**PROYECTO**

**KIT DE CARGA**

**DOCUMENTACION**

**ResponsiveAppBar**

* Se define una constante **pages** que es un array de strings, cada uno representando el nombre de una página en la aplicación.
* Se define el componente **ResponsiveAppBar**. Este componente usa el hook **React.useState** para manejar el estado del menú desplegable de la barra de navegación. **anchorElNav** es el elemento DOM donde se ancla el menú, y **setAnchorElNav** es la función que se usa para actualizar este estado.
* **handleOpenNavMenu** y **handleCloseNavMenu** son funciones que se utilizan para manejar el abrir y cerrar del menú desplegable, respectivamente.
* En el **return** del componente, se define la interfaz de usuario del componente **ResponsiveAppBar**. Esta consiste en un componente **AppBar** que contiene un **Toolbar**.
* Dentro de este **Toolbar**, hay varios componentes, incluyendo un logo (una imagen importada), un componente **Typography** que muestra el texto "KIT DE CARGA", un componente **IconButton** que se utiliza para abrir el menú desplegable en pantallas pequeñas (dispositivos móviles), y finalmente, tres componentes **Button** que se utilizan para la navegación entre las diferentes páginas de la aplicación.

**Props**

* El componente no recibe ninguna prop.

**Estados**

* **anchorElNav** - Representa el elemento DOM que se utiliza para anclar el menú desplegable.

**Funciones**

* **handleOpenNavMenu(event)** - Función que abre el menú desplegable y ancla el menú en el elemento DOM que disparó el evento.
* **handleCloseNavMenu()** - Función que cierra el menú desplegable.

**Salida**

El componente retorna una barra de navegación que se adapta según el tamaño de la pantalla. En pantallas pequeñas, la barra de navegación muestra un menú desplegable al presionar el ícono del menú. En pantallas grandes, se muestran botones para navegar entre las páginas.

**Footer**

Este componente de React representa un pie de página con una imagen y un texto.

**Props**

* No aplica.

**Estados**

* No aplica.

**Funciones**

* No aplica.

**Salida**

El componente retorna un elemento de pie de página que incluye una imagen (logo) y un texto ("Desarrollado por equipo de ventas. Sonda").

**CustomizedSnackbars**

* Se importan varios componentes de Material-UI, incluyendo **Stack**, **Button**, **Snackbar**, y **Alert**.
* Se crea un componente personalizado **Alert** utilizando la función **forwardRef** de React y el componente **MuiAlert** de Material-UI. Este componente personalizado se utiliza luego en el componente principal.
* En el componente **CustomizedSnackbars**, se define un estado **open** utilizando el hook **useState** de React, que determina si el Snackbar está abierto o cerrado.
* El hook **useEffect** se utiliza para abrir el Snackbar automáticamente cuando cambia el prop **alerta**.
* Las funciones **handleClick** y **handleClose** se utilizan para abrir y cerrar el Snackbar, respectivamente.
* En el **return** del componente, se define la interfaz de usuario del componente **CustomizedSnackbars**. Consiste en un **Snackbar** que muestra un **Alert** con un mensaje dependiendo del valor de **tipoAlerta**. El Snackbar se cierra automáticamente después de 2 segundos.

**Props**

* **tipoAlerta** - String que define el tipo de alerta que se mostrará. Puede ser 'warning', 'success' o 'info'.
* **alerta** - Booleano que, cuando cambia, provoca que el Snackbar se abra automáticamente.

**Estados**

* **open** - Booleano que representa si el Snackbar está abierto o cerrado.

**Funciones**

* **handleClick()** - Función que abre el Snackbar.
* **handleClose(event, reason)** - Función que cierra el Snackbar, a menos que se haya hecho clic fuera del Snackbar (clickaway).

**Salida**

El componente retorna un Snackbar que muestra un mensaje de alerta. El tipo de alerta y el mensaje que se muestra dependen del valor de **tipoAlerta**. El Snackbar se cierra automáticamente después de 2 segundos.

**ArmarKit**

Este componente de React permite al usuario seleccionar una serie de archivos, organizarlos en una estructura de carpetas y descargarlos en un archivo .zip.

**Props**

* No aplica.

**Estados**:

* **selectedFiles**: Un objeto que contiene los archivos seleccionados por el usuario.
* **countAlert**: Un número que indica cuántas alertas han sido disparadas.
* **alerta**: Un booleano que, cuando cambia, activa el snackbar de alerta.
* **tipoAlertaMsj**: Un string que define el mensaje que se mostrará en la alerta.

**Funciones**:

* **handleFileSelect**(folder): Función que maneja la selección de archivos, recibiendo como argumento la carpeta en la que se deben almacenar.
* **readFileAsBlob**(file): Función que lee un archivo como blob, recibiendo el archivo como argumento.
* **createAndDownloadFolder**(nombreCarpetaPrincipal): Función que crea y descarga la estructura de carpetas en un archivo .zip, recibiendo el nombre de la carpeta principal como argumento.
* **onSubmit**(data): Función que maneja la acción de envío del formulario.

**Salida**

El componente retorna una interfaz de usuario que permite al usuario seleccionar una serie de archivos, organizarlos en una estructura de carpetas y descargarlos en un archivo .zip. Además, muestra alertas informativas durante el proceso.

**Home**

Este componente de React representa una página de configuración con varios formularios y botones.

**Props**

* No aplica.

**Estados**:

* **checked1, checked2, checked3**: Estos estados controlan la selección de varias casillas de verificación.
* **age**: Este estado controla la selección del campo de selección (select field).

**Funciones**

* **handleChange1\_1, handleChange1\_2, handleChange1\_3, handleChange3\_1, handleChangeParent, handleChangeChild2**: Estas funciones manejan los cambios de estado al interactuar con las casillas de verificación.
* **handleChange**: Esta función maneja el cambio de estado al seleccionar una opción en el campo de selección (select field).

**Salida**

El componente retorna una interfaz de usuario con formularios y botones. Los formularios permiten al usuario interactuar con casillas de verificación y campos de selección, y los botones permiten al usuario realizar acciones como eliminar, crear o guardar.

**Configuraciones**

Este componente de React representa una página de configuración que permite a los usuarios interactuar con casillas de verificación y un campo de selección, y realizar acciones como eliminar, crear o guardar.

**Props**

* Este componente no acepta ninguna prop.

**Estados**

* **checked1, checked2, checked3**: Controlan la selección de varias casillas de verificación.
* **age**: Controla la selección del campo de selección (Select field).

**Funciones**

* **handleChange1\_1, handleChange1\_2, handleChange1\_3, handleChange3\_1, handleChangeParent, handleChangeChild2**: Estas funciones manejan los cambios de estado al interactuar con las casillas de verificación.
* **handleChange**: Esta función maneja el cambio de estado al seleccionar una opción en el campo de selección (Select field).

**Salida**

El componente retorna una interfaz de usuario con varios elementos de formulario y botones. Los formularios permiten al usuario interactuar con casillas de verificación y campos de selección, y los botones permiten al usuario realizar acciones como eliminar, crear o guardar. Todos los elementos interactivos están actualmente desactivados (usando la propiedad disabled), por lo que es necesario eliminar la propiedad disabled para permitir la interacción del usuario.

**App**

Este componente de React representa la estructura principal de la aplicación, estableciendo el enrutamiento y organizando otros componentes como AppBar, Home, Configuracion, ArmarKit y Footer.

**Props**

* Este componente no acepta ninguna prop.

**Estados**

* Este componente no maneja ningún estado.

**Funciones**

* Este componente no tiene ninguna función propia.

**Salida**

El componente retorna una interfaz de usuario con una barra de navegación en la parte superior (AppBar), un contenedor principal que incluye diferentes componentes basados en la ruta actual (Home, Configuracion, ArmarKit) y un pie de página en la parte inferior (Footer). Las rutas y los componentes correspondientes se definen utilizando el componente Routes y Route de React Router Dom.

**Dependencias**

* Material-UI: Se utiliza para los componentes de la interfaz de usuario.
* React Router DOM: Se utiliza para la navegación entre las páginas de la aplicación.
* JSZip: Se utiliza para crear la estructura de carpetas en un archivo .zip.
* File-Saver: Se utiliza para guardar el archivo .zip en el dispositivo del usuario.
* React Hook Form: Se utiliza para manejar el formulario y su validación.

**ENLACES DE INTERES**

* <https://kit-carga-sonda.netlify.app/>
* <https://app.netlify.com/sites/kit-carga-sonda/overview>
* <https://github.com/OwlDeev/kit_carga>