**Universidad Nacional Mayor de San Marcos  
Facultad de Ingeniería de Sistemas e Informática  
E.P. de Ingeniería de Software**

****

**mAI-DPS: Documento de Pruebas del Software**

**Integrantes**

Alva Sáenz, Rodrigo José 22200232  
Del Carpio Martinez, Damaris Marian 22200082  
Diaz Ingol, Jesus Stevan 22200083  
Maylle Colaca, Luis Fernando 21200221  
Monzon Argüelles, Oscar Sebastian 22200094  
Rojas Castañeda, Ruth Camila 22200239  
Rojas Rojas, Sebastian Alberto 22200261  
Zegarra Medina, Jose Antonio 22200102

**Índice**

[1. Introducción 3](#_jpx08hpdnfuv)

[1.1. Objetivos 3](#_gbyu4snysjz7)

[1.2. Descripción del Componente 3](#_97sqapp5qefb)

[2. Información General 4](#_czg0q38tuetx)

[3. Estrategia del Plan de Prueba 5](#_2qfgb4uwocvk)

[4. Entornos de Prueba 5](#_adfc6tb0ypmb)

[5. Ejecución de Pruebas 5](#_garxyn846t2r)

[5.3. User (user.test.js) 6](#_yl0ea6holmk5)

[5.4. Login (Login.test.jsx): 6](#_yna2vshiouv3)

[5.5. Header (Header.test.jsx) 6](#_9dsudsi5ylhi)

[5.6. MenuBurger (MenuBurger.test.jsx) 6](#_1dwx44mwib7v)

[5.7. ModuleSelect 6](#_lminl1gu1zo5)

[5.8. useGUIData 7](#_v3sx07h3dipo)

[6. Conclusión 7](#_9tl3lq9uy17f)

**Documento de Pruebas del Software**

# **Introducción**

Este documento describe el plan de pruebas para el proyecto "Toolkit de Productividad Corporativa (TPC)" desarrollado por la empresa de desarrollo de software mAI Solutions. El propósito de este documento es garantizar que el software cumpla con los requisitos especificados y funcione de manera efectiva y sin errores.

## **Objetivos**

El objetivo del plan de pruebas es asegurar que todos los componentes del sistema se comporten según lo esperado. Los objetivos específicos incluyen:

* Validar la correcta renderización y funcionamiento de los componentes principales del sistema, asegurando que la interfaz de usuario se presente correctamente en todos los navegadores y dispositivos compatibles.
* Asegurar que todas las interacciones del usuario sean manejadas adecuadamente, verificando que las respuestas del sistema sean correctas y se gestionan los errores de manera eficiente.
* Confirmar que los datos se gestionan de manera correcta y segura en toda la aplicación, garantizando la integridad y confidencialidad de la información del usuario.

## **Descripción del Componente**

El sistema se compone de varios componentes que se encargan de diferentes funcionalidades, como la autenticación de usuarios, la navegación por la interfaz y la gestión de datos del usuario. Cada componente ha sido diseñado para cumplir una función específica dentro del TPC y debe ser probado individualmente para asegurar su correcto funcionamiento.

# **Información General**

| **ID de Caso de Prueba** | **Nombre del Caso de Prueba** | **Módulo/Componente** | **Requerimiento** | **Prioridad** | **Tipo de Prueba** |
| --- | --- | --- | --- | --- | --- |
| TC-001 | Renderiza el componente App correctamente | App | El componente debe renderizarse sin errores | Alta | Caja blanca |
| TC-002 | Renderiza el componente UserGUI correctamente | UserGUI | El componente debe renderizarse sin errores | Alta | Caja blanca |
| TC-003 | Verifica autenticación de usuario | User | El usuario debe poder autenticarse | Alta | Caja blanca |
| TC-004 | Renderiza el componente Login correctamente | Login | El componente debe renderizarse sin errores | Alta | Caja blanca |
| TC-005 | Renderiza el componente Header correctamente | Header | El componente debe renderizarse sin errores | Alta | Caja blanca |
| TC-006 | Renderiza el componente MenuBurger correctamente | MenuBurger | El componente debe renderizarse sin errores | Alta | Caja blanca |
| TC-007 | Verifica la navegación entre módulos | ModuleSelect | El componente debe permitir la navegación entre módulos | Media | Caja blanca |
| TC-008 | Verifica la actualización de rutas | useGUIData | Las rutas deben actualizarse correctamente | Alta | Caja blanca |

# **Estrategia del Plan de Prueba**

El tipo de pruebas realizadas son pruebas de caja blanca, enfocadas en la estructura interna del código. Se utiliza la biblioteca Jest para ejecutar las pruebas y se emplean mocks para aislar los componentes y hooks durante las pruebas. Se seguirán las siguientes estrategias:

* **Pruebas Unitarias:** Asegurarse de que cada componente individual funcione correctamente por sí solo.
* **Pruebas de Integración:** Verificar que los componentes funcionen correctamente en conjunto.
* **Pruebas de Regresión:** Asegurar que los cambios en el código no introduzcan nuevos errores.

# **Entornos de Prueba**

Las pruebas se realizan utilizando las siguientes herramientas y entornos:

* **Herramienta de Prueba:** Jest, Vitest
* **Entorno de Desarrollo:** Node.js, React, BrowserRouter
* **Bibliotecas de Prueba:** @testing-library/react, @mantine/core, vitest, @tabler/icons-react
* **Mocking:** Jest y Vitest para simular comportamientos y datos.
* **Entorno de Ejecución:** Pruebas ejecutadas en ambientes de desarrollo locales y CI/CD (Integración Continua y Despliegue Continuo).

# **Ejecución de Pruebas**

Las pruebas cubren las siguientes características de cada componente:

* 1. **App (main.test.jsx)**
* **Funcionalidad Principal:** Renderización correcta del componente principal.
* **Validación:** Aseguramiento de que el componente App se encuentra en el DOM.
  1. **UserGUI (userGUI.test.js)**
* **Funcionalidad Principal:** Renderización correcta del componente UserGUI.
* **Validación:** Verificación de la correcta gestión de rutas y datos del usuario, incluyendo la interacción y actualización de la interfaz.

## **User (user.test.js)**

* **Funcionalidad Principal:** Verificación de la autenticación del usuario.
* **Validación:** Aseguramiento de que el usuario puede iniciar y cerrar sesión correctamente, así como la gestión de los datos de usuario almacenados.

## **Login (Login.test.jsx):**

* **Funcionalidad Principal:** Renderización correcta del formulario de login.
* **Validación:** Manejo correcto de la presentación y validación de credenciales, incluyendo la interacción del usuario con el formulario.

## **Header (Header.test.jsx)**

* **Funcionalidad Principal:** Renderización correcta del componente Header.
* **Validación:** Verificación de que los componentes hijos reciben las props adecuadas y funcionan correctamente dentro del contenedor Header.

## **MenuBurger (MenuBurger.test.jsx)**

* **Funcionalidad Principal:** Renderización y funcionalidad del menú hamburguesa.
* **Validación:** Aseguramiento de que el menú se abre y cierra correctamente, y que la navegación en los submenús funciona adecuadamente.

## **ModuleSelect**

* **Funcionalidad Principal:** Verificación de la navegación entre módulos.
* **Validación:** Aseguramiento de que los usuarios pueden seleccionar y cambiar entre diferentes módulos correctamente.

## **useGUIData**

* **Funcionalidad Principal:** Actualización y gestión de rutas.
* **Validación:** Verificación de que las rutas se actualizan correctamente según las interacciones del usuario.

# **Conclusión**

Este plan de pruebas asegura una cobertura adecuada de los componentes críticos del proyecto "Toolkit de Productividad Corporativa (TPC)", garantizando que funcionen correctamente y cumplan con los requisitos establecidos. La ejecución de estas pruebas proporcionará confianza en la calidad del software desarrollado, minimizando el riesgo de errores en producción y asegurando una experiencia de usuario satisfactoria.

Con este plan, se busca no solo identificar y corregir errores, sino también mejorar continuamente la calidad del código y la eficiencia del equipo de desarrollo.