



**Instituto Tecnológico de Costa Rica
Campus Tecnológico Central Cartago
Escuela de Computación**

Curso: Proyecto de Ingeniería de Software

Proyecto: Expediente de Profesores

Informe de Pruebas

Profesora:

Alicia Marcela Salazar Hernández

Integrantes del grupo:

Gabriela Gutiérrez Valverde

David Jose Espinoza Soto

Ulises Rodríguez Pérez

Fecha de entrega: Lunes 16 de Octubre del 2023

Introducción.....	3
Procedimientos de pruebas.....	3
Resultados de Pruebas.....	19
Conclusiones.....	30
Recomendaciones.....	30

Introducción

El presente documento tiene como objetivo presentar los resultados obtenidos basados en las pruebas que se hicieron siguiendo el plan de pruebas de aseguramiento de la calidad del Software realizadas en la aplicación Expediente de Profesores para la Escuela de Computación del Instituto Tecnológico de Costa Rica. Estas pruebas se llevaron a cabo con el propósito de identificar y corregir posibles errores, esto lo realizamos usando: Pruebas Estáticas, Dinámicas, Unitarias, de Integración, de Sistema y de Aceptación de Usuario.

Procedimientos de pruebas

Objetivo RF001	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	Prueba unitaria escrita en JavaScript usando la biblioteca de pruebas Jest y la biblioteca de pruebas de componentes Enzyme.
Documentación	<p>Código:</p> <pre>// Renders the Router component with Routes and multiple Route components with their respective paths and elements. it('should render the Router component with Routes and multiple Route components', () => { // Mock the necessary dependencies jest.mock('react-router-dom', () => ({ BrowserRouter: ({ children }) => <div>{children}</div>, Route: ({ path, element }) => <div>{path}{element}</div>, Routes: ({ children }) => <div>{children}</div>, })); // Import the necessary dependencies const ReactDOM = require('react-dom'); const { BrowserRouter, Route, Routes } = ReactDOM;</pre>

```
// Render the App component
const wrapper = mount(
  <BrowserRouter>
    <Routes>
      <App />
    </Routes>
  </BrowserRouter>
);

// Assert that the Router component is rendered
expect(wrapper.find(BrowserRouter)).toHaveLength(1);

// Assert that the Routes component is rendered
expect(wrapper.find(Routes)).toHaveLength(1);

// Assert that the Route components are rendered with
their respective paths and elements
expect(wrapper.find(Route)).toHaveLength(5);

expect(wrapper.find(Route).at(0).props().path).toBe('/login');

expect(wrapper.find(Route).at(0).props().element.type).toBe(Login);
    expect(wrapper.find(Route).at(1).props().path).toBe('/');

expect(wrapper.find(Route).at(1).props().element.props.component.type).toBe(PrivateRoute);

expect(wrapper.find(Route).at(1).props().element.props.component.props.component.type).toBe(Home);

expect(wrapper.find(Route).at(2).props().path).toBe('/register');

expect(wrapper.find(Route).at(2).props().element.props.requiredRoles).toEqual(['admin']);

expect(wrapper.find(Route).at(2).props().element.props.component.type).toBe(RoleBasedRoute);

expect(wrapper.find(Route).at(2).props().element.props.component.props.component.type).toBe(Register);

expect(wrapper.find(Route).at(3).props().path).toBe('/admin-dashboard');
```

	<pre> expect(wrapper.find(Route).at(3).props().element.props.required Roles).toEqual(['admin']); expect(wrapper.find(Route).at(3).props().element.props.component .type).toBe(RoleBasedRoute); expect(wrapper.find(Route).at(3).props().element.props.component .props.component.type).toBe(AdminDashboard); expect(wrapper.find(Route).at(4).props().path).toBe('/user-dash board'); expect(wrapper.find(Route).at(4).props().element.props.required Roles).toEqual(['user', 'admin']); expect(wrapper.find(Route).at(4).props().element.props.component .type).toBe(RoleBasedRoute); expect(wrapper.find(Route).at(4).props().element.props.component .props.component.type).toBe(UserDashboard); }); </pre>
Caso de prueba 2	
Ejecución	Prueba unitaria está diseñada para verificar si el componente Login se renderiza correctamente cuando la ruta (path) es igual a '/login'.
Documentación	<pre> it('should render the Login component when the path is "/login"', () => { // Mock the necessary dependencies jest.mock('react-router-dom', () => ({ BrowserRouter: ({ children }) => <div>{children}</div>, Route: ({ path, element }) => { if (path === '/login') { return <div>{element}</div>; } else { return null; } }, Routes: ({ children }) => <div>{children}</div>, })); // Import the necessary dependencies const ReactRouterDom = require('react-router-dom'); const { BrowserRouter, Route, Routes } = ReactRouterDom; // Render the App component const wrapper = mount(</pre>

	<pre> <BrowserRouter> <Routes> <App /> </Routes> </BrowserRouter>); // Assert that the Login component is rendered when the path is '/login' expect(wrapper.find(Route).at(0).props().path).toBe('/login'); expect(wrapper.find(Route).at(0).props().element.type).toBe(Login); }); </pre>
Caso de prueba 3	
Ejecución	<p>Prueba Unitaria tiene como objetivo verificar si el componente Home se renderiza correctamente cuando la ruta (path) es igual a '/' y el usuario está autenticado.</p>
Documentación	<pre> it('should render the Home component when the path is "/" and the user is authenticated', () => { // Mock the necessary dependencies jest.mock('react-router-dom', () => ({ BrowserRouter: ({ children }) => <div>{children}</div>, Route: ({ path, element }) => { if (path === '/') { return <div>{element}</div>; } else { return null; } }, Routes: ({ children }) => <div>{children}</div>, })); // Mock the localStorage global.localStorage = { getItem: jest.fn().mockReturnValue(JSON.stringify({ id: 1, name: 'John', email: 'john@example.com', roles: ['user'] })), }; // Import the necessary dependencies const ReactRouterDom = require('react-router-dom'); const { BrowserRouter, Route, Routes } = ReactRouterDom; // Render the App component const wrapper = mount(<BrowserRouter> <Routes> <App /> </Routes> </BrowserRouter>); </pre>

	<pre> // Assert that the Home component is rendered when the path is '/' and the user is authenticated expect(wrapper.find(Route).at(1).props().path).toBe('/'); expect(wrapper.find(Route).at(1).props().element.props.component.props.component.type).toBe(Home); }); </pre>
Caso de prueba 4	
Ejecución	<p>Esta prueba unitaria tiene como objetivo verificar si el componente Login se renderiza correctamente cuando la ruta (path) es igual a '/' y el usuario no está autenticado.</p>
Documentación	<pre> it('should render the Login component when the path is "/" and the user is not authenticated', () => { // Mock the necessary dependencies jest.mock('react-router-dom', () => ({ BrowserRouter: ({ children }) => <div>{children}</div>, Route: ({ path, element }) => { if (path === '/') { return <div>{element}</div>; } else { return null; } }, Routes: ({ children }) => <div>{children}</div>, })); // Mock the localStorage global.localStorage = { getItem: jest.fn().mockReturnValue(null), }; // Import the necessary dependencies const ReactDOM = require('react-router-dom'); const { BrowserRouter, Route, Routes } = ReactDOM; // Render the App component const wrapper = mount(<BrowserRouter> <Routes> <App /> </Routes> </BrowserRouter>); // Assert that the Login component is rendered when the path is '/' and the user is not authenticated expect(wrapper.find(Route).at(1).props().path).toBe('/'); expect(wrapper.find(Route).at(1).props().element.props.component.type).toBe(Login); }); </pre>

Objetivo RF002	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	
Documentación	
Caso de prueba 2	
Ejecución	
Documentación	

Objetivo RF003	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	
Documentación	
Caso de prueba 2	

Ejecución	
Documentación	

Objetivo RF004	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	
Documentación	
Caso de prueba 2	
Ejecución	
Documentación	

Objetivo RF005	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	

Ejecución	
Documentación	
Caso de prueba 2	
Ejecución	
Documentación	

Objetivo RF006	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	
Documentación	
Caso de prueba 2	
Ejecución	
Documentación	

Objetivo RF007

Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	
Documentación	
Caso de prueba 2	
Ejecución	
Documentación	

Objetivo RF008	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	
Documentación	
Caso de prueba 2	
Ejecución	

Documentación	
----------------------	--

Objetivo RF009	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	
Documentación	
Caso de prueba 2	
Ejecución	
Documentación	

Objetivo RF010	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	
Documentación	

Caso de prueba 2	
Ejecución	
Documentación	

Objetivo RF011	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	
Documentación	
Caso de prueba 2	
Ejecución	
Documentación	

Objetivo RF012	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	

Ejecución	
Documentación	
Caso de prueba 2	
Ejecución	
Documentación	

Objetivo RF013	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	
Documentación	
Caso de prueba 2	
Ejecución	
Documentación	

Objetivo RF014

Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	
Documentación	
Caso de prueba 2	
Ejecución	
Documentación	

Objetivo RF015	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	
Documentación	
Caso de prueba 2	
Ejecución	

Documentación	
----------------------	--

Objetivo RF016	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	
Documentación	
Caso de prueba 2	
Ejecución	
Documentación	

Objetivo RF017	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	
Documentación	

Caso de prueba 2	
Ejecución	
Documentación	

Objetivo RF018	
Configuración del entorno	Ejecutar el comando npm run dev y acceder a la URL http://127.0.0.1:3000/
Caso de prueba 1	
Ejecución	
Documentación	
Caso de prueba 2	
Ejecución	
Documentación	

Resultados de Pruebas

Objetivo RF001	
Estadísticas de Ejecución	Cantidad de pruebas realizadas: 4 Número de errores encontrados: 0 Tasa de éxito en las pruebas: 100%
Problemas o errores	
Caso de prueba 1	
Resultados	Se confirma que la configuración de enrutamiento de la aplicación se haya configurado correctamente y que los componentes Route se hayan configurado adecuadamente con sus rutas y elementos correspondientes.
Caso de prueba 2	
Resultados	Las afirmaciones comprueban si, cuando la ruta es igual a '/login', el componente Login se renderiza correctamente. Si ambas afirmaciones son verdaderas, la prueba se considera exitosa y demuestra que el enrutamiento y la representación del componente Login funcionan como se esperaba en ese escenario específico.
Caso de prueba 3	
Resultados	Las afirmaciones comprueban si, cuando la ruta es igual a '/' y el usuario está autenticado, el componente Home se

	renderiza correctamente. Si ambas afirmaciones son verdaderas, la prueba se considera exitosa y demuestra que el enrutamiento y la representación del componente Home funcionan como se esperaba en ese escenario específico de usuario autenticado.
Caso de prueba 4	
Resultados	Las afirmaciones comprueban si, cuando la ruta es igual a '/' y el usuario no está autenticado, el componente Login se renderiza correctamente. Si ambas afirmaciones son verdaderas, la prueba se considera exitosa y demuestra que el enrutamiento y la representación del componente Login funcionan como se esperaba en ese escenario específico de usuario no autenticado.

Objetivo RF002	
Estadísticas de Ejecución	Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas o errores	
Caso de prueba 1	
Resultados	
Caso de prueba 2	

Resultados	
-------------------	--

Objetivo RF003	
Estadísticas de Ejecución	Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas o errores	
Caso de prueba 1	
Resultados	
Caso de prueba 2	
Resultados	

Objetivo RF004	
Estadísticas de Ejecución	Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas o errores	
Caso de prueba 1	

Resultados	
Caso de prueba 2	
Resultados	

Objetivo RF005	
Estadísticas de Ejecución	Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas o errores	
Caso de prueba 1	
Resultados	
Caso de prueba 2	
Resultados	

Objetivo RF006	
Estadísticas de Ejecución	Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:

Problemas errores	o	
Caso de prueba 1		
Resultados		
Caso de prueba 2		
Resultados		

Objetivo RF007		
Estadísticas Ejecución	de	Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas errores	o	
Caso de prueba 1		
Resultados		
Caso de prueba 2		
Resultados		

Objetivo RF008		
-----------------------	--	--

Estadísticas de Ejecución	Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas o errores	
Caso de prueba 1	
Resultados	
Caso de prueba 2	
Resultados	

Objetivo RF009	
Estadísticas de Ejecución	Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas o errores	
Caso de prueba 1	
Resultados	
Caso de prueba 2	
Resultados	

Objetivo RF010		
Estadísticas de Ejecución		Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas o errores		
Caso de prueba 1		
Resultados		
Caso de prueba 2		
Resultados		

Objetivo RF011		
Estadísticas de Ejecución		Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas o errores		
Caso de prueba 1		
Resultados		
Caso de prueba 2		

Resultados	
-------------------	--

Objetivo RF012	
Estadísticas de Ejecución	Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas o errores	
Caso de prueba 1	
Resultados	
Caso de prueba 2	
Resultados	

Objetivo RF013	
Estadísticas de Ejecución	Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas o errores	
Caso de prueba 1	
Resultados	

Caso de prueba 2	
Resultados	

Objetivo RF014	
Estadísticas de Ejecución	Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas o errores	
Caso de prueba 1	
Resultados	
Caso de prueba 2	
Resultados	

Objetivo RF015	
Estadísticas de Ejecución	Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas o errores	
Caso de prueba 1	

Resultados	
Caso de prueba 2	
Resultados	

Objetivo RF016	
Estadísticas de Ejecución	Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas o errores	
Caso de prueba 1	
Resultados	
Caso de prueba 2	
Resultados	

Objetivo RF017	
Estadísticas de Ejecución	Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas o	

errores	
Caso de prueba 1	
Resultados	
Caso de prueba 2	
Resultados	

Objetivo RF018	
Estadísticas de Ejecución	Cantidad de pruebas realizadas: Número de errores encontrados: Tasa de éxito en las pruebas:
Problemas o errores	
Caso de prueba 1	
Resultados	
Caso de prueba 2	
Resultados	

Conclusiones

- Las pruebas unitarias han demostrado que los componentes de enrutamiento y representación en la aplicación React funcionan según lo esperado en diferentes escenarios.
- Se ha verificado que el componente principal App configura adecuadamente el enrutamiento de la aplicación, incluyendo rutas protegidas y rutas basadas en roles.
- Los componentes Route, BrowserRouter, y Routes se han utilizado de manera efectiva para definir las rutas y sus elementos correspondientes.
- La prueba ha ayudado a identificar posibles problemas en la configuración de rutas y en la representación de componentes, asegurando la correcta funcionalidad de la aplicación en términos de enrutamiento.

Recomendaciones

- Mantener actualizadas las pruebas unitarias a medida que se realicen cambios en la aplicación para garantizar que sigan siendo válidas y útiles para la detección temprana de problemas.
- Explorar la posibilidad de agregar más pruebas que cubran otros escenarios y casos de uso, como pruebas para rutas adicionales o para verificar el comportamiento de componentes en diferentes estados.
- Considerar la incorporación de pruebas de integración o pruebas de extremo a extremo para evaluar la interacción entre varios componentes y funcionalidades de la aplicación en su conjunto.
- Documentar adecuadamente las pruebas unitarias para que otros miembros del equipo puedan comprender fácilmente su propósito y funcionamiento.