

Quacker



Vamos a desarrollar una aplicación web al estilo X (anteriormente Twitter), centrándonos en la parte de backend. Para ello, haremos un análisis previo del diseño lógico que debe cumplir la aplicación, basándonos en el patrón de diseño de **Modelo Vista Controlador** (MVC). Para llevar a cabo el desarrollo haremos uso de **Laravel** como framework de PHP para la parte de backend, dejando libertad para elegir cómo plantear la parte de frontend a criterio y conocimientos de cada uno.

Funcionalidad mínima esperada

Un usuario (**User**) debe ser capaz de:

- **Registrarse:** darse de alta en la aplicación
- **Iniciar sesión:** iniciar sesión en la aplicación
- **Modificar datos personales:** modificar sus datos personales y eliminar su cuenta
- **Quackear:** escribir una pequeño mensaje de carácter público e indexable en el feed de sus seguidores
- **Requackear:** compartir el Quack de otro usuario en el feed de sus seguidores
- **Quavear:** indicar que le gusta el Quack de otro usuario, pero sin compartirlo en el feed de sus seguidores
- **Seguir:** seguir a otro usuario para que sus Quacks y Requacks aparezcan en su feed
- **Dejar de seguir:** dejar de seguir a otro usuario para que sus Quacks y Requacks desaparezcan de su feed
- **Buscar:** buscar Quacks por contenido o por Quashtag

Un **Quack** tiene:

- **Contenido:** mensaje súper interesante y nada tóxico que un usuario comparte con el mundo
- **Fecha de publicación:** fecha y hora de publicación

- **Número de Quavs:** recuento de usuarios que han hecho Quav
- **Número de Requacks:** recuento de usuarios que han hecho Requack
- **Quashtag:** palabra precedida de un carácter especial que facilita su búsqueda e indexación en vistas específicas

Vistas esperadas

Con respecto a **User**:

- Formulario de registro
- Formulario de login
- Modificación de datos personales

Con respecto a **Quack**:

- Formulario de creación
- Feed personal: aquí aparecen tus Quacks, Requacks y los Quacks y Requacks de usuarios a los que sigues
- Feed de usuario: feed de un único usuario, donde solo aparecen sus Quacks, Requacks y Quavs
- Feed de búsqueda: aquí aparecen todos los Quacks que cumplen con el criterio de búsqueda, independientemente de si son de usuarios a los que sigues o no
- Quack: vista de un único Quack, con más detalles que los que se muestran en un feed

Con respecto a **Quashtag**:

- Lista de Quahstags populares: según criterio, ordena los Quashtags por popularidad y muestra un resumen (por ejemplo: número de Quacks en las últimas 24 horas)

Modo de trabajo

El desarrollo se dividirá en diferentes hitos, siendo estos, aproximadamente, los siguientes:

- **Hito 1:** análisis y diseño
- **Hito 2:** CRUD sobre Quack
- **Hito 3:** sesiones y autenticación de usuarios
- **Hito 4:** relaciones (hasMany, belongsToMany...) y cookies
- **Hito 5:** desarrollo de API
- **Hito 6:** consumo de APIs de terceros

Con respecto a la gestión del proyecto, sus cambios y liberación de versiones para las entregas de los hitos, se hará uso de **git + GitHub**