

Presentación 2 TSC

Fabián Da Silva R.
Felipe Mancilla S.
Néstor Venegas P.

Requerimientos a Implementar

1. Registro de Usuario.
2. Crear, Eliminar y Modificar Notas.
3. Crear, Eliminar y Modificar Etiquetas.
4. Asignar etiquetas a una nota.
5. Desplegar la información de notas y etiquetas de cada usuario.
6. Autenticación Básica.

Algunas Consideraciones

- Ocuparemos la gema “rails-api” para configurar nuestro proyecto como API REST.
- Usaremos las gemas “jbuilder” y “aescrypt” para usar las vistas “json” de nuestro proyecto y para encriptar los passwords de los usuarios, respectivamente.

Registrar Usuario

- Creamos el modelo “User” con los atributos pedidos, su controlador y las vistas json respectivas.
(rails g scaffold User username:string password:string fullname:string email:string).
- Migramos su tabla a la base de datos.
(rake db:migrate).

CRUD Notas

- Creamos el modelo “Note” con los atributos pedidos, su controlador y las vistas json respectivas.
(rails g scaffold Note title:string description:string user:references).
- Relacionamos el modelo “Note” con el modelo “User”.
(belongs_to :user, has_many :notes).
- Migramos su tabla a la base de datos.
(rake db:migrate).

CRUD etiquetas.

- Creamos el modelo “Tag” con los atributos pedidos, su controlador y las vistas json respectivas.
(rails g scaffold tag tagname:string description:string user:references).
- Relacionamos el modelo “Tag” con el modelo “User”.
(belongs_to :user, has_many :tags).
- Migramos su tabla a la base de datos.
(rake db:migrate).

Asignación de Etiquetas

- Creamos una tabla intermedia “note_tags” (rails g migration CreateNotesTag note:references tag:references).
- Hacemos la relación entre los modelos “Note” y “Tag”.
(has_and_belongs_to_many :notes y has_and_belongs_to_many :tags).
- Configuramos su controlador para asignar una etiqueta a una nota.
(def assign) y agregamos esta acción a una ruta específica.
- Migramos su tabla a la base de datos.
(rake db:migrate).

Desplegar Información

http://localhost:3000/ x + SurveyMonkey Env

GET http://localhost:3000/users Params Send Save

Body Cookies Headers (9) Tests Status: 200 OK Time: 233 ms

Pretty Raw Preview JSON

```
1- [{
2-   "user": {
3-     "id": 1,
4-     "username": "famancil",
5-     "password": "holal23",
6-     "fullname": "Felipe Alexis Mancilla Sepulveda",
7-     "notes": [
8-       {
9-         "id": "Primera Nota",
10-        "description": "Este es mi primera nota",
11-        "tags": [],
12-        "user_id": 1
13-      }
14-    ]
15-  },
16-   "url": "http://localhost:3000/users/1"
17- },
18- {
19-   "user": {
20-     "id": 2,
21-     "username": "fadasilva",
22-     "password": "holal234",
23-     "fullname": "Fabian Alexander Da Silva Retamales",
24-     "notes": [
25-       {
26-         "id": "Segunda Nota",
27-         "description": "Este es mi segunda nota"
```

http://localhost:3000/ x + SurveyMonkey Env

GET http://localhost:3000/users Params Send Save

Body Cookies Headers (9) Tests Status: 200 OK Time: 233 ms

Pretty Raw Preview JSON

```
15-   },
16-   "url": "http://localhost:3000/users/1"
17- },
18- {
19-   "user": {
20-     "id": 2,
21-     "username": "fadasilva",
22-     "password": "holal234",
23-     "fullname": "Fabian Alexander Da Silva Retamales",
24-     "notes": [
25-       {
26-         "id": "Segunda Nota",
27-         "description": "Este es mi segunda nota",
28-         "tags": [
29-           {
30-             "tagname": "Primera Etiqueta",
31-             "description": "Este es mi primer etiqueta",
32-             "user_id": 2
33-           }
34-         ],
35-         "user_id": 2
36-       }
37-     ],
38-     "url": "http://localhost:3000/users/2"
39-   }
40- ],
41- ]
42- }
```


Autenticación Básica

- Incluimos la librería
“`ApiController::HttpAuthentication::Basic::ControllerMethods`”
para ocupar sus métodos de autenticación.
- Definimos una función en el controlador de “User” para autenticar a un usuario que se encuentra en el Sistema.

Presentación 2 TSC

Fabián Da Silva R.
Felipe Mancilla S.
Néstor Venegas P.