

ACTIVIDAD 2

APLICACIÓN CRUD: GESTIÓN DE USUARIOS

En esta actividad crearemos una aplicación para gestionar usuarios usando una API externa, que permitirá mostrar un listado, crear nuevos usuarios, así como editar y borrar los existentes.

Ten en cuenta que los 12 primeros registros no pueden ser borrados ni actualizados. Tus pruebas se realizarán a partir del registro número 13.

El formato de la aplicación será **libre**, pero será necesario usar **componentes mui**.

PRIMERA PARTE: LISTAR USUARIOS DE UNA API

La API que usaremos es <https://melvecode.com/>, que nos devolverá información de este tipo:

```
{
  "id": 1,
  "fname": "Karn",
  "lname": "Yong",
  "username": "karn.yong@melvecode.com",
  "avatar": "https://www.melvecode.com/users/1.png"
}
```

Este array servirá para inicializar una variable de estado **“users”**, necesaria para realizar el resto de los ejercicios.

En este apartado será necesario crear un componente **UserList**, que usará una tabla para mostrar la siguiente información de los clientes.

Aplicación CRUD con API externa

ID	Imagen	Nombre	Apellido	Nombre de usuario
1		Karn	Yong	karn.yong@melvecode.com
2		Ivy	Cal	ivy.cal@melvecode.com
3		Walter	Beau	walter.beau@melvecode.com
4		Gayla	Bertrand	gayla.bertrand@melvecode.com
5		Benjamin	Chaz	benjamin.chaz@melvecode.com
6		Delia	Robin	delia.robin@melvecode.com
7		Walter	Beau	walter.beau@melvecode.com

La barra de navegación usada en el ejemplo es [AppBar de mui](#). Para realizar la tabla se han usado los componentes [Table](#), [TableBody](#), [TableCell](#), [TableContainer](#), [TableHead](#), [TableRow](#) de [mui](#).

SEGUNDA PARTE: CREAR UN USUARIO MEDIANTE UN FORMULARIO QUE SE ENCUENTRA EN UNA RUTA DIFERENTE

Para añadir un usuario, añadiremos un botón en la interfaz principal, sobre la tabla de usuarios que permitirá movernos a una ruta diferente, que mostrará el componente **UserCreate**, que contendrá el formulario para añadir un usuario. Será necesario pasarle (con lifting o con contextos) la función de añadir usuario que básicamente modificará la variable de estado users, añadiendo el nuevo elemento. Una vez añadido volverá a la pantalla inicial con el listado ya actualizado.

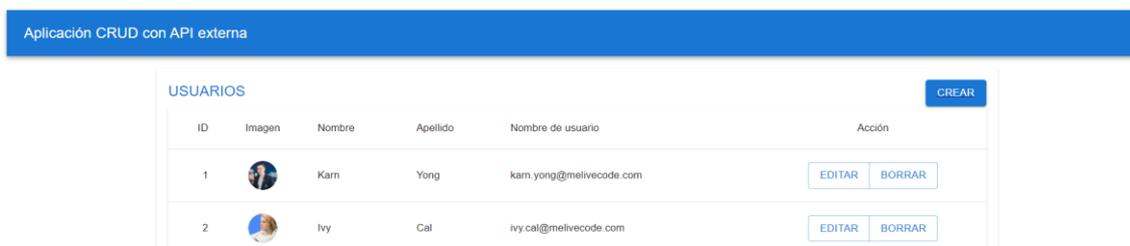
Datos del nuevo usuario:

<input type="text" value="Nombre *"/>	<input type="text" value="Apellido *"/>
<input type="text" value="Nombre de usuario *"/>	
<input type="text" value="Email *"/>	
<input type="text" value="Avatar *"/>	

Para los estilos en este formulario se ha usado *Container*, *Grid*, *TextField* y *Button*.

TERCERA PARTE: REALIZAR LA ACTUALIZACIÓN Y EL BORRADO

A continuación, nos falta por implementar los métodos de actualización y borrado. Será necesario añadir una nueva columna en nuestra tabla de usuarios para estas acciones:



- En el caso de la **edición** será necesario pasarle en la ruta el id, y acceder al mismo mediante el hook **useParams**. Este id será utilizado por el formulario para rellenar los datos del usuario a modificar. Recuerda usar **método PUT** en el fetch.
- Para el **borrado** la URL es <https://www.melivecode.com/users/delete>, y es necesario pasarle en el fetch el id del usuario que se quiere borrar. Una vez realizado el borrado y si todo ha ido bien llamaremos el método que recarga los usuarios (ya implementado al hacer el listado).

RETO OPCIONAL

Realiza el mismo ejercicio pero en lugar de utilizar rutas, usa ventanas modales para la creación y edición de usuarios.