

Quacker - Atando cabos

Antes de continuar, aplicad las correcciones que consideréis sobre las entregas anteriores. Basándoos en ello, vamos a completar la interacción con la aplicación. Se espera lo siguiente:

Rutas

`/`

Si el usuario no está autenticado, deberá redirigir a `/login`

Si el usuario está autenticado, deberá redirigir a `/feed`

`/login`

Formulario de inicio de sesión. Deberá implementar validación de datos

- **Email:** no vacío, en formato email
- **Contraseña:** no vacío

`/feed`

Lista de Quacks. Deben mostrarse las siguientes fuentes:

- Quacks y requacks del usuario autenticado
- Quacks y requacks de los usuarios a los que sigue el usuario autenticado

Deberán estar ordenados cronológicamente (de más reciente a más antiguo) siguiendo las siguientes normas:

- Si se trata de un Quack, según la columna `created_at`

- Si se trata de un requack, según la columna `created_at` de la tabla intermedia, no del Quack original

`/users/{user}/quacks`

Lista de Quacks. Deben mostrarse las siguientes fuentes:

- Quacks y requacks de ese usuario

Deberán estar ordenados cronológicamente (de más reciente a más antiguo) siguiendo las siguientes normas:

- Si se trata de un Quack, según la columna `created_at`
- Si se trata de un requack, según la columna `created_at` de la tabla intermedia, no del Quack original

`/quacks/{quack}`

Detalles de un Quack concreto

`/users/{user}`

Detalles de un Usuario concreto

`/users/edit`

Vista de edición de los datos personales del usuario autenticado

`/quashtags/{quashtag}/quacks`

Lista de Quacks. Deben mostrarse las siguientes fuentes:

- Quacks que tengan ese Quashtag asociado

Deberán estar ordenados cronológicamente (de más reciente a más antiguo) usando la columna `created_at`

Información a mostrar

Esta información deberá mostrarse en cada una de las vistas en las que aparezcan representados

User

- Nickname
- Número de `Quacks` publicados
- Número de seguidores
- Número de seguidos
- Popularidad: recuento de Quavs que tengan todos sus Quacks
- Viralidad: recuento de Requacks que tengan todos sus Quacks

Quack

- Texto del propio Quack
- Nickname del `User` propietario. Deberá tener un enlace a `/users/{user}/quacks`
- Fecha de publicación
- Lista de `Quashtags` asociados (el nombre de cada uno). Cada uno deberá tener un enlace a `/quashtags/{quashtag}/quacks`
- Número de Requacks
- Número de Quavs

Interacción

User

- Si se trata de un usuario diferente al autenticado, las siguientes opciones:
 - Si no sigo a ese usuario: Seguir
 - Si sigo al usuario: Dejar de seguir
- Si se trata de mi usuario, la siguiente opción:

- Editar perfil: enlace a `/users/edit`

Quack

- Si soy el propietario: **eliminar**
- Si no lo he requackeado: **requackear**
- Si lo he requackeado: **eliminar requack**
- Si no lo he quaveado: **quavear**
- Si lo he quaveado: **eliminar quav**
- Un `Quack` se creará desde un formulario disponible en `/feed`
 - Si el texto contiene una palabra precedida del carácter `#`, se le asociará un `Quashtag` con dicho nombre. Por ejemplo: "Hoy es un buen día para `#quackear`" deberá crear un `Quack` con el texto "Hoy es un buen día para `#quackear`" y asociarle un `Quashtag` con nombre "quackear"
 - Si el `Quashtag` no existe, deberá crearse una instancia y una entrada en la tabla intermedia
 - Si el `Quashtag` sí existe, deberá crearse solamente una entrada en la tabla intermedia
 - Deberemos hacer esto para todos los `Quashtags` del texto

Esto puede ser útil, adáptalo a tu código:

```
# 1
preg_match_all('/#(\w+)/u', $quack->content, $matches);
$quashtagNames = array_unique($matches[1]);

# 2
$quashtags = [];
foreach ($quashtagNames as $quastagName) {
    $quashtags[] = Quashtag::firstOrCreate(['name' =>
$quastagName]);
}

# 3
if (!empty($quashtags)) {
    $quack->quashtags()->saveMany($quashtags);
}
```

```
}
```

1. Aplica una expresión regular para encontrar todas las palabras precedidas por `#`. Al hacer `$matches[1]` se accede al primer grupo de cada match, que se corresponde con la palabra sin el `#`
2. Recorremos los nombres de los `Quashtags` y los buscamos o los creamos si no existen
3. Asociamos todos los `$quashtags` a la relación `quashtags()` de `$quack`. Esto se traduce en la creación de entradas en la tabla intermedia `quack_quashtag`

Normas de entrega

Deberéis entregar un fichero **repositorio.txt** con un enlace al repositorio.

El repositorio **DEBERÁ CONTENER** una publicación (release) disponible:

- **NO** voy a hacer **clone** de ningún proyecto

Además, la release **DEBERÁ CONTENER** un fichero con instrucciones de despliegue:

- **NO** voy a pelearme para hacer funcionar vuestro proyecto
- **SÍ** voy a seguir al pie de la letra las instrucciones de despliegue
 - Si me indicáis que debo levantar una BBDD MySQL, lo haré. Si no funciona en MySQL, asumiré que no funciona, en general

El fichero con instrucciones de despliegue **DEBERÁ CONTENER** las credenciales de un usuario de ejemplo con el que pueda probar la autenticación:

- **NO** voy a mirar **Factory/Seeder** buscando el usuario