Open Opened 3 weeks ago by Rubén Montero

## Subir (POST) una pregunta (II)



- Añadiremos los estados formTitle y formDescription a DashboardDetailPage.js
- Los vincularemos a los <input> del formulario
- Cuando cliquemos en *submit* se verá un console.log . ¡Todavía no se enviará la petición HTTP!

### Descripción

Vamos a hacer nuestro formulario un poquito más funcional... ¡Aunque todavía no del todo!



#### La tarea

**Añade** dos nuevos estados formTitle y formDescription a DashboardDetailPage.js:

```
const [formTitle, setFormTitle] = useState('');
const [formDescription, setFormDescription] = useState('');
```

Ahora vamos a hacer que los estados reflejen lo que el usuario va tecleando en el formulario.

Primero, asocia value (en los <input> del formulario) con formTitle y formDescription:

```
- <input data-cy='newQuestionTitle' placeholder='Título de la pregunta'></input><br/>
- <input data-cy='newQuestionDescription' placeholder='Texto de la pregunta'></input><br/>
+ <input data-cy='newQuestionTitle' placeholder='Título de la pregunta' value={formTitle}></input><br/>
+ <input data-cy='newQuestionDescription' placeholder='Texto de la pregunta' value={formDescription}></input><br/>
- <input data-cy='newQuestionDescription' placeholder='Texto de la pregunta' value={formDescription}>
```

A continuación, **crea** estas dos funciones que modifican los valores internos de **formData**:

```
const handleChangeTitle = (event) => {
   setFormTitle(event.target.value);
}

const handleChangeDescription = (event) => {
   setFormDescription(event.target.value)
}
```

j...y asócialas al onChange de cada campo del formulario!

```
- <input data-cy='newQuestionTitle' placeholder='Título de la pregunta' value={formTitle}></input><br/>- <input data-cy='newQuestionDescription' placeholder='Texto de la pregunta' value={formDescription}></input><br/>- <input data-cy='newQuestionTitle' placeholder='Título de la pregunta' value={formTitle} onChange={onChangeTitle}></input><br/>- <input data-cy='newQuestionDescription' placeholder='Texto de la pregunta' value={formDescription} onChange={onChangeDescription}</pre>
```

Ya tienes vinculados los textos del formulario a los estados. Son como un espejo. Lo uno refleja lo del otro.

#### onSubmit

Para terminar, añade esta nueva función a DashboardDetailPage.js:

```
const handleSubmit = (e) => {
    e.preventDefault();
    if ((formTitle.length === 0) || (formDescription.length == 0)) {
        return;
    }
    console.log("Quieres publicar la pregunta con título:")
    console.log(formTitle);
    console.log("...y descripción:");
    console.log(formDescription);
```

https://raspi/francisco.gomez/dwec/issues/120

```
// Vaciar el formulario
  setFormTitle('');
  setFormDescription('');
}
```

...y **asóciala** al onSubmit del <form>:

```
<form data-cy='newQuestionForm'>
+
        <form data-cy='newQuestionForm' onSubmit={handleSubmit}>
```

¡Listo!

Ya puedes verificar en la página de un dashboard como <a href="http://localhost:3000/dashboards/1">http://localhost:3000/dashboards/1</a> que tu DashboardDetailPage.js entiende los valores del formulario.

Simplemente, abre Inspeccionar > Consola y comprueba que el mensaje sale correctamente al accionar el formulario.



# Por último

Sube tus cambios al repositorio en un nuevo commit.



Rubén Montero @ruben.montero changed milestone to <a href="Msprint">MSprint 4</a> 3 weeks ago