

Documentación de Pruebas de Calidad

Módulo de Donaciones - Componente DonationForm y DonacionesDashboard

Introducción

Este documento presenta el plan de pruebas de calidad para el módulo de Donaciones, que incluye el formulario de donaciones (DonationForm) para usuarios finales y el dashboard de administración (DonacionesDashboard) para administradores. Se detallan los casos de prueba, criterios de aceptación y escenarios de prueba para asegurar la correcta funcionalidad del módulo.

Componentes Analizados:

- DonationForm (formulario de donaciones)
- DonacionesDashboard (dashboard administrativo)

Tecnologías utilizadas: React, Chart.js, React Router DOM, Bootstrap

Fecha de evaluación: 4 de junio de 2025

Descripción de los Componentes

DonationForm

El componente DonationForm permite a los usuarios registrados realizar donaciones al sistema CDMI. Incluye:

- Validación de rol de usuario
- Formulario con campos de donante y donación
- Selección de moneda
- Mensaje personalizado opcional
- Imagen de carrusel informativa

DonacionesDashboard

El componente DonacionesDashboard proporciona una vista administrativa de las donaciones, incluyendo:

- Validación de rol de administrador
- Tabla de registros contables
- Gráfico de barras con Chart.js

- Navegación lateral administrativa
- Datos de ingresos, gastos y saldos

Funcionalidades a Probar

DonationForm

1. Autenticación y Autorización

- Verificación de token de autenticación
- Validación de rol de usuario
- Redirección a login si no está autorizado

2. Formulario de Donación

- Ingreso de datos del donante
- Validación de campos obligatorios
- Selección de moneda
- Envío de formulario

DonacionesDashboard

1. Panel Administrativo

- Verificación de rol de administrador
- Visualización de datos contables
- Generación de gráfico de barras
- Navegación del panel

Casos de Prueba

Pruebas Funcionales - DonationForm

ID	Caso de Prueba	Datos de Entrada	Resultado Esperado	Criterio de Aceptación
DF-01	Donación exitosa con todos los datos	Nombre: "Juan Pérez" Email: "juan@email.com" Cantidad: 50000 Moneda: COP Mensaje: "Apoyo a la causa"	El formulario se envía correctamente y se muestra confirmación	La donación se registra en el sistema
DF-02	Donación sin mensaje opcional	Nombre: "María García" Email: "maria@email.com" Cantidad: 100 Moneda: USD Mensaje: vacío	El formulario se envía sin problemas	El campo mensaje es opcional y no impide el envío
DF-03	Intentar enviar con campos vacíos	Omitir nombre, email o cantidad	El formulario bloquea el envío y muestra indicaciones	No se envía hasta completar campos requeridos
DF-04	Validación de email inválido	Email: "correo-invalido"	El sistema debe validar formato de email	Se muestra error de formato de email
DF-05	Cantidad negativa o cero	Cantidad: 0 o -10	El sistema debe rechazar valores inválidos	Se impide el ingreso de valores negativos
DF-06	Acceso sin autenticación	Sin token o token inválido	Redirección automática a login	Usuario no autenticado no puede acceder
DF-07	Acceso con rol incorrecto	Token válido pero rol "administrador"	Redirección automática a login	Solo usuarios con rol "usuario" pueden acceder

Pruebas Funcionales - DonacionesDashboard

ID	Caso de Prueba	Datos de Entrada	Resultado Esperado	Criterio de Aceptación
DD-01	Acceso exitoso como administrador	Token válido con rol "administrador"	Dashboard se carga con todos los elementos	Tabla y gráfico se muestran correctamente
DD-02	Visualización de datos contables	Datos de ejemplo cargados	Tabla muestra fecha, cuenta, debe, haber, saldo y porcentaje	Todos los campos numéricos se formatean correctamente
DD-03	Generación de gráfico	Datos de saldo disponibles	Gráfico de barras se genera usando Chart.js	Gráfico es responsive y legible
DD-04	Acceso sin autenticación	Sin token	Redirección automática a login	Usuario no autenticado no puede acceder
DD-05	Acceso con rol incorrecto	Token válido pero rol "usuario"	Redirección automática a login	Solo administradores pueden acceder
DD-06	Funcionalidad de cerrar sesión	Click en "Cerrar sesión"	Se muestra alerta de confirmación	Alert funciona correctamente

Pruebas de Interfaz de Usuario

ID	Caso de Prueba	Pasos	Resultado Esperado	Criterio de Aceptación
UI-01	Visualización del carrusel	Acceder al formulario de donación	Imagen del carrusel se muestra correctamente	Imagen se carga sin errores
UI-02	Responsividad del formulario	Acceder desde dispositivo móvil	El formulario se adapta al tamaño de pantalla	Todos los elementos son visibles y utilizables
UI-03	Selección de moneda	Usar dropdown de moneda	Se pueden seleccionar USD, EUR, COP	Todas las opciones están disponibles
UI-04	Feedback visual campos requeridos	Intentar enviar sin completar campos	Se resaltan visualmente los campos obligatorios	Los campos requeridos muestran indicación clara
UI-05	Visualización de tabla de datos	Acceder al dashboard	Tabla se muestra con formato correcto	Datos numéricos se formatean con 2 decimales
UI-06	Interactividad del gráfico	Visualizar gráfico de barras	Gráfico es responsive y muestra datos correctos	Chart.js renderiza correctamente
UI-07	Navegación lateral	Usar menú del dashboard	Enlaces de navegación funcionan	Sidebar se muestra correctamente

Pruebas de Seguridad

ID	Caso de Prueba	Descripción	Resultado Esperado	Criterio de Aceptación
SEC-01	Validación de token DonationForm	Manipular o eliminar token de localStorage	El sistema detecta token inválido y redirige	Usuario es redirigido a login
SEC-02	Validación de token DonacionesDashboard	Manipular o eliminar token de localStorage	El sistema detecta token inválido y redirige	Usuario es redirigido a login
SEC-03	Inyección de código en campos	Intentar ingresar código HTML/JavaScript en campos de texto	El sistema sanitiza la entrada	El contenido malicioso no se ejecuta
SEC-04	Validación de roles	Intentar acceder con rol incorrecto	El sistema valida roles correctamente	Acceso restringido según rol
SEC-05	Manipulación de datos del gráfico	Intentar modificar datos hardcoded	Los datos se mantienen íntegros	Datos no pueden ser manipulados desde el cliente

Pruebas de Rendimiento

ID	Caso de Prueba	Descripción	Resultado Esperado	Criterio de Aceptación
PERF-01	Tiempo de carga DonationForm	Medir tiempo de carga inicial	El componente se carga en menos de 2 segundos	Tiempo de carga < 2s
PERF-02	Tiempo de carga DonacionesDashboard	Medir tiempo de carga del dashboard	Dashboard se carga en menos de 3 segundos	Tiempo de carga < 3s
PERF-03	Rendimiento del gráfico	Medir tiempo de generación del gráfico	Chart.js genera el gráfico en menos de 1 segundo	Tiempo de renderizado < 1s
PERF-04	Respuesta del formulario	Medir tiempo desde envío hasta respuesta	La respuesta se recibe en menos de 3 segundos	Tiempo de respuesta < 3s

Pruebas de Accesibilidad

ID	Caso de Prueba	Descripción	Resultado Esperado	Criterio de Aceptación
ACC-01	Navegación con teclado	Navegar por el formulario usando solo el teclado	Todos los campos son accesibles mediante teclado	Navegación sin ratón es posible
ACC-02	Compatibilidad con lectores de pantalla	Probar componentes con lector de pantalla	El lector identifica correctamente todos los elementos	Todos los elementos tienen etiquetas accesibles
ACC-03	Contraste de colores	Revisar contraste entre texto y fondo	El contraste cumple con estándares WCAG	Proporción de contraste según WCAG AA o superior
ACC-04	Etiquetas de formulario	Verificar asociación label-input	Todas las etiquetas están correctamente asociadas	Labels asociados con sus inputs correspondientes

Defectos y Mejoras Sugeridas

Posibles Defectos Identificados

1. **Falta de validación de envío del formulario:** El DonationForm no tiene implementado el manejo del evento submit.
2. **Datos hardcoded en DonacionesDashboard:** Los datos contables están codificados directamente en el componente.
3. **Falta de manejo de errores:** No hay manejo de errores para la carga de datos o fallas de Chart.js.
4. **Validación de cantidad insuficiente:** Solo valida que sea mayor a 1, pero no valida límites superiores.
5. **Falta de loading states:** No hay indicadores de carga durante procesos asíncronos.

Mejoras Sugeridas

1. Implementar manejo del formulario de donación

javascript

```
const [donation, setDonation] = useState({
  nombre: '',
  email: '',
  cantidad: '',
  moneda: 'COP',
  mensaje: ''
});

const [isLoading, setIsLoading] = useState(false);

const handleChange = (e) => {
  setDonation({
    ...donation,
    [e.target.name]: e.target.value
  });
};

const handleSubmit = async (e) => {
  e.preventDefault();
  setIsLoading(true);

  try {
    const response = await axios.post('/api/donaciones', donation, {
      headers: {
        'Authorization': `Bearer ${token}`
      }
    });
  }

  if (response.status === 200) {
    Swal.fire({
      title: 'Donación exitosa',
      text: 'Gracias por tu generosa donación',
      icon: 'success',
      confirmButtonText: 'Aceptar'
    });

    // Limpiar formulario
    setDonation({
      nombre: '',
      email: '',
      cantidad: '',
      moneda: 'COP',
      mensaje: ''
    });
  }
};
```



```

    });
  }
} catch (error) {
  Swal.fire({
    title: 'Error',
    text: 'No se pudo procesar la donación',
    icon: 'error',
    confirmButtonText: 'Aceptar'
  });
} finally {
  setIsLoading(false);
}
};

```

2. Conectar DonacionesDashboard con API

javascript

```

const [donationsData, setDonationsData] = useState([]);
const [loading, setLoading] = useState(true);
const [error, setError] = useState(null);

useEffect(() => {
  const fetchDonations = async () => {
    try {
      const response = await axios.get('/api/donaciones/dashboard', {
        headers: {
          'Authorization': `Bearer ${token}`
        }
      });
    } catch (err) {
      setError('Error al cargar datos de donaciones');
      console.error(err);
    } finally {
      setLoading(false);
    }
  };

  fetchDonations();
}, [token]);

```

3. Mejorar validaciones del formulario

javascript

```
const validateDonation = () => {
  const errors = {};

  if (!donation.nombre.trim()) {
    errors.nombre = 'El nombre es requerido';
  }

  if (!donation.email.trim()) {
    errors.email = 'El email es requerido';
  } else if (!/\S+@\S+\.\S+/.test(donation.email)) {
    errors.email = 'El formato del email no es válido';
  }

  if (!donation.cantidad || donation.cantidad <= 0) {
    errors.cantidad = 'La cantidad debe ser mayor a 0';
  } else if (donation.cantidad > 1000000) {
    errors.cantidad = 'La cantidad máxima es $1,000,000';
  }

  return errors;
};
```

4. Añadir indicadores de carga

javascript

// En el botón de envío del formulario

```
<button
  type="submit"
  className="btn btn-outline-light"
  disabled={isLoading}
>
  {isLoading ? (
    <>
      <span className="spinner-border spinner-border-sm me-2" role="status" aria-hidden="true">
        Procesando...
      </>
    </>
  ) : (
    'Donar'
  )}
</button>
```

// En el dashboard

```
{loading ? (
  <div className="text-center">
    <div className="spinner-border" role="status">
      <span className="visually-hidden">Cargando...</span>
    </div>
  </div>
) : error ? (
  <div className="alert alert-danger" role="alert">
    {error}
  </div>
) : (
  // Contenido del dashboard
)}
```

5. Implementar manejo de sesión mejorado

javascript

```
const handleLogout = () => {
  Swal.fire({
    title: '¿Cerrar sesión?',
    text: 'Se cerrará tu sesión actual',
    icon: 'question',
    showCancelButton: true,
    confirmButtonText: 'Sí, cerrar',
    cancelButtonText: 'Cancelar'
  }).then((result) => {
    if (result.isConfirmed) {
      localStorage.removeItem('auth-token');
      localStorage.removeItem('rol');
      navigate('/login');
    }
  });
};
```

Conclusiones

Los componentes del módulo de Donaciones cumplen con las funcionalidades básicas requeridas para el registro de donaciones y la visualización administrativa. Sin embargo, se recomienda implementar las mejoras sugeridas para:

- Fortalecer la seguridad y validación de datos
- Mejorar la experiencia del usuario con indicadores de carga
- Conectar con APIs reales en lugar de datos hardcodeados
- Implementar manejo robusto de errores
- Añadir validaciones más completas

La implementación de estas mejoras garantizará un módulo más robusto, seguro y escalable para el manejo de donaciones en el sistema CDMI.

Entorno de Pruebas Recomendado

Navegadores a probar:

- Google Chrome (última versión)
- Mozilla Firefox (última versión)
- Microsoft Edge (última versión)

- Safari (última versión)

Dispositivos a probar:

- Escritorio (resoluciones 1920x1080, 1366x768)
- Tableta (iPad, resolución 768x1024)
- Móvil (iPhone, Galaxy, resoluciones 375x667, 414x896)

Herramientas recomendadas:

- React Testing Library para pruebas unitarias
- Cypress para pruebas de extremo a extremo
- Lighthouse para evaluación de rendimiento
- WAVE o axe para pruebas de accesibilidad
- Jest para pruebas de lógica de negocio