

Ejercicios para resolver con componentes SFC, Vue CLI y Vue-Router

1. Crea una página web dinámica(haciendo uso de vue-router) de un periódico, que:
 - a. Tenga una cabecera que ocupe todo el ancho de la página. En ella se verá el nombre del periódico centrado.
 - b. Tenga un menú principal en el lateral izquierdo de la página. Dicho menú tendrá tres opciones:
 - i. Inicio
 - ii. Artículos
 - iii. Acerca de
 - c. En el resto de la página será donde se carguen las diferentes vistas con información de acuerdo a las siguientes indicaciones.
 - d. Cuando se pulse la opción Inicio, se mostrará un mensaje de bienvenida al usuario.
 - e. Cuando se pulse la opción Acerca de, se mostrará información de los trabajadores del periódico, la ubicación de la sede y teléfono e email de contacto.
 - f. Por último, cuando se pulse sobre Artículos, se mostrará una lista con el nombre de todos los artículos disponibles. Esta lista se recuperará como sigue:
 - i. Al cargar la vista(VER punto h), haciendo uso de la API JSONPlaceholder se solicitarán a la misma todos los artículos que tiene almacenados en su sección posts.
 - ii. Estos artículos se almacenarán en un array. Cuando se acceda a la sección de artículos se mostrará un enlace para cada uno de los artículos existentes.
NOTA: Estos enlaces se harán haciendo uso de la etiqueta router-link. El valor de su atributo to será necesario bindearlo ya que no se le va a pasar la ruta como tal si no que ésta se formará a partir de la variable que contenga el id del artículo en cuestión.
 - g. Cuando se pulse sobre el enlace de cualquier artículo, se mostrará en la zona de vistas una **vista** con el texto del artículo en cuestión. Para hacer esta parte no se creará una ruta por cada artículo sino que se tratará de una ruta con parámetros en la que se pasará el id del artículo. Y, cuando se cargue esta vista, se hará otra petición a la misma API para recuperar el artículo en cuestión.
 - h. **IMPORTANTE:** La petición se hará una vez se haya creado el componente. Esto es, se hará una vez el componente se haya cargado. Para esto se hará uso de la función hook adecuada.

2. Partiendo de la solución implementada para el ejercicio anterior:
- a. En la sección lateral de navegación, debajo de todos los enlaces (será un componente nuevo) se incluirá un pequeño formulario compuesto por un campo de entrada numérico y un botón:
 - i. El botón, de inicio, estará desactivado.
 - ii. Cuando el usuario introduzca un valor en el campo se habilitará automáticamente el botón.
 - iii. Cuando el usuario pulse el botón se cargará, en la zona central, el artículo correspondiente al número introducido por el usuario.
 - b. Cuando se cargue la página de inicio, se pedirá al usuario por pantalla que introduzca su edad:
 - i. Este dato se almacenará y, dependiendo del valor que el usuario introduzca, la página se comportará como sigue:
 - 1. Si es menor de 18 años, no podrá acceder a ver el contenido de ningún artículo.
 - 2. Si es menor de 15, no podrá siquiera ver el listado de artículos disponibles.
 - c. Además, se incluirá una nueva vista con un formulario para quejas, dudas, sugerencias, etc. que incluirá los siguientes campos:
 - i. Nombre.
 - ii. Apellidos.
 - iii. Edad.
 - iv. Correo electrónico.
 - v. Sugerencia.
 - vi. Botón de envío.
 - d. Cuando se envíe una nueva queja, se almacenará en un array con todos los datos introducidos.
 - e. El texto de cada queja, junto con el nombre y edad de quien la haya mandado se mostrará en una tabla. Esta tabla se verá cuando el usuario pulse sobre un nuevo elemento de la barra de navegación: Quejas

3. En este ejercicio se va a modificar y completar el ejercicio 2. Haz de uso de todos los recursos conocidos: Vue-router, axios, SFC, etc.:
- a. Crea un fichero json que incluya artículos y quejas:
 - i. De un artículo se almacenará un id(valor generado por el servidor), un título y un contenido.
 - ii. De cada queja se almacenará toda la información incluida en el formulario del ejercicio 2 mas un id(valor generado por el servidor).
 - b. Dicho fichero json lo servirá un json-server arrancado en tu equipo.
 - c. La página que muestra los títulos de todos los artículos, ahora solicitará dichos artículos a tu servidor local. La forma de mostrarlos será la misma, cambiarán los nombres de las propiedades del objeto.
 - d. La página que muestra un artículo concreto:
 - i. recogerá los datos del mismo del servidor local.
 - ii. Incorporará, tras el artículo, un botón que permitirá eliminar el artículo. Este botón se creará en un componente aparte y en dicho componente se realizará la petición correspondiente para eliminar del servidor el artículo en cuestión.
 - iii. Comprueba que, tras eliminar un artículo, cuando se carga la página de todos los enlaces a los artículos el eliminado no aparece más.
 - e. Cuando el usuario introduzca una nueva queja, ésta se enviará al servidor para almacenarla en lugar de almacenarla en un array.
 - f. Cuando se visualice la tabla de quejas, éstas se pedirán al servidor.