

# Enrutamiento, páginas de errores y personalización del tema con React con TypeScript.

---

Antes de nada:

No podemos olvidarnos de la instalación de:

```
npm install react-router-dom @mui/material @mui/icons-material  
@emotion/react @emotion/styled  
npm install --save-dev @types/react-router-dom
```

## 1. Enrutamiento con react-router-dom

El enrutamiento permite dividir la aplicación en distintas vistas (Login, Inicio, Informes...). Con la librería react-router-dom se define un objeto que contiene todas las rutas disponibles y el componente que se debe mostrar en cada una.

Para crear el enrutador se utiliza la función `createBrowserRouter`, que recibe un arreglo con las rutas. Cada ruta incluye un 'path' (dirección) y un 'element' (el componente que se muestra). Después, se utiliza `RouterProvider` para activar el enrutamiento en toda la aplicación.

Ejemplo de cómo hacer esto:

- `src/router/enrutador.tsx` (define rutas y `RouterProvider`)
- `src/App.tsx`

Librerías a importar:

- `react-router-dom` → `createBrowserRouter`, `RouterProvider`

Trozo de código — src/router/enrutador.tsx:

```
import React from 'react'
import { createBrowserRouter, RouterProvider } from 'react-router-dom'

import Acceso from './paginas/Acceso'
import Panel from './paginas/Panel'
import Estadisticas from './paginas/Estadisticas'
import PaginaErrorReferencia from './paginas/PaginaErrorReferencia'

// Definimos las rutas (cada ruta tiene 'path' y 'element')
const rutas = createBrowserRouter([
  { path: '/acceso', element: <Acceso />, errorElement:
<PaginaErrorReferencia /> },
  { path: '/panel', element: <Panel /> },
  { path: '/estadisticas', element: <Estadisticas /> }
])

// Proveedor de rutas para usar en la App
export function ProveedorRutas(){
  return <RouterProvider router={rutas} />
}
```

Trozo de código — src/App.tsx (inyectar proveedor de rutas):

```
import React from 'react'
import { ProveedorTema } from './tema/ProveedorTema'
import { ProveedorRutas } from './router/enrutador'

export default function App(){
  return (
```

```
<ProveedorTema>
  <ProveedorRutas />
</ProveedorTema>
)
}
```

## 2. Página de error

La página de error aparece cuando el usuario intenta acceder a una ruta que no existe. Se define con la propiedad `errorElement` dentro del enrutador.

El hook `useRouteError` permite capturar y mostrar el error correspondiente, por ejemplo un 404. En esta página se puede incluir un enlace para volver al inicio de sesión.

Dentro de nuestra carpeta `src` creamos la carpeta `páginas` y ahí irá el archivo `PaginaErrorReferencia.tsx`

Librerías a importar:

- `react-router-dom` → `useRouteError`, `isRouteErrorResponse`, `Link`

Trozo de código — `src/páginas/PaginaErrorReferencia.tsx`:

```
import React from 'react'
import { useRouteError, isRouteErrorResponse, Link } from 'react-router-dom'
import { Box, Typography } from '@mui/material'

export default function PaginaErrorReferencia(){
  const error = useRouteError() // Lee el error lanzado por el router

  const titulo = isRouteErrorResponse(error)
```

```

? `${error.status} · ${error.statusText}`
: 'Ruta no encontrada'

const detalle = isRouteErrorResponse(error)
? (error.data || 'Ha ocurrido un error al cargar esta vista.')
: 'Comprueba la URL o vuelve al acceso.'

return (
  <Box sx={{ minHeight:'100vh', display:'grid', placeItems:'center',
textAlign:'center', p:3 }}>
    <div>
      <Typography variant="h3" gutterBottom>{titulo}</Typography>
      <Typography sx={{ mb:3 }}>{String(detalle)}</Typography>
      <Link to="/acceso">Volver al acceso</Link>
    </div>
  </Box>
)
}

```

### 3. Tema de MUI

MUI (Material UI) permite personalizar el aspecto general de la aplicación mediante un tema global. Con `createTheme` se definen los colores principales y secundarios, la tipografía y el fondo. El componente `ThemeProvider` aplica ese tema a toda la aplicación. Además, `CssBaseline` se usa para homogeneizar los estilos base de todos los navegadores.

- `src/tema/ProveedorTema.tsx` (define y aplica el tema)
- `src/App.tsx` (usa `ProveedorTema` como contenedor)

Librerías a importar:

- @mui/material → CssBaseline
- @mui/material/styles → ThemeProvider, createTheme

Trozo de código — src/tema/ProveedorTema.tsx:

```

import React from 'react'
import CssBaseline from '@mui/material/CssBaseline'
import ThemeProvider from '@mui/material/styles/ThemeProvider'
import createTheme from '@mui/material/styles/createTheme'

const temaLumos = createTheme({
  palette: {
    primary: { main: '#264653' },
    secondary: { main: '#e76f51' },
    background: { default: '#f7f7f7' }
  },
  typography: { fontFamily: 'Inter, system-ui, Arial' }
})

export function ProveedorTema({ children }: { children: React.ReactNode }) {
  return (
    <ThemeProvider theme={temaLumos}>
      <CssBaseline />
      {children}
    </ThemeProvider>
  )
}

```

#### 4. Integración del Login con el enrutamiento

Una vez comprendidos los formularios, se añade la funcionalidad de navegación tras el inicio de sesión. Cuando el usuario introduce las

Si las credenciales son correctas, el sistema muestra un mensaje de éxito y lo redirige a la ruta '/inicio' utilizando el hook `useNavigate`. Si las credenciales no son válidas, se muestra un mensaje de error y no se realiza la navegación.

- `src/paginas/Acceso.tsx` (NO trae el formulario resuelto: solo referencia)

Librerías a importar:

- `react-router-dom` → `useNavigate`

Trozo de código — `src/paginas/Acceso.tsx` (solo referencia, en comentario):

```
// import { useNavigate } from 'react-router-dom'  
// const navegar = useNavigate()  
  
// const manejarEnvio = (evento: React.FormEvent<HTMLFormElement>)  
=> {  
  // evento.preventDefault()  
  // const accesoCorrecto = /* tu validación aquí */  
  // if (accesoCorrecto) {  
  //   navegar('/panel') // Cambia de ruta al panel  
  // }  
  //}  
// }
```

## 5. Páginas de destino

Las páginas Inicio e Informes son ejemplos sencillos que sirven para verificar que el enrutamiento funciona correctamente. En la página Inicio se incluyen botones para ir a Informes o cerrar sesión. En la página Informes se muestra contenido simulado.

- src/paginas/Panel.tsx
- src/paginas/Estadisticas.tsx

Trozo de código — src/paginas/Panel.tsx:

```
import React from 'react'
import { Box, Typography } from '@mui/material'

export default function Panel(){
  return (
    <Box sx={{ p:3 }}>
      <Typography variant="h4" gutterBottom>Panel LUMOS</Typography>
      <Typography>TODO: Contenido del panel tras iniciar
sesión.</Typography>
    </Box>
  )
}
```

Trozo de código — src/paginas/Estadisticas.tsx:

```
import React from 'react'
import { Box, Typography } from '@mui/material'

export default function Estadisticas(){
  return (
    <Box sx={{ p:3 }}>
      <Typography variant="h4" gutterBottom>Estadísticas</Typography>
      <Typography>TODO: Datos ficticios o gráficos que el alumnado
diseñe.</Typography>
    </Box>
  )
}
```

}