Ejercicio de frontend

- Lea atentamente las instrucciones y especificaciones antes de comenzar el ejercicio.
- Cada epígrafe se relaciona con el anterior desde el punto de vista técnico pero se valorarán por separado.
- Las valoraciones las puede encontrar en los criterios de valoración.

Especificaciones del ejercicio

1. Se pide la creación de un proyecto de vue

a. Tiene que emplear vue 2
b. Ha de tener instalado vuetify (versión más reciente - 2.4.9)
c. Debe tener una store con vuex.

2. Layout

a. Estamos ante una SPA que puede, opcionalmente, emplear el enrutamiento a través del router de vue.
b. Todo el layout y la aplicación web ha de ser responsive y se adaptará adecuadamente a entornos:
    i. De grandes pantallas
    ii. Pantallas de PC ordinarias
    iii. Tablets
    iv. Móviles
c. Los estilos y apariencia (UI/Ux) es totalmente libre y se dejan al arbitrio del programador mientras cumpla las especificaciones.

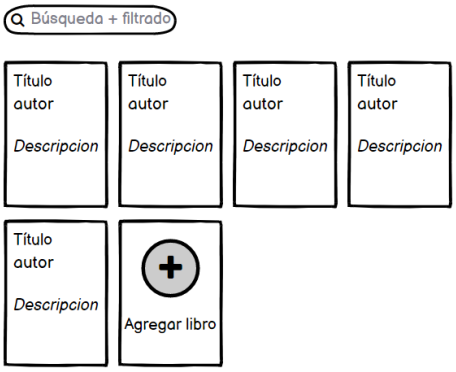
3. Autenticación

a. Los usuarios se loguean mediante usuario/contraseña.
b. Hacer un mock del cliente de una api para conectarse con las credenciales pasadas por el usuario
c. El mock de api devuelve un token de acceso que se almacenará en local storage en el navegador del cliente.
d. Si el token no está presente debe aparecer un diálogo de pantalla completa que obligue a loguear.
e. Estos logins tienen que ser válidos (cómo mínimo)
    i. user / mipass
    ii. admin / admin
    iii. user2 / mipass2

4. Backend

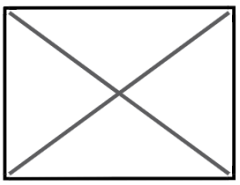
a. La aplicación es un UI para la gestión de una librería online.
b. El backend se encuentra en esta url: https://demo.api-platform.com/
c. El token de autenticación generado en el epígrafe anterior no tiene que enviarse.
d. Hay que generar un cliente para conectarse al backend. Emplear los medios que se consideren adecuados.
e. La gestión de los datos debe realizarse, en la medida de lo posible, a través de la store de vuex instalada en el epígrafe 1.

5. Funcionalidad de bookshelf



a. Los libros deben mostrarse en tarjetas en formato bookshelf (ver figura)
b. Debe existir un buscador que permita filtrar por título, autor o palabra encontrada en la descripción.
c. Debe existir un icono para crear un nuevo libro. Debe llevar a la funcionalidad de alta de libro (consultar epígrafe 7)
d. En caso de hacer click sobre un libro debería cargarse la funcionalidad de información ampliada de dicho libro (consultar epígrafe 6)

6. Funcionalidad de visionado de libro



Titulo
Autor
Descripción

Reseñas (1..N)

- a. Muestra toda la información de un libro
    - i. Titulo
    - ii. Autor
    - iii. Imagen (puede ser aleatoria o una igual para todos)
    - iv. Descripción
    - v. Reseñas (ordenadas por tiempo de realización descendente)
- b. El autor es un botón que permite abrir el bookshelf (ver epígrafe 5) filtrados por los del autor concreto.

7. Funcionalidad de alta de libro

- a. Debe crearse un formulario que permita introducir y (enviar vía api cuando se acepte)
    - i. Título
    - ii. Autor
    - iii. Descripción
    - iv. Isbn
    - v. fecha de publicación
- b. El formulario debe validar
    - i. El título no existe ya
    - ii. El isbn es correcto
    - iii. El autor no está vacío
    - iv. La descripción no está vacía
- c. El formulario, en caso de estar medianamente cumplimentado, debe resistir reinicios o refrescos de la pantalla (se puede almacenar la información en el localStorage)

8. Organización del código y repositorio

- a. El código debe estar en un repositorio de git.
- b. Se coloca en un repositorio privado de github

9. Docker y linter
    a. Se proporciona una imagen de Docker para correr el proyecto y hacer el build para producción.
    b. Se proporciona una documentación de como:
        i. Construir la imagen
        ii. Arrancar en modo desarrollo
        iii. Compilar para producción
    c. Hay un linter instalado que controla el estilo del código.

10. Valoración general de estilos y UI
    a. Se valorará el correcto uso de los componentes de vuetify.
    b. Se valorará el estilo general y el acabado de la aplicación.

Criterios de valoración

1. Proyecto que cumpla las especificaciones básicas establecidas **(1 punto)**
2. La aplicación web **(1 punto)**
    a. Es SPA (0.5 puntos)
    b. Se adapta adecuadamente a los distintos entornos (0.5 puntos)
3. La autenticación en la aplicación **(1 punto)**
    a. controla correctamente el ingreso del usuario (0.5 puntos)
    b. el mock de api funciona y devuelve un token (0.25 puntos)
    c. los logins exigidos están disponibles (0.25 puntos)
4. La gestión del backend (**1 punto**)
    a. Gestiona adecuadamente los peticiones a la API. (0.5 puntos)
    b. Modela los datos en la store interna de vuex. (0.5 puntos)
5. Funcionalidad de bookshelf (**1 punto**)
    a. Los libros se exponen correctamente (0.5 puntos)
    b. El filtrado funciona de forma adecuada (0.25 puntos)
    c. Los links a libros y a creacion de libro funcionan. (0.25 puntos)
6. Funcionalidad de visionado del libro **(1 punto)**
    a. El libro se muestra correctamente. (0.5 puntos)
    b. Se ven bien las reseñas. (0.25 puntos)
    c. El links a publicaciones del autor funciona. (0.25 puntos)
7. Funcionalidad de creación de libro **(1 punto)**
    a. El formulario funciona y envía los datos correctamente. (0.5 puntos)
    b. Realiza las validaciones solicitadas. (0.25 puntos)
    c. El formulario está dotado de persistencia local. (0.25 puntos)
8. El repositorio de código **(1 punto)**
    a. Está en git (0.75 puntos)
    b. Se distribuye a través de un repositorio de github privado. (0.25 puntos)
9. Docker y linter **(1 punto)**

    a. Hay una imagen de Docker para correr el proyecto (y build del mismo). (0.5 puntos)
    b. La documentación de la operativa es correcta y suficiente. (0.25 puntos)
    c. Hay un linter instalado y se corrige el código. (0.25 puntos)
10. Valoración general de estilos y UI. (**1 punto**)
    a. A valorar por parte de la empresa.