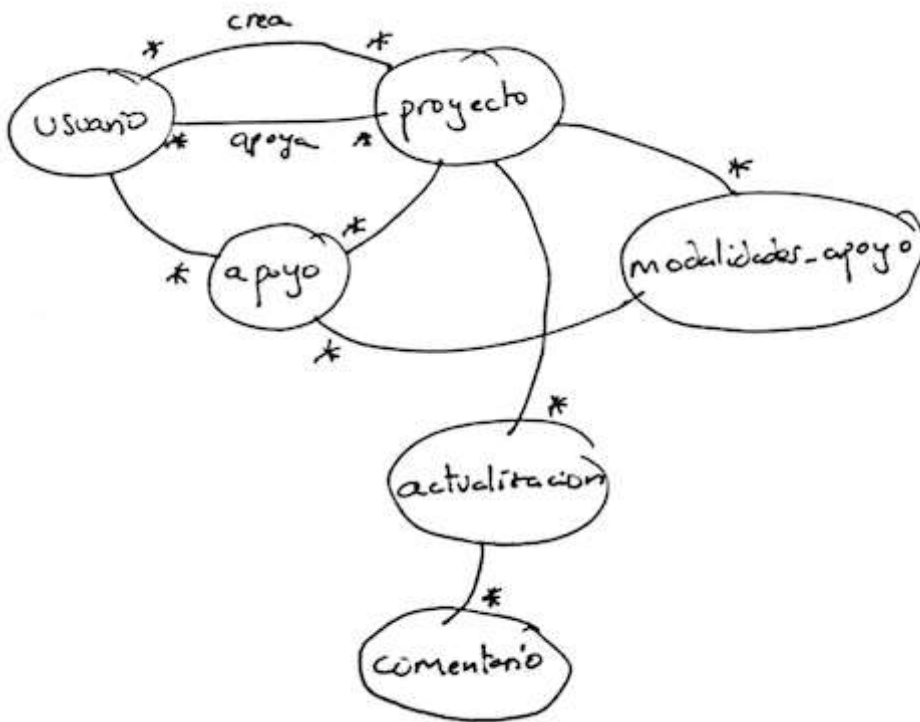
En esta fase inicial de la práctica 1 únicamente vamos a hacer el diseño inicial, especificando los recursos, los casos de uso y las llamadas al API. En la siguiente fase desarrollaremos la implementación.

#### 1. Identificación de los recursos

---

Debéis identificar los recursos que vais a modelar en el API, que se corresponderán con las entidades del dominio de vuestra aplicación. Por ejemplo en un API para un sitio de *crowdfunding* tendríamos "proyectos", "usuarios", "modalidades de apoyo" (una recompensa que se obtiene cuando un usuario apoya un proyecto con una determinada cantidad), etc. Identificad también los campos de cada recurso. Por ejemplo un proyecto tendrá un título, una descripción, una fecha límite, etc.

Además de **identificar los recursos y sus campos** debéis identificar las **relaciones entre recursos**. Dibujad un grafo que muestre estas relaciones. No es necesario que uséis una notación estándar para las relaciones, ni tampoco que dibujéis con una herramienta gráfica, podéis simplemente dibujar a mano y hacer una foto.



En este ejemplo (cutre, lo sé, pero no espero que hagáis nada más sofisticado) los arcos indican relaciones y los \* indican a muchos.

## 2. Formulación de los casos de uso simplificados

Con "simplificados" queremos decir que basta con formularlos de forma sencilla en un par de líneas. En nuestro ejemplo del *crowdfunding* serían algo como:

- Un usuario sin estar autenticado debe poder ver los datos más importantes de la lista de proyectos más populares en el sitio
- Un usuario sin estar autenticado debe poder ver todos los datos de un proyecto
- Un usuario autenticado debe poder elegir una modalidad de apoyo y apoyar un proyecto con esa cantidad
- El usuario que ha creado un proyecto, si está autenticado debe poder enviar actualizaciones (=noticias) sobre el estado del mismo
- Un usuario debe poder hacer login en la aplicación ...

En la fase de implementación *no será necesario que implementéis todos los casos de uso*, así que no os quedéis cortos en este apartado, cuanto más pongáis más os hacéis una idea del potencial y las posibilidades de vuestro API. El enunciado de la fase de implementación se publicará la semana del 3 al 9 de octubre.

## 3. Diseño básico del API REST

Tenéis que transformar los casos de uso anteriores en llamadas al API. Estas llamadas deben estar orientadas a satisfacer los casos de uso, en general habrá una correspondencia 1:1 entre

caso de uso y llamada al API, aunque no es necesario. *Es posible que para resolver un caso de uso hagan falta varias llamadas al API. Por ejemplo es posible que para ver un proyecto incluyendo todas las novedades (actualizaciones, en el diagrama) hagan falta dos llamadas al API: una para los datos básicos del proyecto y otra para las actualizaciones.*

Para cada llamada al API tenéis que especificar:

- Formato de la petición HTTP
  - La URL
  - El método HTTP (GET, POST, PUT o DELETE)
  - Si el cliente envía datos al servidor cómo lo hace: en parámetros HTTP o en el cuerpo de la petición
  - Cabeceras HTTP que debe enviar el cliente, si las hay
- Formato de la respuesta HTTP
  - Código de estado si éxito
  - Códigos de estado de los errores más habituales
  - Cabeceras HTTP que debe enviar el servidor, si las hay
  - Contenido del cuerpo de la respuesta en caso de éxito
  - Contenido del cuerpo de la respuesta en caso de error, para al menos los 2-3 errores que creéis que pueden ser más habituales

De momento, en este apartado ignoraremos los casos de uso de login/logout y todo lo relativo a autenticación del usuario. Por ejemplo es evidente que en el caso del *crowdfunding* no deberíamos poder aportar dinero a un proyecto sin estar logueados, ni tampoco borrar el proyecto si no somos sus creadores. Como veremos, para identificarnos en el API típicamente se deben enviar unas cabeceras HTTP especiales en la petición, pero de momento vamos a ignorar toda esta cuestión. Lo veremos en la sesión de teoría del 4 de octubre.

## 4. Formato y plazo de entrega

El plazo de entrega de la documentación concluye el **viernes día 7 de octubre a las 23:59**. La entrega se realizará en moodle enviando un único archivo comprimido en .zip o similar.

Se entregará un archivo con la documentación de casos de uso, de relaciones entre llamadas al API y casos de uso y el diagrama de relaciones entre recursos. Puede ser tan simple como un txt y un png, o un pdf. La extensión de la documentación estará entre 1-3 páginas aproximadamente (puede ser mayor pero no es necesario para la máxima nota).

Recordad que la documentación consta de:

- Recursos del API, cada uno con sus campos.
- Diagrama especificando las relaciones entre los recursos.

- Listado de casos de uso.
- Especificación de llamadas al API usando la plantilla que se describe en el apartado 3.

**IMPORTANTE:** es bastante probable que cuando estéis implementando la práctica os deis cuenta de que queréis hacer modificaciones en el diseño inicial: añadir o modificar algún recurso, cambiar sus campos, cambiar algún caso de uso o llamada del API... No hay problema siempre que en la práctica completa entreguéis las modificaciones al diseño inicial.

**La documentación no se puntuará por separado sino que formará parte del baremo de la práctica 1 completa,** con una puntuación máxima de 1.5 puntos.