

Mostrando la lista de dashboards (II)

Resumen

- En `Main.js` pediremos la lista de `dashboards` y los almacenaremos en un nuevo estado `dashboards`
- Mediante `dashboards.map` transformaremos cada elemento en un `<DashboardListItem>`

Descripción

Para ayudarnos a entender lo que haremos en esta tarea, nos acompañará una metáfora: Queremos montar un museo de juguetes.

Para ello, pediremos los juguetes desde el museo. Cuando los recibamos, guardaremos las cajas. Dentro del museo, los sacaremos de las cajas y los prepararemos para la exposición.



¡Vamos a ello!

La tarea

Lo primero será *pedir los juguetes al proveedor, y recibirlas en sus cajas*:

Añade a `Main.js` un estado `dashboards`:

```
const [dashboards, setDashboards] = useState([])
```

Más abajo, en `Main.js`, añade un `useEffect` que se encarga de pedir los datos (GET) a <http://raspi:8081/api/v1/dashboards> y usa `.then` para gestionar la respuesta:

```
useEffect(() => {
  axios.get('http://raspi:8081/api/v1/dashboards').then(response => {
    setDashboards(response.data);
  })
}, [])
```

Ahora toca *sacar los juguetes de las cajas y mostrarlos*.

En `Main.js`, añade un `<div data-cy='dashboardsList'>` debajo del `<h2>` (antes del `<footer>`):

```
<div data-cy='dashboardsList'>
</div>
```

Y, dentro de él, invoca `.map` sobre el estado `dashboards`, de forma que cada `dashboard` sea transformado en un `<DashboardListItem />`.

Será fácil, pues en la tarea anterior ya preparamos `DashboardListItem.js` para que recibiera una `prop dashboard` (recuerda: Es un objeto con `id`, `title` y `description`):

```
<div data-cy='dashboardsList'>
  { dashboards.map((d) => { return <DashboardListItem dashboard={d}/> })
}</div>
```

(Acuérdate de importar `DashboardListItem`)

¡Los juguetes ya están montados y listos para exponer!

Verifica que puedes ver la lista de *dashboards* en <http://localhost:3000>

Por último

Sube tus cambios al repositorio en un nuevo *commit*.



Rubén Montero @ruben.montero changed milestone to %Sprint 3 3 hours ago