



PROGRAMACIÓN WEB 2



Profesor(a):

Carlo Jose Luis Corrales Delgado

Estudiantes:

Yenaro Joel Noa Camino
Armando Steven Cuno Cahuari
Victor Narciso Mamani Anahua
Jorge Luis Mamani Huarsaya

Repositorio GitHub:

<https://github.com/ynoacamino/pweb2-final-project>

6 de julio, 2024

Frontend de la aplicación

El directorio `client/` contiene el código fuente que está organizado de acuerdo al uso de **webpack**, un empaquetador de módulos JavaScript. Webpack es un paquete que se encarga de compilar múltiples archivos JavaScript, CSS y otros activos en un solo archivo o en varios archivos optimizados para la producción. Este paquete es utilizado por React, el framework más popular a nivel mundial para construir interfaces de usuario.

tecnologías usadas

Framework React

React es una biblioteca de JavaScript desarrollada por Facebook para construir interfaces de usuario. Se basa en componentes, lo que permite crear elementos de UI reutilizables y gestionables. Cada componente de React puede manejar su propio estado y lógica, y se puede combinar con otros componentes para crear aplicaciones complejas.

React también facilita la creación de aplicaciones web modernas con arquitecturas basadas en componentes, lo que permite a los desarrolladores dividir la UI en piezas más pequeñas y manejables. Esta modularidad no solo mejora la organización del código, sino que también facilita el mantenimiento y la escalabilidad del proyecto.

Ventajas de Usar React

- **Componentes Reutilizables:** Los componentes de React pueden ser reutilizados en diferentes partes de la aplicación, lo que reduce la duplicación de código y facilita el mantenimiento.
- **Virtual DOM:** Permite actualizaciones eficientes y rápidas del DOM, mejorando el rendimiento de la aplicación.
- **JSX:** Una extensión de JavaScript que permite escribir HTML dentro de JavaScript, haciendo que el código sea más legible y fácil de escribir.
- **Unidirectional Data Flow:** La gestión del estado en React es más predecible gracias al flujo de datos unidireccional, lo que facilita la depuración y el seguimiento de cómo los datos fluyen a través de la aplicación.

Estructura de un Componente React Cada componente en React puede definirse como una función o una clase que devuelve un fragmento de la interfaz de usuario. A continuación, se muestra un ejemplo de un componente funcional básico en React:

```
1 import React from 'react';
2
3 function Welcome(props) {
4   return <h1>Hello, {props.name}</h1>;
5 }
6
7 export default Welcome;
```

En este ejemplo, `Welcome` es un componente funcional que toma **props** (propiedades) como argumento y retorna un encabezado con un saludo personalizado. Los componentes de clase también pueden definir métodos de ciclo de vida y manejar estados internos más complejos.

Integración con Tailwind CSS

Para estilizar los componentes de React, en este proyecto se utiliza Tailwind CSS. Tailwind es un framework de utilidades CSS que permite aplicar estilos directamente en los componentes mediante clases predefinidas. Esta metodología facilita el desarrollo de interfaces de usuario rápidas y consistentes.

El siguiente es un ejemplo de un componente estilizado con Tailwind CSS:

```
1 import React from 'react';
2
3 function Button({ label }) {
4   return (
5     <button className="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4
6       ↪ rounded">
7       {label}
8     </button>
9   );
10 }
11 export default Button;
```

En este ejemplo, las clases de Tailwind se utilizan para definir el estilo del botón, como el color de fondo, el comportamiento al pasar el ratón por encima, el color del texto y el espaciado.

Webpack

Webpack es una herramienta de construcción de módulos que se utiliza para agrupar, compilar y gestionar dependencias en proyectos JavaScript. Permite transformar archivos de diferentes tipos (JavaScript, CSS, imágenes, etc.) en un conjunto de activos que pueden ser servidos por un servidor web. Algunas características clave de Webpack incluyen:

- **Entrada y Salida:** Webpack toma uno o varios archivos de entrada y los transforma en uno o varios archivos de salida optimizados.
- **Loaders:** Transforman archivos de otros tipos en módulos válidos que Webpack puede procesar.
- **Plugins:** Extienden la funcionalidad de Webpack para optimizar el empaquetado, gestionar el hot reloading, minificación, entre otros.
- **Code Splitting:** Permite dividir el código en diferentes archivos para optimizar el rendimiento y la carga inicial de la aplicación.

Next.js

Next.js es un framework para React que permite la creación de aplicaciones web tanto del lado del cliente como del servidor. Facilita la implementación de aplicaciones web con funcionalidades como el renderizado del lado del servidor (SSR), generación de sitios estáticos (SSG), y rutas dinámicas.

- **Renderizado del Lado del Servidor (SSR):** Permite generar contenido HTML en el servidor en lugar del cliente, lo que puede mejorar el rendimiento y SEO.
- **Generación de Sitios Estáticos (SSG):** Genera páginas HTML estáticas en el momento de la compilación, optimizando el rendimiento y tiempo de carga.
- **Rutas Dinámicas:** Facilita la creación de rutas dinámicas basadas en la estructura de archivos del proyecto.
- **API Routes:** Permite definir endpoints API dentro del proyecto Next.js, facilitando la creación de backends ligeros.

Este directorio `client/` contiene una estructura definida y varios archivos de configuración esenciales para la correcta implementación de la aplicación.

- `.eslintrc.json`: Configuración de ESLint para mantener la calidad y consistencia del código. A continuación se muestra un extracto relevante:

```
1 {
2   "extends": ["airbnb", "airbnb-typescript", "next/core-web-vitals"],
3   "parserOptions": {
4     "project": "./tsconfig.json"
5   },
6   "rules": {
7     "linebreak-style": 0,
8     "react/jsx-props-no-spreading": 0,
9     "import/no-extraneous-dependencies": 0,
10    "import/prefer-default-export": 0,
11    "react/require-default-props": 0,
12    "jsx-a11y/label-has-associated-control": 0,
13    "@typescript-eslint/naming-convention": 0,
14    "eslint-disable no-param-reassign": 0,
15    "no-param-reassign": 0
16  }
17 }
```

- `tailwind.config.ts`: Configuración de Tailwind CSS para definir temas, colores, espaciados y otras utilidades personalizadas que se usarán en el proyecto:

```
1 import type { Config } from 'tailwindcss';
2
3 const { fontFamily } = require('tailwindcss/defaultTheme');
4
5 const config = {
6   darkMode: ['class'],
7   content: [
8     './pages/**/*.{ts,tsx}',
9     './components/**/*.{ts,tsx}',
10    './app/**/*.{ts,tsx}',
11    './src/**/*.{ts,tsx}',
12  ],
13   prefix: '',
14   theme: {
15     container: {
16       center: true,
17       padding: '2rem',
18       screens: {
19         '2xl': '1400px',
20       },
21     },
22     extend: {
23       colors: {
24         border: 'hsl(var(--border))',
25         input: 'hsl(var(--input))',
26         ring: 'hsl(var(--ring))',
27         background: 'hsl(var(--background))',
```

```
28     foreground: 'hsl(var(--foreground))',
29     primary: {
30       DEFAULT: 'hsl(var(--primary))',
31       text: 'hsl(var(--primary-text))',
32       foreground: 'hsl(var(--primary-foreground))',
33     },
34     secondary: {
35       DEFAULT: 'hsl(var(--secondary))',
36       foreground: 'hsl(var(--secondary-foreground))',
37     },
38     destructive: {
39       DEFAULT: 'hsl(var(--destructive))',
40       foreground: 'hsl(var(--destructive-foreground))',
41     },
42     muted: {
43       DEFAULT: 'hsl(var(--muted))',
44       foreground: 'hsl(var(--muted-foreground))',
45     },
46     accent: {
47       DEFAULT: 'hsl(var(--accent))',
48       foreground: 'hsl(var(--accent-foreground))',
49     },
50     popover: {
51       DEFAULT: 'hsl(var(--popover))',
52       foreground: 'hsl(var(--popover-foreground))',
53     },
54     card: {
55       DEFAULT: 'hsl(var(--card))',
56       foreground: 'hsl(var(--card-foreground))',
57     },
58   },
59   borderRadius: {
60     lg: 'var(--radius)',
61     md: 'calc(var(--radius) - 2px)',
62     sm: 'calc(var(--radius) - 4px)',
63   },
64   keyframes: {
65     'accordion-down': {
66       from: { height: '0' },
67       to: { height: 'var(--radix-accordion-content-height)' },
68     },
69     'accordion-up': {
70       from: { height: 'var(--radix-accordion-content-height)' },
71       to: { height: '0' },
72     },
73   },
74   animation: {
75     'accordion-down': 'accordion-down 0.2s ease-out',
76     'accordion-up': 'accordion-up 0.2s ease-out',
77   },
78   fontFamily: {
79     sans: ['var(--font-sans)', ...fontFamily.sans],
80   },
81 },
```

```
82   },
83   plugins: [require('tailwindcss-animate')],
84 } satisfies Config;
85
86 export default config;
```

- **postcss.config.mjs**: Configuración de PostCSS para transformar CSS con plugins, como autoprefixer:

```
1  /** @type {import('postcss-load-config').Config} */
2  const config = {
3    plugins: {
4      tailwindcss: {},
5    },
6  };
7
8  export default config;
```

- **tsconfig.json**: Configuración del compilador TypeScript para especificar opciones de compilación y manejo de archivos:

```
1  {
2    "compilerOptions": {
3      "lib": ["dom", "dom.iterable", "esnext"],
4      "allowJs": true,
5      "skipLibCheck": true,
6      "strict": true,
7      "noEmit": true,
8      "esModuleInterop": true,
9      "module": "esnext",
10     "moduleResolution": "bundler",
11     "resolveJsonModule": true,
12     "isolatedModules": true,
13     "jsx": "preserve",
14     "incremental": true,
15     "plugins": [
16       {
17         "name": "next"
18       }
19     ],
20     "paths": {
21       "@/*": [".src/*"]
22     }
23   },
24   "include": ["next-env.d.ts", "**/*.ts", "**/*.tsx", ".next/types/**/*.ts"],
25   "exclude": ["node_modules"]
26 }
```

- **next.config.mjs**: Configuración específica de Next.js para definir rutas personalizadas, configuraciones de webpack y variables de entorno:

```
1  /** @type {import('next').NextConfig} */
2  const nextConfig = {
3    images: {
4      remotePatterns: [
5        {
6          hostname: 'lh3.googleusercontent.com',
7          protocol: 'https',
8        },
9        {
10         hostname: 'avatars.githubusercontent.com',
11         protocol: 'https',
12       },
13       {
14         hostname: 'res.cloudinary.com',
15         protocol: 'https'
16       }
17     ]
18   }
19 };
20
21 export default nextConfig;
```

- **package.json:** Listado de dependencias y scripts del proyecto, incluyendo librerías y herramientas necesarias para el desarrollo y producción:

```
1  {
2    "name": "client",
3    "version": "0.1.0",
4    "private": true,
5    "scripts": {
6      "dev": "next dev",
7      "build": "next build",
8      "start": "next start",
9      "lint": "next lint"
10   },
11   "dependencies": {
12     "@radix-ui/react-dialog": "^1.1.1",
13     "@radix-ui/react-dropdown-menu": "^2.1.1",
14     "@radix-ui/react-label": "^2.1.0",
15     "@radix-ui/react-slot": "^1.0.2",
16     "class-variance-authority": "^0.7.0",
17     "clsx": "^2.1.1",
18     "embla-carousel-react": "^8.1.5",
19     "lucide-react": "^0.395.0",
20     "next": "14.2.4",
21     "next-auth": "^4.24.7",
22     "next-themes": "^0.3.0",
23     "react": "^18",
24     "react-dom": "^18",
25     "tailwind-merge": "^2.3.0",
26     "tailwindcss-animate": "^1.0.7"
27   },
28   "devDependencies": {
```

```
29   "@types/node": "^20",
30   "@types/react": "^18",
31   "@types/react-dom": "^18",
32   "@typescript-eslint/eslint-plugin": "^6.0.0",
33   "@typescript-eslint/parser": "^6.0.0",
34   "eslint": "^8",
35   "eslint-config-airbnb": "^19.0.4",
36   "eslint-config-airbnb-typescript": "^17.1.0",
37   "eslint-config-next": "14.2.4",
38   "postcss": "^8",
39   "tailwindcss": "^3.4.1",
40   "typescript": "^5"
41 }
42 }
```

- **public/**: Contiene archivos estáticos como imágenes y fuentes que son servidos directamente por el servidor web.

Autenticación con NextAuth

NextAuth.js es una biblioteca completa para la autenticación en aplicaciones Next.js, soportando varios proveedores de autenticación.

Configuración de NextAuth

La configuración se encuentra en `src/app/api/auth/[...nextauth]/authOptions.ts`. Este archivo define los proveedores de autenticación y los callbacks para manejar eventos durante el flujo de autenticación: