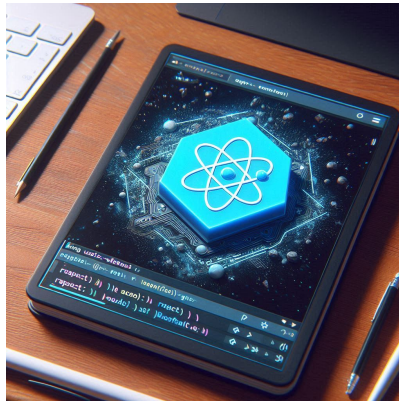


Examen de JavaScript y React



Instrucciones Generales

Crea una aplicación de **Encuestas** utilizando Vite y React 18. La aplicación debe permitir a los usuarios editar las encuestas, eliminar encuestas, así como ver estadísticas sobre las respuestas de las encuestas. Debe cumplir con los siguientes requisitos:

- Debe estar hecha con Vite y React 18.
- Debe usar Contextos Globales.
- Debe usar Rutas con React Router Dom. En las rutas debe haber children que se rendericen en Outlet.
- Debe usar una API local con JSON Server.
- Debe guardar y leer datos en el LocalStorage.

Ejercicio 1: Configuración del Proyecto (1p)

1. Configura un proyecto de React usando Vite.
2. Configura React Router Dom para manejar las rutas de la aplicación.
3. Crea un script para lanzar la base de datos que está en **src/db/db.json**

Rutas:

- **/**: Página de inicio que muestra una lista de encuestas disponibles.
 - **/login**: Ruta para conectar o hacer login a nuestra aplicación.
 - **/edit/:id**: Página para crear editar una encuesta específica.
 - **/stats**: Página para ver las estadísticas de las respuestas asociadas a cada encuesta.
4. Disponemos de una **cabecera** con nuestro nombre y el botón de **login** o **cerrar** sesión en función de si estamos logueados o no, es decir será un botón condicional, que mostrará login si no estamos conectados o cerrar sesión si ya estamos conectados.

Ejercicio 2: Contexto Global de Usuario (1,25p)

1. Implementa un contexto global para manejar la autenticación del usuario. Llama al contexto `UserContextLogin` encargado de mantener el estado del usuario (logueado o no loggeado).
2. La clave por defecto para entrar será `usuario:profesor, password:isaías`
3. Obligatoriamente proporciona `funciones` para iniciar y cerrar sesión.
4. Cuando iniciemos sesión, almacenaremos en el `localStorage` una clave llamada `access_token` que guarde el usuario y la password.
5. Cuando cerremos sesión, eliminaremos el `access_token` del `localStorage`.
6. Sólo si hemos iniciado sesión podremos editar un cuestionario.
7. Si no existe la clave en el `localStorage` al iniciar sesión redireccionaremos a `/login`.

Ejercicio 3: Login (1p)

1. Crea un formulario para realizar el Login con usuario y password. Validará las credenciales a través del contexto. El acceso será como ya te he indicado con

```
usuario:profesor  
password:isaías
```

Guarda dichos valores en **Variables de Entorno** dentro del fichero `.env`

2. **[JAVASCRIPT]** Validar con Javascript que los campos de formulario no estén vacíos y que la longitud del password sea superior a 5 caracteres.

Ejercicio 4: API Local con JSON Server (0,5p)

1. Configura JSON Server para simular una API local.
2. Crea un script que permita levantar archivo `db.json` que contenga la estructura de datos para las encuestas.

El aspecto del json será aproximadamente como se muestra a continuación.

```
{  
  "surveys": [  
    {  
      "id": 1,  
      "title": "Encuesta de Satisfacción",  
      "questions": [  
        {  
          "id": 1,  
          "text": "¿Cómo calificaría nuestro servicio?",  
          "options": ["Excelente", "Bueno", "Regular", "Malo"],  
          .....  
        }  
      ]  
    }  
  ]  
}
```

Ejercicio 5: Página Principal. (2p)

1. Partiendo de la ruta `/` nos encontraremos unas tarjetas con las encuestas que tenemos disponibles en nuestra api.
2. **[JAVASCRIPT]** Crea una función en JavaScript que le pase el id de un cuestionario y obtenga una pregunta al azar de entre todas las preguntas disponibles en ese cuestionario. Debe devolver un objeto con la pregunta y las posibles respuesta a elegir.
3. De cada encuesta necesito saber el ID, el Título de la encuesta y una pregunta al azar obtenida al lanzar la función JavaScript del apartado anterior. Renderizaremos la pregunta y la respuesta en la tarjeta (sin posibilidad de modificar nada.. será una visualización simplemente).
4. Si hemos realizado el Login, cada tarjeta dispondrá del botón de **Eliminar Encuesta** y de **editar la Encuesta**.
 1. El Botón de **Eliminar** encuesta usará Sweet-Alert2 para validar la eliminación. Esto acarreará la eliminación de la encuesta de la API.
 2. El botón de **Editar la Encuesta** nos redireccionará a la ruta `/edit/:id` para realizar la edición de la encuesta con id seleccionado.
5. Adicionalmente cada tarjeta tendrá el botón de **Estadísticas** (estemos logueados o no) que redireccionará a `/stats` donde veremos las estadísticas de **TODAS** las encuestas.

Ejercicio 6: Editar la Encuesta con id seleccionado. (2,5p)

1. Al pulsar en el botón editar del ejercicio anterior, accederemos a la ruta `/edit/:id`.
2. Crearemos un componente que de forma iterativa permitirá mostrar todas las preguntas de esa encuesta y editarlas. Al pulsar en guardar se almacenarán los cambios en la api y se mostrarán en la parte inferior de la zona de edición de las preguntas.

Ejercicio 7: Estadísticas. (1,75p)

1. **[JAVASCRIPT]** Crear una función en javascript que le pasemos el id de una encuesta y obtenga por cada pregunta, la respuesta que ha sido seleccionada más veces. Dicha información se almacenará en una estructura MAP (numero_de_pregunta, texto_respuesta_más_seleccionada, número_veces_seleccionada)
2. Crea un componente que muestre una tarjeta por cada encuesta, donde aparezca el nombre de la encuesta y por cada pregunta, la respuesta más seleccionada y el número de veces que ha sido seleccionada. Utiliza la función Javascript del apartado anterior.

NOTAS:

1. El proyecto ha de ser comprimido y enviado a la tarea asignada.
2. No comprimir la carpeta `node_modules`.
3. Está permitido el uso del material de clase.
4. Esta prueba evalúa TODOS los RA del módulo de DWEC.