

Clase N° 7: NextJS



Objetivos de la clase

- Despejar dudas de la clase pasada.
- Profundizar en rutas dinámicas.
- Server side VS Client side.
- Presentar los posibles proyectos.



Link – useRouter – redirect

- **Link:** es el estándar para navegar entre páginas en la aplicación de NextJs. Cuenta con las ventajas de no recargar toda la página y la precarga de contenido. Ej: ir del Home al About.
- **useRouter:** se usa para el manejo de rutas en respuesta a eventos. Por ejemplo al completar un formulario, queremos que nuestro usuario vuelva al inicio de la página. También permite acceder a parámetros de la URL.
- **redirect:** se utiliza en server side. Útil para bloquear acceso a zonas restringidas o que requieren autenticación.

Ejemplos

→ Vamos al código para poder entenderlo mejor

```
app > login > page.tsx > Login
1  "use client";
2
3  import { useRouter } from "next/navigation";
4  import { useState } from "react";
5
6  const Login = () => {
7    const router = useRouter();
8    const [email, setEmail] = useState("");
9    const [password, setPassword] = useState("");
10
11    const handleLogin = async () => {
12      if (email === "user@example.com" && password === "password123") {
13        localStorage.setItem("isLoggedIn", "true");
14        router.push("/dashboard");
15      } else {
16        alert("Credenciales incorrectas");
17      }
18    }
19  }
```

Importante notar que:

- Persistencia del login: En aplicaciones reales, se recomienda usar cookies o sessionStorage junto con tokens seguros como JWT para mayor seguridad.
- router.replace vs router.push: **router.replace("/login")** no agrega la redirección al historial del navegador, lo que evita que el usuario regrese al dashboard presionando "atrás".
- LocalStorage y cliente: Como localStorage solo está disponible en el **cliente**, usamos el hook **useEffect** para verificar después de que el componente haya sido renderizado.

useRouter vs redirect

useRouter	redirect
Se usa en componentes clientSide.	se usa en componentes ServerSide.
Ideal para redirecciones basadas en interacciones del usuario.	Ideal para proteger rutas o redirecciones basadas en lógica del servidor.
Ej: Navegar al carrito despues de agregar un producto	Redirigir al login si el usuario no está autenticado.

[Client vs Server] components

Característica	useRouter	redirect
Renderizado	En el navegador del cliente	En el servidor
Interactividad	Si.	No.
Acceso a navegador	Si (window, localStorage).	No.
Estados y efectos	Si.	No.
SEO	Menos eficiente.	Excelente.

Proyectos

- La idea de estos proyectos es que vayan evolucionando a lo largo del cursado.
- Hay algunos puntos los cuales aún no hemos cubierto.
- Cada idea de proyecto es solo a modo de plantilla y puede ser personalizado a gusto de cada uno de ustedes.
- Si tienen otra idea previa que quieren implementar, siempre es bienvenida.

Preguntas?