

## ACTIVIDAD 1

### APLICACIÓN CRUD: GESTIÓN DE USUARIOS

#### PREPARACIÓN PREVIA

Tras crear el proyecto, será necesario instalar las siguientes dependencias, ya que trabajaremos con mui, y con rutas.

- `npm install @mui/material @emotion/react @emotion/styled`
- `npm install @mui/icons-material`
- `npm install @fontsource/roboto`
- `npm install react-router-dom`

#### PRIMERA PARTE: APLICACIÓN QUE SÓLO MUESTRA DATOS DEFINIDOS EN EL CLIENTE

En esta primera versión crearemos una aplicación que constará de una barra de navegación (de formato libre, puedes echarle un vistazo a [AppBar de mui](#)) y un componente llamado **UserList**, que usará una tabla para mostrar información de unos clientes que tendremos almacenados en un array de objetos literales de este tipo (se proporciona archivo con los datos):








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

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

Para realizar la tabla puedes usar los componentes [Table](#), [TableBody](#), [TableCell](#), [TableContainer](#), [TableHead](#), [TableRow](#) de [mui](#).

Aplicación CRUD con API externa

#### USUARIOS

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

## SEGUNDA PARTE: CREAR DATOS DEFINIDOS EN EL CLIENTE

Para añadir un usuario, añadiremos un botón en la interfaz principal, sobre la tabla de usuarios y crearemos un componente específico **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:

CREAR USUARIO

```
<Container maxWidth="xs" sx={{ marginTop: '8',
  display: 'flex',
  flexDirection: 'column',
  alignItems: 'center'}}>
  <Typography sx={{p:5}} component="h1" variant="h5">
    Datos del nuevo usuario:
  </Typography>
  <Form sx={{ width: '100%',marginTop: '3',}}
    onSubmit={handleSubmit}>
    <Grid container spacing={2}>
      <Grid item xs={12} sm={6}>
        <TextField
          name="firstName"
          variant="outlined"
          required
          label="Nombre"
          onChange={(e)=>setFname(e.target.value)}
          autoFocus/>
      </Grid>
    </Grid>
    <Button
      type="submit"
      fullWidth
      variant="contained"
      color="primary"
      className='submit'>
      Crear Usuario
    </Button>
  </Form>
</Container>
```

Para trabajar con varias pantallas será necesario incorporar rutas en este punto.



Nuestra App podría ser algo así:

```
<Router>
  <Routes>
    <Route path="/" element={<UserList/>} />
    <Route path="/create" element={<UserCreate/>} />
  </Routes>
</Router>
```

## TERCERA PARTE: LA APLICACIÓN HACE LAS MISMAS OPERACIONES, PERO USANDO UNA API (LOS DATOS PASAN A ESTAR EN EL SERVIDOR)

Vamos a actualizar nuestros componentes para que los datos residan en un servidor. La API que usaremos es <https://melvecode.com/>. Ten en cuenta que los 12 primeros registros no pueden ser borrados ni actualizados. Tus pruebas se realizarán a partir del registro número 13.

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:

Aplicación CRUD con API externa						
USUARIOS						CREAR
ID	Imagen	Nombre	Apellido	Nombre de usuario	Acción	
1		Kam	Yong	kam.yong@melivecode.com	EDITAR	BORRAR
2		Ivy	Cal	ivy.cal@melivecode.com	EDITAR	BORRAR

- 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).