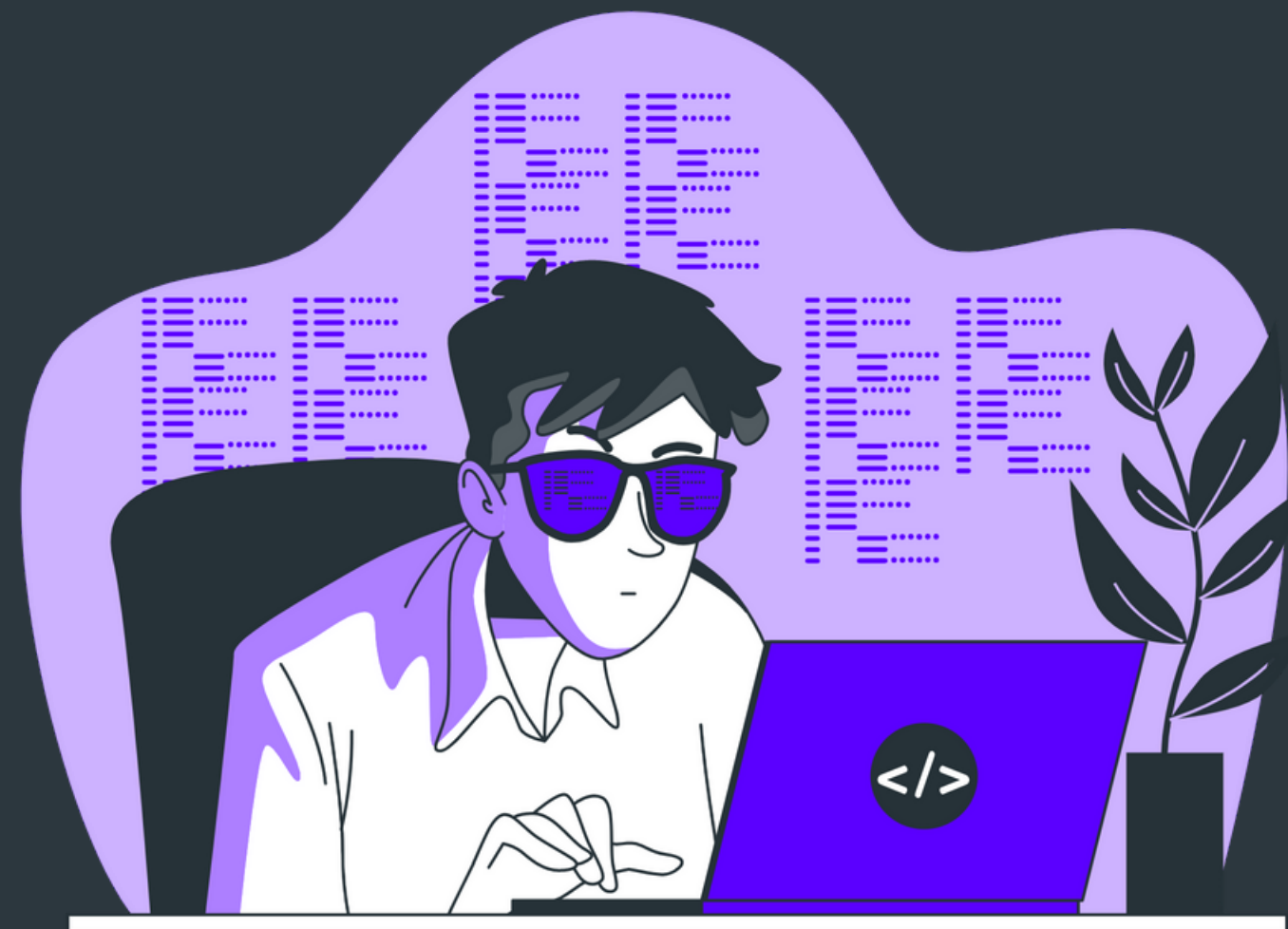


Front

End

2



To-Do: el proceso

Contenido

1. Ver tareas
2. Marcar una tarea como terminada
3. Crear nuevas tareas
4. Crear un nuevo usuario
5. Ingresar a nuestra cuenta
6. Ver la fecha de creación de una tarea
7. Comentarios

1- Ver tareas

Ver tareas

Hacemos un GET al endpoint

`https://ctd-todo-api.herokuapp.com/v1/tasks`

En el header de la petición enviamos el token obtenido del `localStorage`

Parseamos el JSON que nos devuelve la API y enviamos el array como parámetro a la función `crearTareas(tareas)`

Ver tareas

Esta función itera sobre todas las tareas llamando en cada iteración a la función encargada de renderizarla.

`renderizarTarea(t)` es quien determina en base al estado de la tarea si debe renderizarla en el contenedor de tareas pendientes o terminadas.

2- Marcar una tarea como
terminada

Marcar una tarea como terminada

Hacemos un PUT al endpoint

<https://ctd-todo-api.herokuapp.com/v1/>

En el header de la petición enviamos el token obtenido del `localStorage`

En el body enviamos un JSON que tiene el nuevo estado de la tarea

Marcar una tarea como terminada

Personalmente, como no quería vaciar los contenedores y llamar a `crearTareas(tareas)` para que renderice todo de cero, decidí crear una función que recibe el `id` y un `booleano` y se encarga a través de `appendChild(t)` de moverla al contenedor correspondiente.

Marcar una tarea como terminada

En esta instancia me costó resolver cómo modificar el comportamiento una vez que la tarea había sido trasladada.

Finalmente lo que terminé haciendo fue: luego de cambiarlo de contenedor, le agrego un `eventListener` que llame a esa misma función pero negando el `booleano` que recibe inicialmente.

3- Crear nuevas tareas

Crear nuevas tareas

Agregamos el `addEventListener` `onsubmit` al formulario (que consiste de un `input`) con el llamado a `agregarTarea()`

En esa función hacemos un POST al endpoint `https://ctd-todo-api.herokuapp.com/v1/tasks`

En el body enviamos un JSON con el valor obtenido del `input` y en el header la autorización

Crear nuevas tareas

Parseamos la respuesta de la API y enviamos el objeto literal como parámetro a la función `crearTarea(tarea)`

Parseamos la respuesta de la API y enviamos el objeto literal como parámetro a la función `crearTarea(tarea)`

Otra forma de hacer esto es como vimos en clase, usando el `onclick` en el template string creado en `renderizarTarea(t)`

4- Crear un nuevo usuario

Crear un nuevo usuario

Agregamos el `addEventListener` `onsubmit` al formulario para validar los datos, y luego si son correctos crear el nuevo usuario

Si todos los datos son válidos hacemos un POST al endpoint

`https://ctd-todo-api.herokuapp.com/v1/users`

Para algunas validaciones utilizamos expresiones regulares.

Crear un nuevo usuario

En el body de la petición enviamos un JSON con los datos que el usuario ingresa

Parseamos la respuesta de la API y guardamos el `jwt` en el `localStorage` bajo la clave `token`

Esto es lo que nos permite obtener las tareas, crear nuevas y modificar existentes

5- Ingresar a nuestra cuenta

Ingresar a nuestra cuenta

El proceso es muy similar a como creamos nuestra cuenta

En este caso luego de obtener los valores de los **inputs** no validamos, simplemente hacemos un POST al endpoint

<https://ctd-todo-api.herokuapp.com/v1/users/login>

Si todo salió bien, procedemos al igual que antes a guardar en el **localStorage** el **token**

6- Ver la fecha de creación
de una tarea

Ver la fecha de creación de una tarea

Esto lo hacemos al definir el template string en la función `renderizarTarea(t)`

Para que muestre de manera más legible para el usuario decidí crear una función que parsee lo que recibimos de la API a un formato `DD/MM/AAAA`

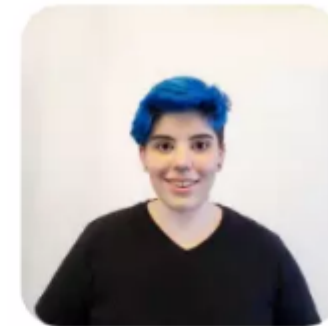
7- Comentarios

Me gustó mucho el proyecto y siento que me sirvió para unificar todos los conceptos que veníamos viendo clase a clase.

En general no tuve dificultades con lo que se planteó para hacer en las clases, sólo con los desafíos personales que me iba poniendo al refactorizar e intentar mejorar lo que veníamos haciendo.


Proyecto Completo

valva-ro/**ToDo**



 1
Contributor

 0
Issues

 0
Stars

 0
Forks



valva-ro/ToDo

Contribute to valva-ro/ToDo development by creating an account on GitHub.

 GitHub