

# Documento de cobertura del testing.

---

*Proyecto: Xitomota*

Revisión: [2]

26/11/2025

# Contenido

<b>Ficha del documento.....</b>	<b>3</b>
<b>1. Objetivo.....</b>	<b>3</b>
<b>2. Alcance de las pruebas.....</b>	<b>3</b>
Archivos cubiertos:.....	3
<b>Resumen de cobertura de archivos cubiertos.....</b>	<b>7</b>
<b>5. Conclusiones.....</b>	<b>7</b>
<b>6. Evidencias.....</b>	<b>7</b>

# Ficha del documento

Fecha	Revisión	Autor	Modificación
30-10-2025	/1/	Brayan Ahumada Cristian Ormazabal	Versión de testing
26-11-2025	/2/	Brayan Ahumada Cristian Ormazabal	Versión de testing

## 1. Objetivo

Describir el nivel de cobertura alcanzado por las pruebas unitarias implementadas con Jest, indicando las áreas del código verificadas y las no cubiertas.

## 2. Alcance de las pruebas

- **Framework:** Jest
- **Tipo de pruebas:** Unitarias / Integración
- **Entorno:**
  - Node.js v24.11.0
  - React 19
  - Vite 7
  - Jest 30 con entorno jsdom

### Archivos cubiertos:

- src / componentes / footer.jsx
- src / componentes / Header.jsx
- src / Paginas / Autenticacion / IniciarSesion.jsx
- src / Paginas / Autenticacion / Registrouuario.jsx
- src / Paginas / Blogs / Blogs.jsx
- src / Paginas / Carrito / Carrito.jsx
- src / Paginas / Categorías / Buzos / Buzos.jsx
- src / Paginas / Categorías / Camisas / Camisas.jsx
- src / Paginas / Categorías / Chaqueta / Chaqueta.jsx
- src / Paginas / Categorías / Pantalones / Pantalones.jsx

- src / Paginas / Categorías / Polerones / Polerones.jsx
- src / Paginas / Categorías / Poleras / Poleras.jsx
- src / Paginas / Checkout / Checkout.jsx
- src / Paginas / Checkout / PagoFallido.jsx
- src / Paginas / Checkout / PagoRealizado.jsx
- src / Paginas / Contacto / Contacto.jsx
- src/ Paginas / Home / Home.jsx
- src/ Paginas / Nosotros / Nosotros.jsx
- src/ Paginas / Ofertas / Ofertas.jsx
- src/ Paginas / Productos / DetalleProducto.jsx
- src/ Paginas / Productos / Productos.jsx
- src/ Paginas / Productos / TarjetaProducto.jsx

Fragmento de código de ejemplo de Checkout.test.jsx

```
it("debe llamar a la API de 'crearOrden' y redirigir al hacer clic en Pagar", async () => {  
  axios.post.mockResolvedValue({  
    data: { id: 123, nombre: 'Pedro' }  
  });  
  
  render(  
    <MemoryRouter initialEntries={['/checkout']}>  
      <Routes>  
        <Route path="/checkout" element={<Checkout />} />  
        <Route path="/gracias" element={<div>¡Pago Exitoso!</div>} />  
      </Routes>  
    </MemoryRouter>  
  );  
});
```

## Fragmento de código de ejemplo de Header.test.jsx

```
it('correcto href', async () => {
  const user = userEvent.setup();
  render(
    <MemoryRouter>
      <Header />
    </MemoryRouter>
  );
  expect(screen.getByRole('link', { name: /XITOMOTA/i })).getAttribute('href').toBe('/');
  expect(screen.getByRole('link', { name: /Inicio/i })).getAttribute('href').toBe('/');
  expect(screen.getByRole('link', { name: /Productos/i })).getAttribute('href').toBe('/productos');
  expect(screen.getByRole('link', { name: /Ofertas/i })).getAttribute('href').toBe('/ofertas');
  expect(screen.getByRole('link', { name: /Nosotros/i })).getAttribute('href').toBe('/nosotros');
  expect(screen.getByRole('link', { name: /Blogs/i })).getAttribute('href').toBe('/blogs');
  expect(screen.getByRole('link', { name: /Contacto/i })).getAttribute('href').toBe('/contacto');

  expect(screen.getByRole('link', { name: /Iniciar Sesión/i })).getAttribute('href').toBe('/login');
  expect(screen.getByRole('link', { name: /Carrito/i })).getAttribute('href').toBe('/carrito');

  const catButton = screen.getByRole('button', { name: /Categorías/i });
  await user.click(catButton);

  const polerasLink = await screen.findByRole('link', { name: /Poleras/i });
  expect(polerasLink.getAttribute('href')).toBe('/categorias/poleras');
```

## Fragmento de código de ejemplo de Contacto.test.jsx

```
test('al enviar el formulario se muestra alert y se limpian los campos', () => {
  const nombreInput = screen.getByLabelText(/Nombre Completo/i);
  const correoInput = screen.getByLabelText(/Correo/i);
  const comentarioInput = screen.getByLabelText(/Comentario/i);
  const enviarButton = screen.getByRole('button', { name: /Enviar/i });

  fireEvent.change(nombreInput, { target: { value: 'Cristian' } });
  fireEvent.change(correoInput, { target: { value: 'test@mail.com' } });
  fireEvent.change(comentarioInput, { target: { value: 'Hola mundo' } });

  fireEvent.click(enviarButton);

  expect(window.alert).toHaveBeenCalledWith('Mensaje enviado (simulación)');
  expect(nombreInput.value).toBe('');
  expect(correoInput.value).toBe('');
  expect(comentarioInput.value).toBe('');
});
```

### Fragmento de código de ejemplo de IniciarSesion.test.jsx

```
test('navega y guarda en localStorage con credenciales correctas', async () => {
  axios.post.mockResolvedValue({ data: { email: 'admin@example.com', rol: 'Administrador' } });

  render(
    <MemoryRouter>
      <IniciarSesion />
    </MemoryRouter>
  );

  fireEvent.change(screen.getByLabelText(/Email/i), { target: { value: 'admin@example.com' } });
  fireEvent.change(screen.getByLabelText(/Password/i), { target: { value: 'admin' } });
  fireEvent.click(screen.getByRole('button', { name: /Entrar/i }));

  // Espera que localStorage se actualice
  await screen.findByText(/Iniciar Sesión/i); // solo para esperar re-render

  expect(localStorage.getItem('usuarioLogueado')).toBe('admin@example.com');
  expect(localStorage.getItem('rolUsuario')).toBe('Administrador');
  expect(mockNavigate).toHaveBeenCalledWith('/admin');
});
```

### Fragmento de código de ejemplo de RegistroUsuario.test.jsx

```
test('debe mostrar un error si las contraseñas no coinciden', async () => {
  render(<RegistroUsuario />);

  await userEvent.type(screen.getByTestId('nombre-input'), 'Test');
  await userEvent.type(screen.getByTestId('apellidos-input'), 'User');
  await userEvent.type(screen.getByTestId('email-input'), 'test@user.com');
  await userEvent.type(screen.getByTestId('password-input'), '123456');
  await userEvent.type(screen.getByTestId('confirm-password-input'), 'abcdef');

  fireEvent.click(screen.getByRole('button', { name: /Registrar/i }));

  expect(await screen.findByText(/Las contraseñas no coinciden/i)).toBeInTheDocument();
  expect(axios.post).not.toHaveBeenCalled();
});
```

## Resumen de cobertura de archivos cubiertos

(Estos datos los obtienes del resumen que Jest genera en consola o del archivo `coverage-summary.json`)

Archivo/modulo

Métrica	Porcentaje	Objetivo	Estado
Statements	86,22%	$\geq 80\%$	Cumple
Branches	70,39%	$\geq 70\%$	Cumple
Functions	92,37%	$\geq 80\%$	Cumple
Lines	86,68%	$\geq 80\%$	Cumple

## 5. Conclusiones

- La cobertura global cumple los objetivos mínimos definidos.
- Se recomienda aumentar las pruebas en los módulos con menor cobertura.
- Las pruebas han validado los escenarios principales y críticos del sistema.

# 6. Evidencias

Capturas del reporte HTML ([coverage/lcov-report/index.html](#)) General

## All files

86.22% Statements 388/450 70.39% Branches 187/152 92.37% Functions 189/118 86.68% Lines 384/443

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

## All files

86.22% Statements 388/450 70.39% Branches 187/152 92.37% Functions 189/118 86.68% Lines 384/443

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File	Statements	Branches	Functions	Lines
Paginas/Autenticacion	<div><div></div></div> 100%61/61	<div><div></div></div> 100%19/19	<div><div></div></div> 100%11/11	<div><div></div></div> 100%61/61
Paginas/Blogs	<div><div></div></div> 100%2/2	<div><div></div></div> 100%0/0	<div><div></div></div> 100%1/1	<div><div></div></div> 100%2/2
Paginas/Carrito	<div><div></div></div> 80.88%55/68	<div><div></div></div> 70%14/20	<div><div></div></div> 82.35%14/17	<div><div></div></div> 83.07%54/65
Paginas/Categorias/Buzos	<div><div></div></div> 100%12/12	<div><div></div></div> 50%1/2	<div><div></div></div> 100%5/5	<div><div></div></div> 100%12/12
Paginas/Categorias/Camisas	<div><div></div></div> 100%12/12	<div><div></div></div> 50%1/2	<div><div></div></div> 100%5/5	<div><div></div></div> 100%12/12
Paginas/Categorias/Chaquetas	<div><div></div></div> 100%12/12	<div><div></div></div> 50%1/2	<div><div></div></div> 100%5/5	<div><div></div></div> 100%12/12
Paginas/Categorias/Pantalones	<div><div></div></div> 100%12/12	<div><div></div></div> 50%1/2	<div><div></div></div> 100%5/5	<div><div></div></div> 100%12/12
Paginas/Categorias/Poleras	<div><div></div></div> 100%12/12	<div><div></div></div> 50%1/2	<div><div></div></div> 100%5/5	<div><div></div></div> 100%12/12
Paginas/Categorias/Polerones	<div><div></div></div> 100%12/12	<div><div></div></div> 50%1/2	<div><div></div></div> 100%5/5	<div><div></div></div> 100%12/12
Paginas/Checkout	<div><div></div></div> 79.48%62/78	<div><div></div></div> 70.37%19/27	<div><div></div></div> 84.21%16/19	<div><div></div></div> 79.48%62/78
Paginas/Contacto	<div><div></div></div> 100%17/17	<div><div></div></div> 100%0/0	<div><div></div></div> 100%6/6	<div><div></div></div> 100%17/17
Paginas/Home	<div><div></div></div> 100%1/1	<div><div></div></div> 100%0/0	<div><div></div></div> 100%1/1	<div><div></div></div> 100%1/1
Paginas/Nosotros	<div><div></div></div> 100%1/1	<div><div></div></div> 100%0/0	<div><div></div></div> 100%1/1	<div><div></div></div> 100%1/1
Paginas/Oferas	<div><div></div></div> 73.33%11/15	<div><div></div></div> 50%2/4	<div><div></div></div> 100%5/5	<div><div></div></div> 73.33%11/15
Paginas/Productos	<div><div></div></div> 83.15%79/95	<div><div></div></div> 70.68%41/58	<div><div></div></div> 100%14/14	<div><div></div></div> 84.04%79/94
componentes	<div><div></div></div> 67.5%27/40	<div><div></div></div> 50%6/12	<div><div></div></div> 76.92%10/13	<div><div></div></div> 64.86%24/37



File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	86.22	70.39	92.37	86.68	
Paginas/Autenticacion	100	100	100	100	
IniciarSesion.jsx	100	100	100	100	
RegistroUsuario.jsx	100	100	100	100	
Paginas/Blogs	100	100	100	100	
Blogs.jsx	100	100	100	100	
Paginas/Carrito	80.88	70	82.35	83.07	
Carrito.jsx	80.88	70	82.35	83.07	37-38,54,70,79,84-92,138
Paginas/Categorias/Buzos	100	50	100	100	
Buzos.jsx	100	50	100	100	49
Paginas/Categorias/Camisas	100	50	100	100	
Camisas.jsx	100	50	100	100	51
Paginas/Categorias/Chaquetas	100	50	100	100	
Chaquetas.jsx	100	50	100	100	51
Paginas/Categorias/Pantalones	100	50	100	100	
Pantalones.jsx	100	50	100	100	49
Paginas/Categorias/Poleras	100	50	100	100	
Poleras.jsx	100	50	100	100	49
Paginas/Categorias/Polerones	100	50	100	100	
Polerones.jsx	100	50	100	100	49
Paginas/Checkout	79.48	70.37	84.21	79.48	
Checkout.jsx	69.76	53.33	100	69.76	31-33,41-42,54-55,65-76
PagoFallido.jsx	83.33	100	60	83.33	18,27
PagoRealizado.jsx	95.65	83.33	83.33	95.65	146
Paginas/Contacto	100	100	100	100	
Contacto.jsx	100	100	100	100	
Paginas/Home	100	100	100	100	
Home.jsx	100	100	100	100	
Paginas/Nosotros	100	100	100	100	
Nosotros.jsx	100	100	100	100	
Paginas/Ofertas	73.33	50	100	73.33	
Ofertas.jsx	73.33	50	100	73.33	45-46,53-54
Paginas/Productos	83.15	70.68	100	84.04	
DetalleProducto.jsx	81.48	61.76	100	83.01	54-55,58-59,71-75,100
Producto.jsx	90	100	100	90	15
TarjetaProducto.jsx	83.87	83.33	100	83.87	34-35,38-39,53
componentes	67.5	50	76.92	64.86	
Footer.jsx	100	100	100	100	
Header.jsx	66.66	50	75	63.88	32-36,43-49,56,109
Test Suites: 22 passed, 22 total					
Tests: 73 passed, 73 total					
Snapshots: 0 total					
Time: 4.702 s					
Ran all test suites.					