**Proyecto: Mockup visual -> modelo de datos -> API**

Página de login/registro:

* Usuario
* Contraseña

Página principal:

* Posts:
  + Foto
  + Contenido principal (nombre y foto de perfil de la persona que posea/comenta)
  + Botones de like, compartir, comentar
  + Comentarios (nombre y foto de perfil de la persona que comenta)
* Menú principal:
  + Perfil (nombre usuario y @, foto perfil)
  + Buscador
  + Amigos (nombres de amigos y foto en miniatura)
  + Mensajes directos (nombres y fotos de usuarios)
  + Quizás conozcas (nombre y foto de perfil de otros usuarios)
* Página de perfil:
  + Nombre
  + Edad
  + Foto de perfil
  + Actividad en la que participa
  + Feed con datos indicados en “posts”
  + Descripción personal

Usuario:

* Nombre y @usuario
* Foto de perfil
* Contraseña

Comentarios:

* Información de usuario que comenta
* Texto

Mensajes:

* Información de usuario(foto, nombre) que manda el mensaje
* Texto descriptivo del mensaje (primeros caracteres).

Dentro del mensaje:

* Ventana de chat
* Caja de texto para escribir
* Botones de emoticonos ¿y adjuntar archivo?.

Actividad:

* Disciplinas deportivas en las que participa
* Club, gimnasio, etc al que asiste o participa

Que entidades hay y que queremos hacer con ellas

Entidades:

Usuario: GET, POST, DELETE

Usuarios : GET/users/[n] – Enfocado a búsquedas o a usuarios que pertenecen a grupos.

Publicaciones de usuarios: POST/publicación/ DELETE/publicación

Mensajes de usuarios: GET/mensajes POST/mensaje/

Grupos (aquí se incluye gimnasios, organizaciones, clubes, etc.): GET, POST, DELETE, PATCH

GET

POST

PUT 401,404,500

DELETE

PATCH: no se necesitan todos los campos

200 – Get

201 – Post

500 – Por si pasa algo en el servidor

400 – Post o get (con parámetros(id, pass, etc)).

Lo de arriba:

* Vistas: Entrar, registrarse, feed principal, perfiles personales/organizaciones, mensajes directos
* Componentes: Imagen circular

Lo del medio:

* Mockoon

Lo de abajo:

* Controllers: INSERT usuario o grupo, INSERT post o comentario, INSERT mensajes privados.
* Middleware: Autorización para POST usuario/grupo, POST publicación, POST comentario en publicación, POST mensaje privado,
* Models: Bases de datos(SQL): bd usuarios/grupos, bd posts asignados a id usuarios/grupos (con comentarios incluidos dentro de cada uno), bd historial de mensajes privados asignados a id usuario.

Respuestas ENDPOINTS mockoon gets - V

Definir tablas SQL:

Tablas SQL:

* USUARIOS: id, name, password, group(false/true)
* POSTS: id, content, id\_fromuser
* Comentarios: id, content, id\_frompost
* Mensajes privados: id, content
* AMIGOS: id, id\_friend, name, group

**Componentes, definirlos y datos necesarios:**

**Circulo avatar**: Recibe imagen de usuario.

**Nombre @nombre:** Nombre de usuario o de grupo.

**Panel de control lateral:** 5 botones que llevan a distintas vistas

**Publicación**: Recibe datos de usuario que la postea, cuerpo de la publicación (contiene texto y multimedia que haya añadido el usuario).

**Cajón de comentarios:** Muestra nº de comentarios y los 3 primeros comentarios. Botón ver más para enseñar más comentarios.

**Botones de reacción:**

* Like (un contador¿?)
* Compartir: duplica el post y permite repostearlo indicando la info del op
* Comentar: Permite insertar comentario en el post

**Resultado búsqueda**: **circulo avatar** referente a la busqueda + C**omponente búsqueda:** datos de la búsqueda (nombre de persona/club, nº miembros/amigos en común, localización geográfica). Muestra el resultado de la búsqueda en texto.

**Mensajes directos**: Nombre de persona con la que se está hablando y preview de mensaje.