Proyecto: tercera iteración

1. Introducción

Este proyecto consiste en la implementación incremental a través de varias iteraciones de una aplicación web. En este documento detallaremos la tercera iteración.

La aplicación consiste en una tienda en la que podemos vender productos. En la primera iteración se desarrolló una interfaz utilizando HTML, CSS y JavaScript. En la segunda se refactorizó el código para incluir React, guardar localmente datos mediante localStorage y aumentar la eficiencia de la aplicación usando Workbox. Partiendo de esta segunda versión, en esta parte crearemos un servidor Express para atender peticiones y usaremos MongoDB como sistema de persistencia.

En esta nueva versión tendremos usuarios que pueden autenticarse en la aplicación, bien como administradores o sin rol. Los datos de estos usuarios se guardarán en Firebase y en MongoDB. Por otro lado, también se guardarán los datos de todos los productos en MongoDB en lugar de tenerlos puestos a mano en el fichero *tienda.js*.

Además, dispondremos de varias secciones en la aplicación para gestionar tanto nuestra cuenta de usuarios como los productos de la tienda (en caso de ser administrador) y un contador de visitas a la página.

Para gestionar las peticiones a la base de datos y las sesiones, se usa un servidor en Express. Este servidor correrá de forma separada a la aplicación web, en el mismo servidor pero en distintos puertos. Para ello, tened en cuenta el CORS.

2. Requisitos tecnológicos

En esta segunda iteración volveremos a utilizar las tecnologías web de la primera iteración (sección 2.1) y la segunda iteración (sección 2.2), e incluiremos nuevas (sección 2.3).

2.1. Tecnologías de la primera iteración

- HTML
- CSS
- JavaScript
- Librerías HTML/CSS/JS: Bootstrap

2.2. Tecnologías de la segunda iteración

- React
- Librerías React: React Drag&Drop
- Workbox (Service Workers)
- localStorage en JavaScript

2.3. Tecnologías adicionales de la tercera iteración

- Express
- MongoDB
- Firebase

3. Requisitos de diseño

Para estos requisitos se ha de tener en cuenta que existen usuarios y, en concreto, hay dos tipos de usuarios: los administradores y los usuarios sin rol.

3.1. Secciones nuevas

Existe una sección nueva para cualquier usuario: “Mi cuenta”. Además, para los usuarios con rol de administrador, se incluyen 2 más: “Añadir un producto” y “Editar/Borrar productos”. En la Figura 1 se muestra un ejemplo del nuevo menú de navegación.

Figura 1. Ejemplo de menú de navegación con las nuevas secciones.

3.2. Autenticación

El <aside> será ahora utilizado para un panel de autenticación tal y como se muestra en la Figura 2. Contendrá un título con un nombre descriptivo y el formulario de doble campo (email y contraseña), además de un botón para autenticarse.

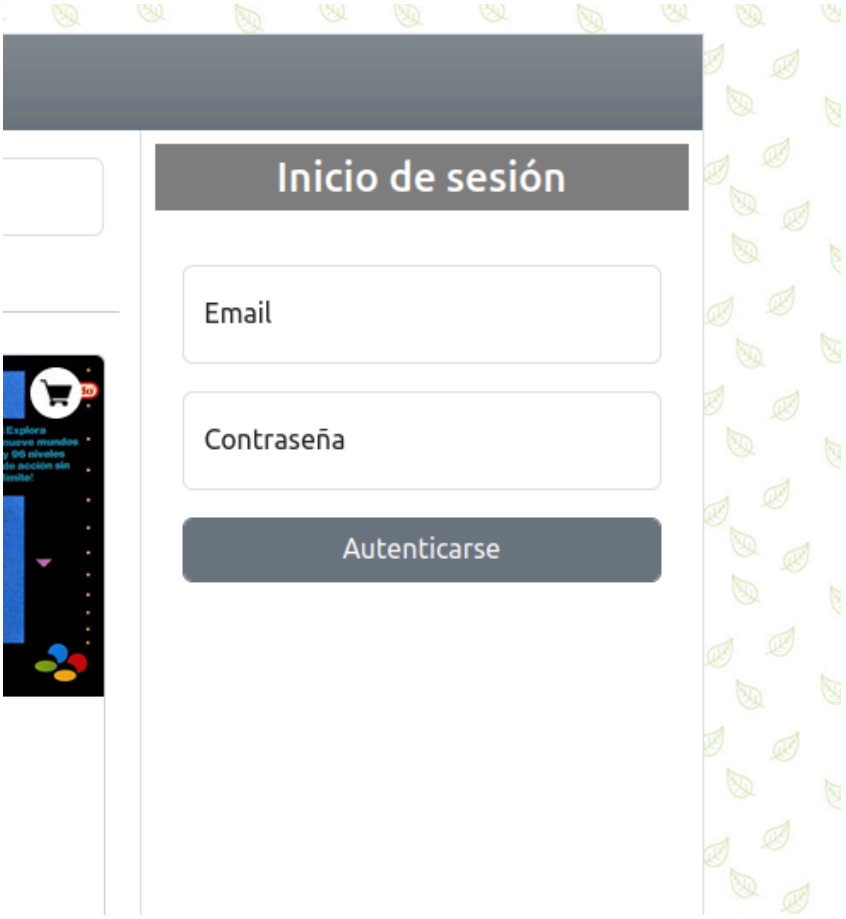

Figura 2. Ejemplo de panel de autenticación de usuarios.

3.3. Panel de usuario

En caso de autenticarnos, veremos en el mismo <aside>, donde antes estaba el panel de autenticación, un panel de usuario. El panel contiene un mensaje de bienvenida y un rectángulo indicando dos datos: el rol (si es que hubiese) y el número de visitas desde el inicio de sesión. También incluye un botón para cerrar la sesión y volver a mostrar el panel de autenticación.

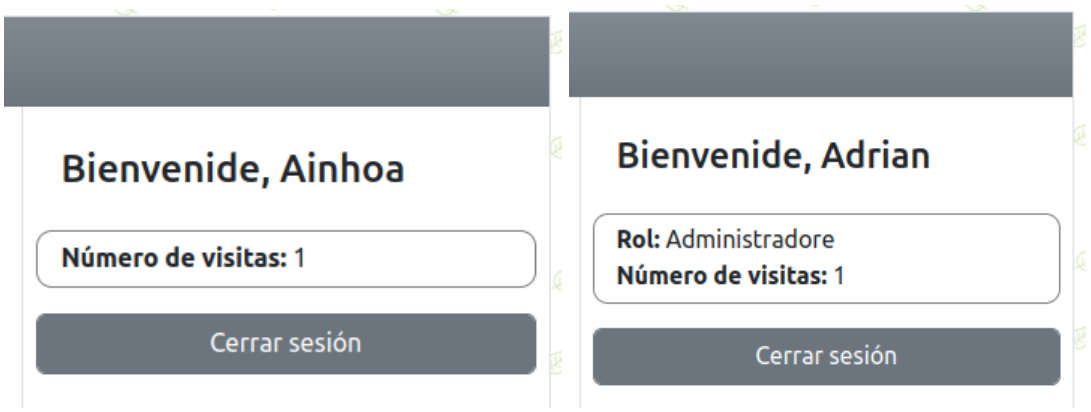

Figura 3. Ejemplo de panel de usuario autenticado. A la izquierda una usuaria sin rol y a la derecha un usuario con rol de administrador.

Si el número de visitas incrementa, se mostrará otro número. El número empieza en 1 cada vez que nos autenticamos, es decir, sólo se guarda y actualiza mientras tengamos la sesión activa. Se debería actualizar cada vez que refresquemos la página estando autenticados.

3.4. Sección “Mi cuenta”

Al hacer click en el menú de navegación en “Mi cuenta” se debería mostrar, en lugar del escaparate, otro componente para poder editar los datos del usuario autenticado.

En concreto, se mostrará como mínimo el nombre del usuario y el email. Este último además tiene que ir siempre deshabilitado y no se podrá modificar. El resto de campos sí que tienen que poder editarse (con la única excepción del rol de usuario, que no se mostrará).

Es necesario incluir el resto de campos de usuario tal y como se pide más adelante.

Por otro lado, todos los campos tienen que tener como valor por defecto los datos del usuario autenticado. Es decir, el campo de nombre, por ejemplo, en lugar de dejarlo vacío, tendría como valor por defecto “Adrián” si ese fuese el nombre del usuario.

Figura 4. Ejemplo de sección "Mi cuenta".

3.5. Sección "Añadir un producto"

Esta sección únicamente se mostrará para usuarios con rol de administrador. Al hacer click en el menú de navegación en "Añadir un producto" se debería mostrar, en lugar del escaparate, otro componente para poder añadir nuevos productos. Se reutilizará el componente ya creado que previamente estaba en el <aside> y que deberá seguir funcionando exactamente igual.

Figura 5. Ejemplo de sección "Añadir un producto".

3.6. Sección "Editar/Borrar productos"

Esta sección únicamente se mostrará para usuarios con rol de administrador. Al hacer click en el menú de navegación en "Editar/Borrar productos" se debería mostrar, en lugar del escaparate, otro componente para poder editar y borrar productos.

Atendiendo a la Figura 6, debería aparecer un componente con una lista mostrando todos los productos y un botón en la parte superior con la opción de borrar todos aquellos elementos seleccionados.

En la lista de productos hay un *checkbox* por producto a la izquierda del todo, pudiendo seleccionar 0 o varios para borrarlos mediante el ya mencionado botón. Junto al *checkbox* tenemos una miniatura de la imagen del producto. A la derecha del todo tenemos un enlace con el texto “Editar”.

En caso de seleccionar al menos 1 producto de la lista y pulsar el botón superior de “Borrar todos los seleccionados”, se eliminará automáticamente ese elemento de la aplicación, no volviendo a mostrarse en esta lista ni en el escaparate y borrándose de la base de datos.

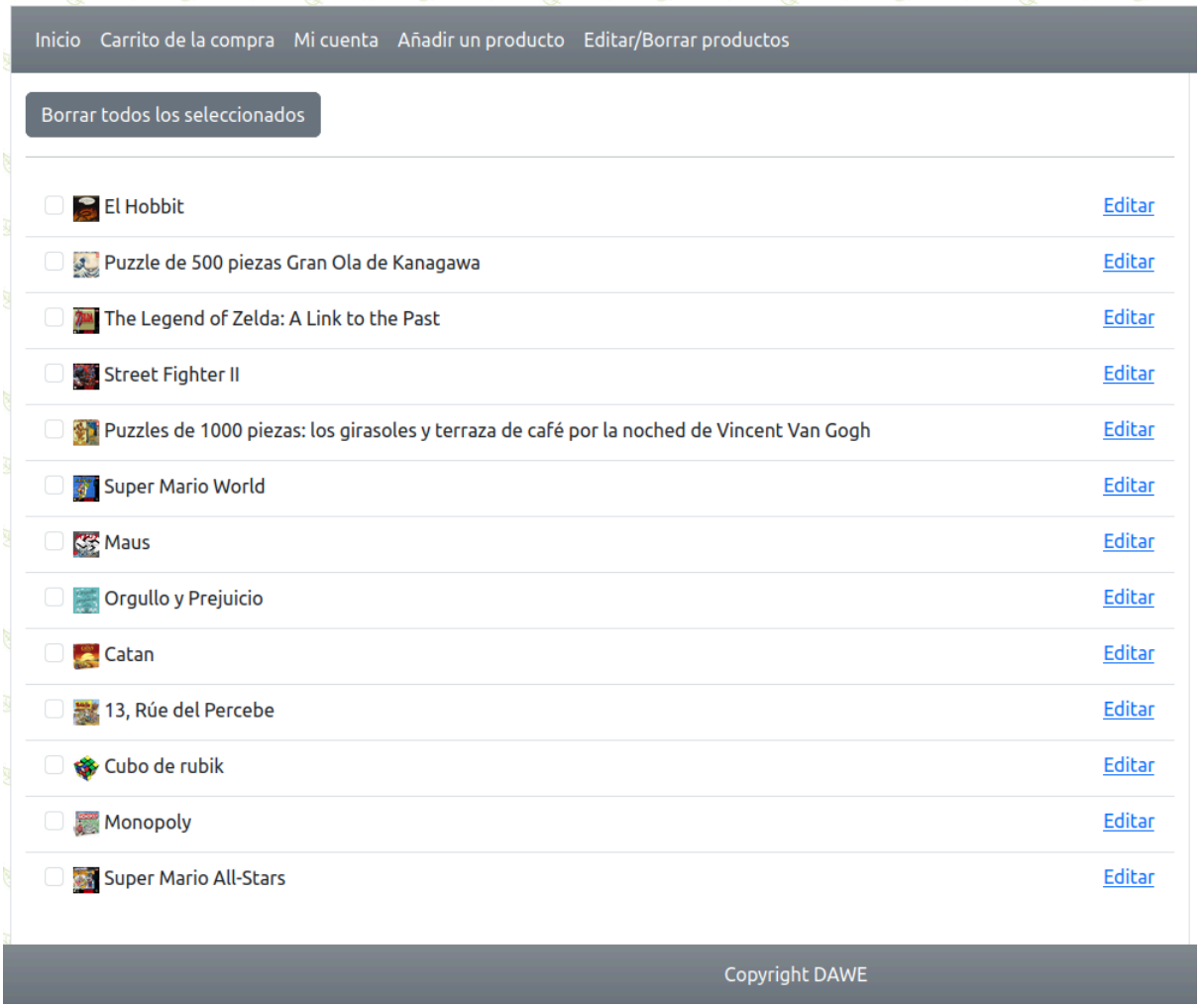

Figura 6. Ejemplo de sección “Editar/Borrar productos”.

En caso de pulsar el botón “Editar” de algún producto, se abrirá un formulario inmediatamente debajo del producto tal y como se muestra en la Figura 7. El enlace de “Editar” pasará a ser “Cerrar” y, en caso de pulsar ahora este enlace, el formulario desaparecerá.

El formulario de edición de productos incluirá todos los campos del producto. El tipo de producto aparecerá, pero estará deshabilitado. En caso de no subir una imagen nueva, se mostrará la anterior. El resto de campos tienen su valor por defecto. Al pulsar el botón de “Guardar cambios” se actualizará en la base de datos el producto y desaparecerá el formulario.

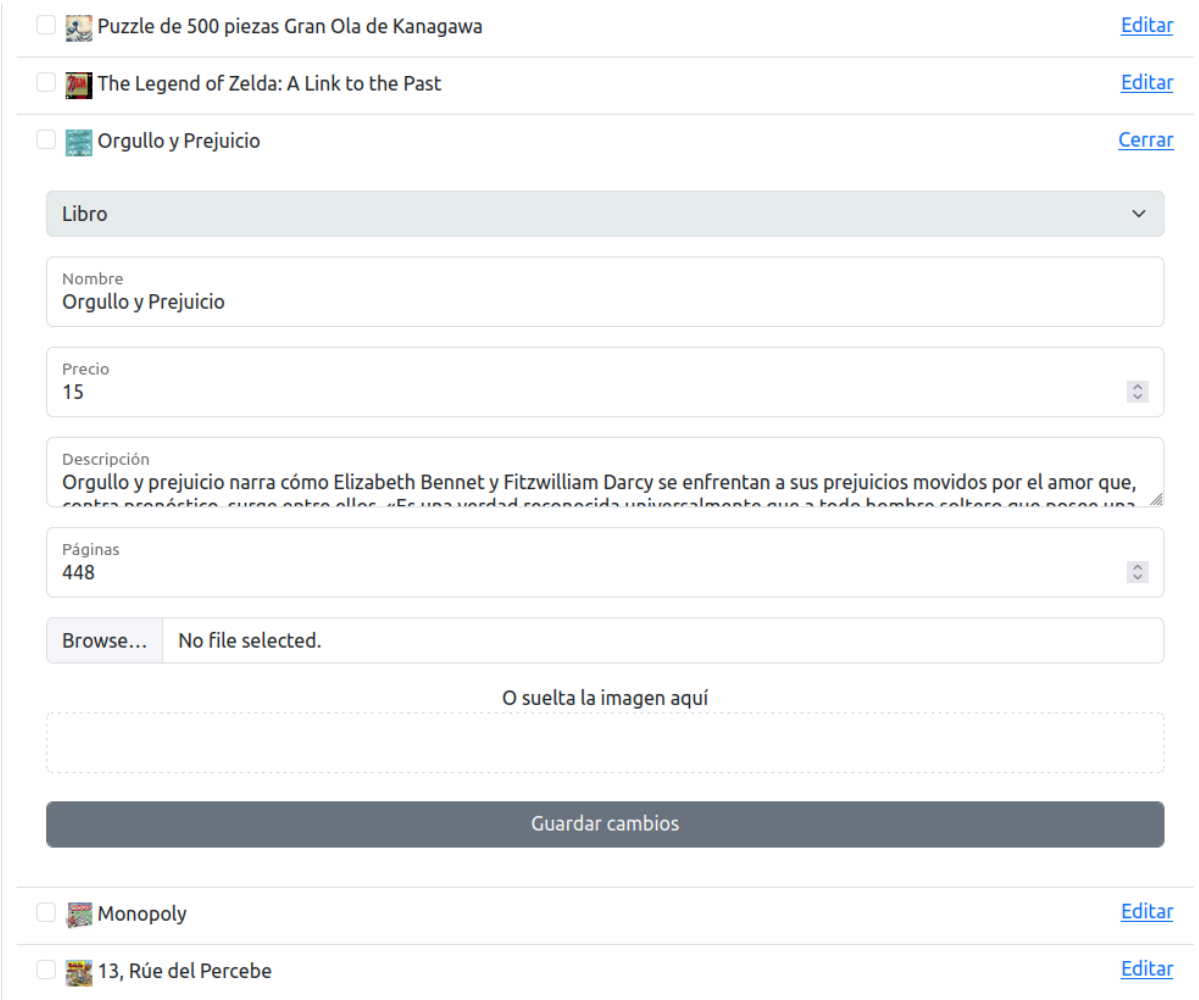

Figura 7. Ejemplo de sección “Editar/Borrar productos” al pulsar el botón “Editar” del producto “Orgullo y Prejuicio”.

4. Requisitos de clases y ficheros

4.1. Estructura del proyecto

El proyecto en sí contendrá dos subproyectos: el servidor y el cliente. El cliente es la aplicación ya diseñada en las dos anteriores iteraciones y debe mantener la estructura mencionada en la segunda iteración.

En cambio, el servidor, es un servidor Express que debe contener la siguiente estructura:

```
package.json
```

```
rutas/
    - usuarios.js
    - productos.js
index.js
```

donde index.js es el servidor Express que incluye manejadores de rutas de dos ficheros: usuarios.js y productos.js. No se incluirán manejadores de rutas en index.js, sólo funciones de middleware.

5. Requisitos funcionales

En los siguientes apartados se describirán los requisitos funcionales necesarios. Una parte del trabajo es saber qué tipo de peticiones hacer al servidor Express en cada caso (es decir, GET, POST, PUT y DELETE). Además, se deben de crear las variables de estado necesarias en cada componente para el correcto funcionamiento de la aplicación.

5.1. Base de datos MongoDB

Se creará en MongoDB una nueva base de datos llamada tienda que incluirá dos colecciones/tablas: usuarios y productos.

La tabla de usuarios incluirá, como mínimo, los campos nombre y email. Se deben incluir, además de estos, 3 campos más a elección del grupo. Además, es necesario incluir un campo extra que no será editable: el rol del usuario. Sólo se mostrará en el panel de usuario autenticado (ver Sección 3.3). Dado que no se ha definido un registro, se introducirán los usuarios a mano.

La tabla de productos incluirá todos los campos utilizados hasta la fecha: tipo de producto, nombre, precio, descripción, el campo extra de cada producto y la imagen (ruta).

Ambas tablas pueden ser modificadas mediante la aplicación tal y como se explicará más adelante.

5.2. Sesión en Express

Se utilizará el paquete express-session de Express para gestionar sesiones de usuario. En concreto, se almacenarán las sesiones de usuario dentro de la base de datos tienda que hemos creado en la Sección 5.1.

Será necesario usar el email como prueba de que la sesión está iniciada y se almacenará un contador de visitas en la sesión, inicializado a 1. Cada vez que el usuario actualiza la página, el contador tiene que incrementarse en 1 y reflejarse en el panel de usuario autenticado (ver Sección 3.3).

5.3. Autenticación

Tal y como se describe en la sección 3.3, hay un panel de autenticación de usuario en la página. En concreto, utiliza la herramienta de autenticación de Firebase para autenticar al

usuario. Los usuarios no se pueden registrar desde la aplicación, es necesario registrarlos a mano. Sólo se usará el email y contraseña para el registro.

En la aplicación, una vez pulsemos el botón de autenticación, tenemos que autenticarnos en Firebase. Si los datos son correctos, se cambiará el panel de autenticación por el panel de bienvenida al usuario.

Si estamos sin conexión, el formulario no debe funcionar (estará deshabilitado).

5.4. Contador de visitas

En la sección 3.3 se describe el panel de usuario, que incluye un contador de visitas. Este contador es una variable de sesión y debe iniciarse a 1 cuando iniciamos sesión. Mientras la sesión se mantenga activa, por cada actualización de la página, el número de visitas aumentará (ver Figura 8).

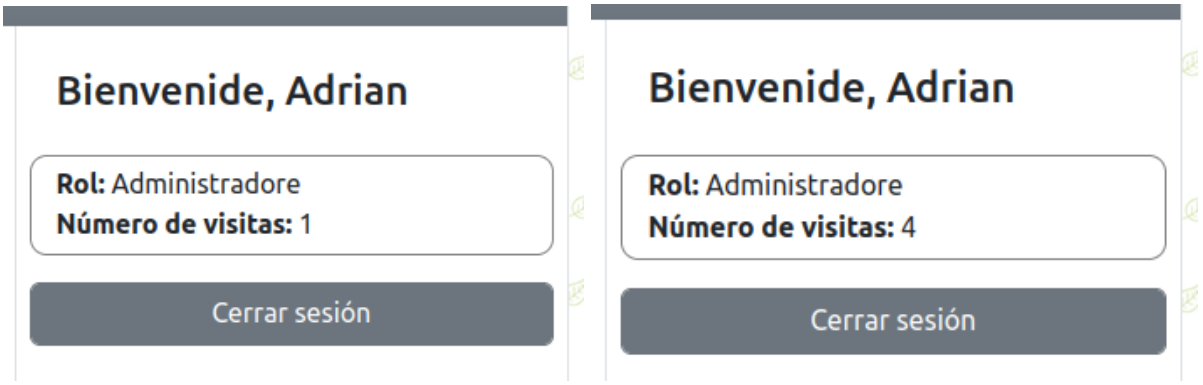

Figura 8. Ejemplo de panel de usuario autenticado tras refrescar la página varias veces.

5.5. Formulario de edición de datos de usuario

En la Sección 3.4 se describe la sección de “Mi cuenta”, donde podemos editar los datos del usuario. Ya se ha mencionado en la Sección 5.1 que la tabla de usuarios contiene al menos 6 campos: nombre, email, rol y 3 campos extra. En este formulario deben aparecer nombre, email (no editable) y los 3 campos extra. Estos últimos deberían estar vacíos cuando creamos la cuenta.

El campo nombre NO se puede dejar vacío. En caso de que esté vacío no se realizará la modificación y aparecerá un mensaje de error en rojo (debajo del formulario), que desaparecerá al de poco tiempo.

Si estamos sin conexión, el formulario no debe funcionar (estará deshabilitado).

5.6. Formulario de subida de productos

En la Sección 3.5 se describe la sección para añadir nuevos productos. Se reutiliza el componente ya creado previamente. En este caso, aunque el funcionamiento sea el mismo, una vez un producto se crea correctamente, se añade a la base de datos MongoDB.

Si estamos sin conexión, el formulario no debe funcionar (estará deshabilitado).

5.7. Formulario de edición y borrado de productos

En la Sección 3.6 se describe la sección para editar y borrar productos. Al pulsar el botón de borrado de productos, se envía una petición para borrar una lista de productos. En caso de editar un producto, al enviar el formulario se envía una petición distinta, para actualizar unos campos concretos de un producto concreto.

Si estamos sin conexión, no se permitirá editar productos (se debe modificar el enlace para que quede claro que no se puede usar) y el botón para borrar productos debe estar deshabilitado.

6. Entrega y evaluación

La entrega y evaluación sigue siendo la misma que la de la iteración 1. Id a dicho documento para consultar cómo funciona.