



Software Inventory

PLAN MAESTRO DE PRUEBAS

Proyecto: Software Inventory

Versión: 1

SENA – Servicio Nacional de Aprendizaje
Análisis y Desarrollo de Sistemas de Información

Ficha: 2395871 – G2

Bogotá – 2023

Historia de revisiones

Versión	Autores	Descripción	Fecha
1	Sergio Galindo	Creación del documento	Marzo 2023

Contenido

INTRODUCCIÓN	4
OBJETIVO	4
REFERENCIA.....	4
ALCANCE.....	4
CONTEXTO DE LAS PRUEBAS	5
PROYECTO	5
OBJETIVOS DE LAS PRUEBAS	5
ESTRATEGIA DE PRUEBAS.....	6
DISEÑO DE CASOS DE PRUEBA.....	6
TÉCNICA DE CAJA NEGRA	6
NIVEL DE PRUEBAS POR COMPONENTES O PRUEBAS UNITARIAS.....	7
NIVEL DE PRUEBAS DE INTEGRACIÓN	10
NIVEL DE PRUEBAS DEL SISTEMA.....	12
PLANIFICACIÓN	14
ENTORNO Y CONFIGURACIÓN DE LAS PRUEBAS	14
CRITERIOS DE INICIO	14
CRITERIOS DE APROBACION / RECHAZO.....	14
EQUIPO DE PRUEBAS Y RESPONSABILIDADES.....	15
RESULTADOS	16

INTRODUCCIÓN

OBJETIVO

Este plan maestro de pruebas sirve como guía proporcionando información y el marco requerido para planificar y desarrollar las actividades del proceso de pruebas del Sistema.

REFERENCIA

- ISO 29119
- ISO 9126
- ISO/IEC 25010 (Describe los tipos y niveles de pruebas de software).
- Especificación de Requerimientos del Proyecto

ALCANCE

El presente documento contiene las directrices para la estrategia, planeación, ejecución y manejo de los resultados de la disciplina de pruebas para los Sistemas de Información que estén en etapa de desarrollo.

CONTEXTO DE LAS PRUEBAS

PROYECTO

La aplicación web software inventory es un sistema de Inventario y ventas que consta de los siguientes módulos.

Perfil 1:

- Administrador

Módulos:

- Gestionar Ventas
- Gestionar Usuarios
- Gestionar Clientes
- Gestionar Categorías
- Gestionar Proveedores
- Gestionar Productos

Perfil 2:

- Vendedor

Módulos:

- Gestionar Ventas
- Gestionar Clientes

Perfil 3:

- Compra

Módulos:

- Gestionar Categorías
- Gestionar Proveedores
- Gestionar Productos

OBJETIVOS DE LAS PRUEBAS

En estos módulos se realizarán pruebas para validar:

- La visualización de los datos, ingresados o modificados.
- La operación de los servicios, para dar respuesta a los productos del sistema.
- La respuesta de las funciones de cada módulo.

- La secuencia lógica de las funcionalidades.

ESTRATEGIA DE PRUEBAS

Las pruebas para la aplicación web incluirán los siguientes niveles de pruebas.

- Pruebas de Componentes / Unitarias

Las Pruebas de Componentes son la primera fase de las pruebas dinámicas y se realizan sobre cada módulo o componente del software que se pueda probar de forma independiente.

El objetivo es comprobar que el componente está correctamente codificado, soportando el ingreso de datos erróneos o inesperados y demostrando así la capacidad de tratar errores de manera controlada.

- Pruebas de Integración

Antes de las pruebas de integración, los componentes tuvieron que haber pasado sus pruebas unitarias, por lo que el enfoque ahora es sobre el flujo de control entre los módulos.

- Pruebas de Sistema

Reúne todos los niveles anteriormente mencionados en una prueba total de todas las funcionalidades más relevantes del sistema, dando así una versión completa del objetivo general, una versión lo más semejante al ambiente productivo.

DISEÑO DE CASOS DE PRUEBA

De acuerdo a la estrategia de pruebas, los casos de prueba son ejecutados por componentes, mediante el tipo de prueba dinámica que se realiza mientras el código está en ejecución. Se aplicarán casos de pruebas de tipo dinámicas, las cuales comprenderán de las siguientes técnicas y niveles.

TÉCNICA DE CAJA NEGRA

La técnica de caja negra, se enfoca en probar el sistema sin tomar en cuenta la estructura interna del mismo, su objetivo es validar que las salidas sean las esperadas. Hay varios tipos de pruebas de caja negra que son posibles, pero se consideraran las dos fundamentales, prueba funcional y prueba no funcional. De acuerdo con las técnicas a continuación indicadas se estarán realizando pruebas funcionales.

- Casos de Uso

- Partición de Equivalencia

NIVEL DE PRUEBAS POR COMPONENTES O PRUEBAS UNITARIAS

- Casos de uso del Perfil 1 (Administrador)
 - Caso de Uso 1

Caso de uso	Gestión de usuarios - Creación		
Actores	Administrador		
Descripción	Se muestra en pantalla un formulario donde se registran los datos del usuario a ingresar, para que este pueda acceder a la aplicación.		
Precondición	El administrador debe haber iniciado sesión en la aplicación. El usuario no debe estar registrado en la base de datos.		
Secuencia Normal	Paso	Acción	
		Actor	Sistema
	1	Ingresar Nombre	
	2		Valida que el campo no este Null
	3	Ingresar Apellido	
	4		Valida que el campo no este Null
	5	Ingresar Login	
	6		Valida que el campo no este Null
	7	Ingresar Contraseña	
	8		Valida que el campo no este Null
	9	Ingresar Rol	
	10		Valida que se seleccione un unico rol (Administrador, Cajero, Bodega)
	11	Ingresar Estado	
	12		Valida que se seleccione Activo o Inactivo
	13		Verifica que el usuario no este registrado
	14		Asigna codigo unico automatico al usuario
	15		Almacena informacion en la BD
	16		Muestra informacion de usuario registrado
Escenarios Alternativos	Paso	Acción	
		Actor	Sistema
	1		
	2		Muestra en pantalla un mensaje de error "El campo no puede ser nulo, intente nuevamente". Regresa al paso 1
	3		
	4		Muestra en pantalla un mensaje de error "El campo no puede ser nulo, intente nuevamente". Regresa al paso 3
	5		
	6		Muestra en pantalla un mensaje de error "El campo no puede ser nulo, intente nuevamente". Regresa al paso 5
	7		
	8		Muestra en pantalla un mensaje de error "El campo no puede ser nulo, intente nuevamente". Regresa al paso 7
	9		
	10		Muestra en pantalla un mensaje de error "El campo no puede ser nulo, intente nuevamente". Regresa al paso 9
	11		
	12		Muestra en pantalla un mensaje de error "Debe seleccionar un estado, intente nuevamente". Regresa al paso 11
	13		Muestra un mensaje de error "El usuario ya se encuentra registrado". Solicita nuevamente los datos del usuario. Regresa al paso 1.
Postcondición	El usuario queda creado correctamente en la base de datos.		

- Partición de Equivalencia del Caso de Uso 1

Probar la creación de un nuevo perfil de usuario con la misma información con la que se creó previamente un perfil anterior, tiene como salida un 'Error en esta consulta', por lo tanto, probar con cualquier valor de cada conjunto de la clase es suficiente para probar

el escenario anterior, preconditionando de esta manera al usuario que no debe crear dos productos con el mismo nombre.

○ Caso de Uso 2

Caso de uso		Crear proveedores	
Actores		Administrador y bodega	
Descripción		por medio de este caso de uso el administrador y la bodega pueden crear la existencia de un proveedor	
Precondición		el administrador debe estar previamente logeado con el usuario administrador	
Secuencia Normal	Paso	Acción	
		Administrador y bodega	Sistema
	1	ingresa al modulo proveedores	
	2		el sistema valida permisos de ingreso
	3	el usuario selecciona el campo crear nuevo registro	
	4		el sistema genera un formulario en blanco
	5	el usuario llena los datos del nuevo proveedor	
	6		el sistema valida la informacion ingresada
	7	el usuario guarda los cambios	
Escenarios Alternativos	Paso	Acción	
		Administrador y bodega	Sistema
	1	ingresa de manera erronea	
	2		no se concede el ingreso
	3		el sistema genera un mensaje de error
	4		el sistema genera un mensaje con la correccion a realizar
	5		el sistema vuelve al ingreso a modulo proveedores
	6	ingresa informacion del nuevo registro de manera erronea	
	7		el sistema muestra un mensaje con la correccion a realizar
	8	el usuario realiza la correccion y guarda nuevamente el registro	
Postcondición	9		el sistema vuelve y valida la informacion
	10		se realiza el guardado del nuevo registro
Postcondición		el sistema permite solo la creacion de nuevos proveedores, este guardado solo se hace efectivo al pulsar la opcion guardar	

○ Partición de Equivalencia del Caso de Uso 2

Probar la creación de un nuevo Proveedor con la misma información con la que se creó previamente un proveedor anterior, tiene como salida un 'Error en esta consulta', por lo tanto, probar con cualquier valor de cada conjunto de la clase es suficiente para probar el escenario anterior, preconditionando de esta manera al usuario que no debe crear dos productos con el mismo nombre.

○ Caso de Uso 3

Caso de uso	iniciar venta		
Actores	cajero		
Descripción	por medio de este caso de uso el sistema inicia la venta de los productos seleccionados		
Precondición	el cajero debe estar logeado al sistema		
Secuencia Normal	Paso	Acción	
		Actor	Sistema
	1	ingresa el nombre de usuario del cajero	
	2		el sistema valida el nombre del usuario solicitado
	3	el cajero solicita informacion al cliente	
	4		el sistema valida la informacion brindada por el cliente
	5	el cajero inicia la facturacion del los productos	
	6		el sistema va guardando los productos facturados
	7	el cajero finaliza la facturacion	
	8		el sistema crea factura de ventas
	Paso	Acción	
		Actor	Sistema
	1	ingreso de manera erroneo	
	2		el sistema genera un mensaje de error
Escenarios Alternativos	3		no se concede el ingreso
	4		el sistema genera un mensaje con la correccion a realizar
	5		el sistema vuelve al ingreso a modulo usuarios
	6	el cajero no cuenta con usuario	
	7		el sistema solicita notificar al administrador
	8		
	9		
	10		
Postcondición	el sistema permite dar inicio una venta y a la creacion de facturas		

Del conjunto de casos de pruebas definidos, deberán identificarse aquellos que formarán parte del conjunto de pruebas que deberán realizarse para asegurar el correcto despliegue de la aplicación.

NIVEL DE PRUEBAS DE INTEGRACIÓN

Las pruebas de integración verifican que los módulos del sistema funcionan adecuadamente en conjunto. Para ello mediante un caso de prueba dinámica con técnica de caja negra se demostrará la funcionalidad entre el módulo de crear un producto, un cliente y una venta.

Con un enfoque ascendente se describe la prueba de integración de la siguiente manera:

Identificador	Nombre
Prueba de Integración	Crear un Cliente y crear una Venta
Requerimiento de prueba	
Creación Cliente y Creación Venta	
Precondiciones	
<ul style="list-style-type: none">• No se crea una venta si no se crea de forma previa un cliente.• No se crea una venta si no se crea de forma previa un producto, modulo que solo puede ser manipulado por un perfil Administrador o Compra.• Sólo el perfil Administrador y Vendedor puede crear un cliente y consecutivamente una venta.	
Objetivo	
<ul style="list-style-type: none">• Permitir al usuario con rol de Administrador o Compra crear o agregar un producto con sus respectivos atributos.• Permitir al usuario con rol de Administrador o Vendedor crear o agregar un cliente.• Permitir al usuario con rol de Administrador o Vendedor crear una venta.	
Flujo Principal o Pasos	

1. El caso de uso inicia cuando el usuario ejecuta la aplicación mediante el login.
2. El módulo presenta un menú al lado izquierdo de la pantalla, allí se da clic en el botón llamado Categorías y se accede para crear el nombre de una nueva categoría.
3. Dar clic en el botón 'Guardar'.
4. Si la ejecución es correcta el módulo manda el mensaje "Categoría Registrada".
5. En el menú lateral presenta el botón llamado 'Productos', dar clic y acceder a la tabla que permite visualizar y administrar productos.
6. Dar clic en el botón 'Nuevo' el cual nos lleva al formulario que nos permite ingresar la información del producto que quiere ingresar al inventario.
7. Llenar los campos que requiere de forma obligatoria el formulario tipeando la referencia del producto, eligiendo su correspondiente categoría y proveedor, tipeamos el stock en números, el precio de compra, el precio de venta y dar clic al botón 'Guardar'.
8. Si la ejecución es correcta el módulo manda el mensaje "Producto Registrado".
9. En el menú lateral presenta el botón llamado 'Clientes', dar clic y acceder a la tabla que permite visualizar y administrar clientes.
10. Dar clic en el botón 'Nuevo' el cual nos lleva al formulario que nos permite ingresar la información del cliente.
11. Llenar los campos que requiere de forma obligatoria el formulario tipeando la identificación, nombre, dirección, teléfono y email y dar clic al botón 'Guardar'.
12. Si la ejecución es correcta el módulo manda el mensaje "Cliente Registrado".
13. Para el caso de uso de creación de una venta se da clic en el botón de 'Venta' el cual proporciona inicialmente un buscador en el cual se parametriza escribiendo el documento del cliente.
14. Se debe digitar el código del producto.
15. Cuando el producto sea identificado se debe escribir la cantidad de productos y dar clic al botón "Agregar".
16. El producto será adicionado al listado de productos por vender.
17. Si se desea quitar productos es necesario dar clic al botón "Eliminar".
18. Para finalizar y guardar la venta se debe dar clic al botón "Generar Venta".
19. Al agregarse la venta sale en pantalla un mensaje que dice '¡Venta Generada!'
20. Fin de prueba de integración.

Resultado esperado de la ejecución del caso de prueba

- La nueva categoría se visualiza en la tabla del botón categorías, en la sección 'Categorías'.
- El producto inventariado, agregado después de los pasos anteriores, se puede visualizar en el panel de control en la sección 'Productos'.
- El cliente creado puede ser consultado o visualizado en la lista de clientes.
- La venta agregada se guarda de manera correcta en la base de datos.

NIVEL DE PRUEBAS DEL SISTEMA

Se diseña una prueba de sistema dinámica para verificar que la integración de los componentes del sistema realiza las funciones de manera apropiada. Para esto se realiza prueba de ejecución desde el login y siguiendo un flujo de procesos hasta terminar con el logout del usuario.

PRUEBA DE SEGURIDAD

Se verifica mediante esta prueba que los mecanismos de protección evitarán accesos inapropiados al sistema. El caso de prueba que llevará a cabo la validación de seguridad se describe de la siguiente manera.

Identificador	Tipo de Prueba
Prueba de Seguridad	Prueba Dinámica
Técnica de Prueba	Objetivo
Técnica de Caja Negra	Probar la seguridad del acceso al sistema por Login
Precondiciones	
<ul style="list-style-type: none">Las credenciales para acceder como Administrador son creadas desde el momento que se crea la base de datos.Al perfil de vendedor y compra se le asignará un usuario y una contraseña que el Administrador previamente ha creado al momento de registrar los datos de dicho perfil.Sólo el perfil de Administrador puede crear perfiles de usuarios.	
Flujo Principal o Pasos	
<ol style="list-style-type: none">El caso de uso inicia cuando un usuario ejecuta la aplicación mediante el login.Para comprobar la seguridad del sistema se hacen ingresos de datos errados, se ingresa una palabra aleatoria tanto para el campo de escritura de Usuario como de Contraseña.Si el código de forma interna está estructurado para evitar los ingresos inapropiados, entonces validará que los datos ingresados de forma errada no corresponden al código que trabaja de forma interna en el sistema.Si la información es incorrecta el módulo manda el mensaje "Usuario o Contraseña Incorrecta".Fin de prueba de seguridad	

Resultado Esperado

Se valida y se comprueba que efectivamente no se puede ingresar al sistema si los datos del login están errados, se espera un mensaje alerte 'Usuario o Contraseña Incorrecta'

TÉCNICA DE CAJA BLANCA

Para la prueba de seguridad también se implementa la técnica de caja blanca que examina el código interno que valida el ingreso inapropiado de datos en el módulo del login del sistema.

```
index.php x
index.php > ...
1  <?php
2  $alert = '';
3  session_start();
4  if (!empty($_SESSION['active'])) {
5      header('location: sistema/');
6  } else {
7      if (!empty($_POST)) {
8          if (empty($_POST['usuario']) || empty($_POST['contrasena'])) {
9              $alert = '<div class="alert alert-danger" role="alert">
10                 Ingrese su usuario y su clave
11                 </div>';
12                 echo'<script type="text/javascript">
13                 alert("Ingrese su usuario y su clave.");
14                 window.location.href="index.php";
15                 </script>';
16             } else {
17                 require_once "conexion.php";
18                 $user = mysqli_real_escape_string($conectar, $_POST['usuario']);
19                 $clave = md5(mysqli_real_escape_string($conectar, $_POST['contrasena']));
20                 $query = mysqli_query($conectar, "SELECT u.idusuario, u.nombreempleado, u.email,u.usuario,r.idrol,r.
21                 mysqli_close($conectar);
22                 $resultado = mysqli_num_rows($query);
23                 if ($resultado > 0) {
24                     $dato = mysqli_fetch_array($query);
25                     $_SESSION['active'] = true;
26                     $_SESSION['idUser'] = $dato['idusuario'];
27                     $_SESSION['nombreempleado'] = $dato['nombreempleado'];
28                     $_SESSION['email'] = $dato['email'];
29                     $_SESSION['user'] = $dato['usuario'];
30                     $_SESSION['rol'] = $dato['idrol'];
31                     $_SESSION['rol_name'] = $dato['rol'];
32                     header('location: sistema/');
33                 } else {
34                     $alert = '<div class="alert alert-danger" role="alert">
35                         Usuario o Contraseña Incorrecta
36                         </div>';
37                     echo'<script type="text/javascript">
38                     alert("Usuario o Contraseña Incorrecta");
39                     window.location.href="index.php";
40                     </script>';
41                 }
42                 session_destroy();
43             }
44         }
45     }
46 }
47 ?>
48
```

PLANIFICACIÓN

ENTORNO Y CONFIGURACIÓN DE LAS PRUEBAS

Para el proceso de pruebas del proyecto se requiere de la disponibilidad de los siguientes entornos:

- a. Servidor Local instalado en entorno del sistema operativo Windows 10 con conexión a Internet.
 - i. Equipo Marca SAMSUNG Inter(R) Core (TM) i5-3210M CPU @2.50GHz, 12GB RAM, 1T HDD, 1366x768 LCD (60 Hz), Windows 10 Home Single Language versión 1909.
- b. Herramienta para la ejecución de Pruebas de Software:
 - i. CYPRESS versión 11.2.0
- c. Base de Datos SQL. Reside en equipo de servidor de pruebas detallado en el punto a del entorno y configuración de las pruebas.

Todos ellos configurados por el equipo de desarrollo.

CRITERIOS DE INICIO

Aceptación del plan de pruebas. Revisión y aceptación del documento que contiene los casos de pruebas para la certificación del proyecto.

Aceptación de paquetes. Revisión y aceptación de los paquetes de desarrollo, y que este cumpla con las condiciones de aceptación.

Aceptación de ambiente. Revisión y aceptación del ambiente de certificación, y que este cumpla con las condiciones de aceptación.

CRITERIOS DE APROBACION / RECHAZO

Errores Graves: Información crítica presentada erróneamente, información mal registrada en la base de datos, caídas de programas, incumplimiento de objetivos en funciones principales, etc.

Errores Medios (comunes): Errores en documentos impresos que se entregan a personas ajenas a la organización, errores en presentación de datos, incumplimiento de objetivos en funciones secundarias, caídas de programas auxiliares, etc.

Errores Leves: Errores en presentación de datos secundarios, no adecuación a estándares, comportamientos correctos pero diferentes en situaciones similares, dificultades de operación, etc.

EQUIPO DE PRUEBAS Y RESPONSABILIDADES

Nombre	Responsabilidad
Sergio Galindo	Testing de Solución, responsable de la generación del plan de pruebas. Responsable de evaluar las condiciones de término para el proceso de pruebas junto al Analista Funcional.
Juan Lozano	Analista funcional, responsable de la resolución de las incidencias de certificación para los módulos de Proyectos, Revisión y Aprobación

RESULTADOS

El proceso de pruebas finaliza cuando están implementados todos los requerimientos y se encuentran el 100 % de las pruebas ejecutadas y pasadas satisfactoriamente, no deben existir errores críticos.

Para el reporte de los resultados de las pruebas se implementará la herramienta llamada Cypress que nos permitirá visualizar los resultados de cada prueba.

```
Johnny@DESKTOP-MASTER MINGW64 /c/xampp/htdocs/proyecto_sergio
$ npm install
npm notice Beginning October 4, 2021, all connections to the npm registry - including for package installation - must use TLS 1.2 or higher. You are currently using plaintext http to connect. Please visit the GitHub blog for more information: https://github.blog/2021-08-23-npm-registry-deprecating-tls-1-0-tls-1-1/
npm notice Beginning October 4, 2021, all connections to the npm registry - including for package installation - must use TLS 1.2 or higher. You are currently using plaintext http to connect. Please visit the GitHub blog for more information: https://github.blog/2021-08-23-npm-registry-deprecating-tls-1-0-tls-1-1/

added 172 packages in 26s

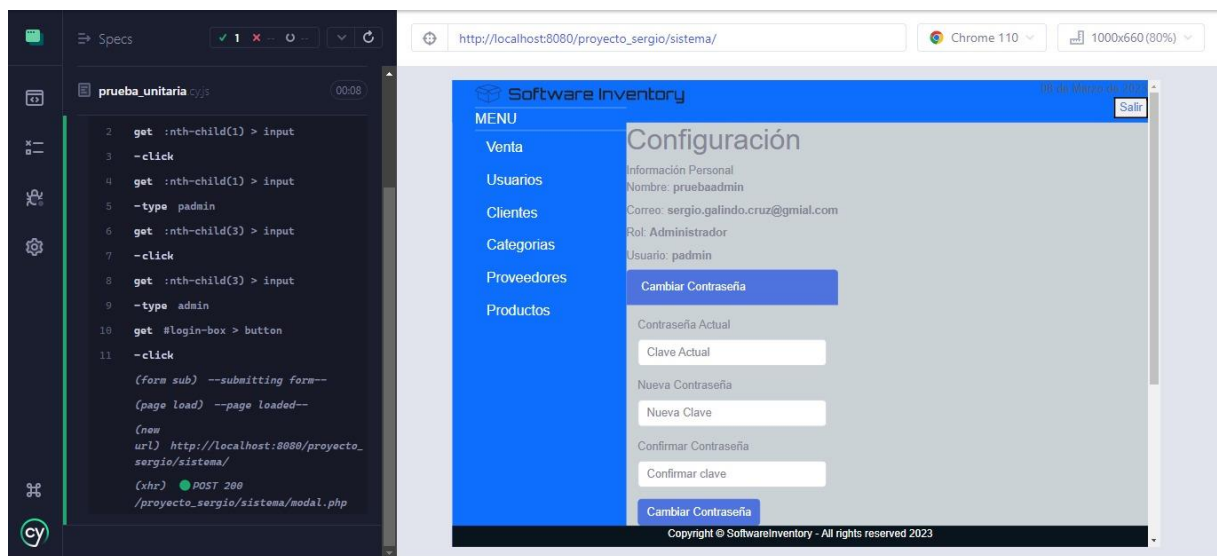
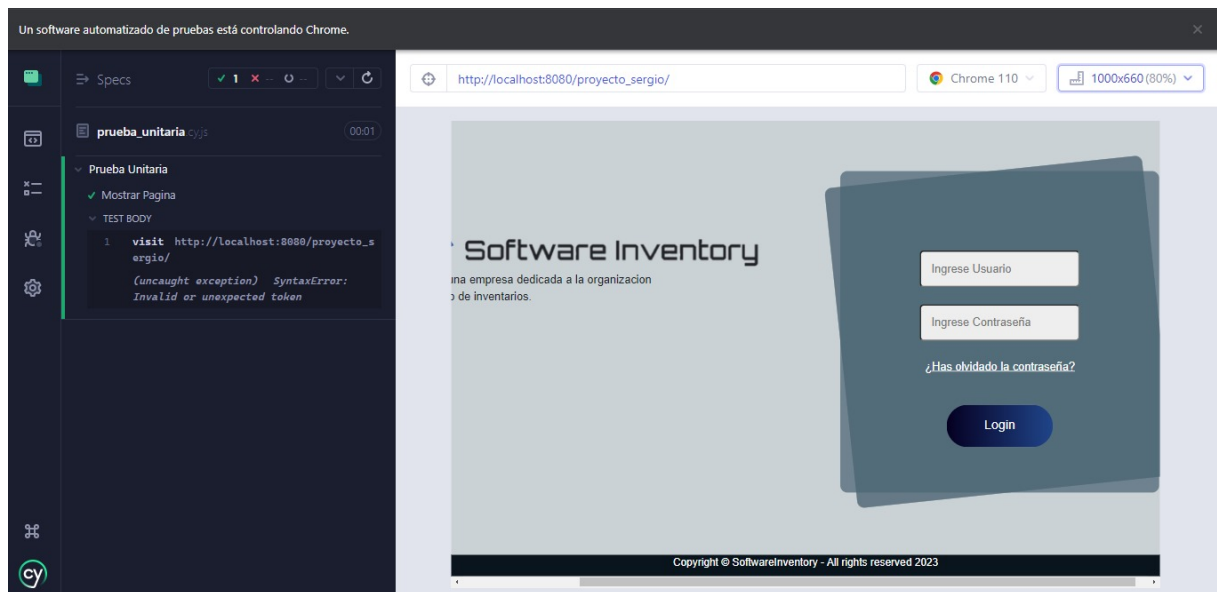
35 packages are looking for funding
  run 'npm fund' for details
```

```
Johnny@DESKTOP-MASTER MINGW64 /c/xampp/htdocs/proyecto_sergio
$ npm run cypress:open

> proyecto_sergio@1.0.0 cypress:open
> cypress open
```

Resultados Nivel de Pruebas Unitarias

```
prueba_unitaria.cy.js X
proyecto_sergio > cypress > e2e > prueba_unitaria.cy.js > ...
1 describe('Prueba Unitaria', () => {
2   it('Mostrar Pagina y Login', () => {
3
4     // Corrige eventos no detectados por Cypress:
5     cy.on("uncaught:exception", (e, runnable) => {
6       console.log("error", e);
7       console.log("runnable", runnable);
8       return false;
9     });
10
11     cy.visit('http://localhost:8080/proyecto_sergio/');
12     cy.get(':nth-child(1) > input').click();
13     cy.get(':nth-child(1) > input').type('padmin');
14     cy.get(':nth-child(3) > input').click();
15     cy.get(':nth-child(3) > input').type('admin');
16     cy.get('#login-box > button').click();
17   })
18 })
19
20
```

Resultados Nivel de Prueba de Integración

Relación entre el módulo de crear categoría y crear un producto.

```

prueba_componente.cyjs X
proyecto_sergio > cypress > e2e > prueba_componente.cyjs > ...
1 describe("prueba_componente", () => {
2   it("tests prueba_componente", () => {
3
4     // Corrige eventos no detectados por Cypress:
5     cy.on("uncaught:exception", (e, runnable) => {
6       console.log("error", e);
7       console.log("runnable", runnable);
8       return false;
9     });
10
11
12     cy.viewport(771, 561);
13
14     cy.visit("http://localhost:8080/proyecto_sergio/");
15
16     cy.get("#login-box > div > div:nth-child(1) > input[type=text]").click();
17
18     cy.get("#login-box > div > div:nth-child(1) > input[type=text]").type("padmin");
19
20     cy.get("#login-box > div > div:nth-child(3) > input[type=password]").click();
21
22     cy.get("#login-box > div > div:nth-child(3) > input[type=password]").type("admin");
23
24     cy.get("#login-box > button").click();
25     cy.location("href").should("eq", "http://localhost:8080/proyecto_sergio/sistema/");
26
27     cy.get("#menu > ul > li:nth-child(8) > a > span").click();
28     cy.location("href").should("eq", "http://localhost:8080/proyecto_sergio/sistema/listar_categorias.php");
29
30     cy.get("#contenido > div > div.d-sm-flex.align-items-center.justify-content-between.mb-4 > a").click();
31     cy.location("href").should("eq", "http://localhost:8080/proyecto_sergio/sistema/registro_categorias.php");

```

```

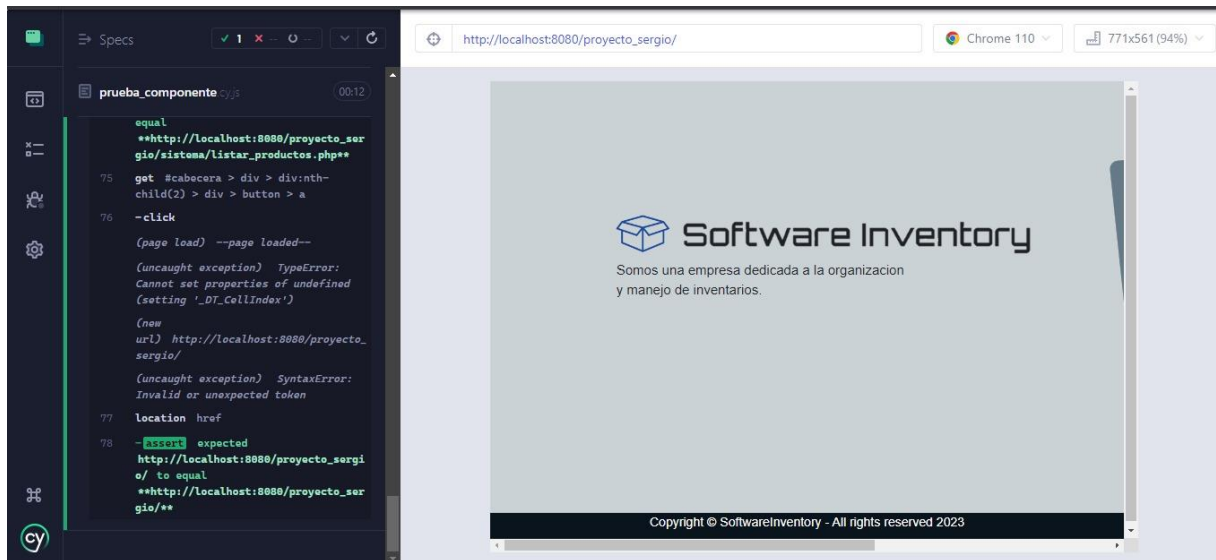
prueba_componente.cyjs X
proyecto_sergio > cypress > e2e > prueba_componente.cyjs > ...
32
33     cy.get("#nombre").click();
34
35     cy.get("#nombre").type("Frutas");
36
37     cy.get("#contenido > div.row > div > form > input").click();
38     cy.location("href").should("eq", "http://localhost:8080/proyecto_sergio/sistema/registro_categorias.php");
39
40     cy.get("#menu > ul > li:nth-child(10) > a > span").click();
41     cy.location("href").should("eq", "http://localhost:8080/proyecto_sergio/sistema/listar_productos.php");
42
43     cy.get("#contenido > div > div.d-sm-flex.align-items-center.justify-content-between.mb-4 > a").click();
44     cy.location("href").should("eq", "http://localhost:8080/proyecto_sergio/sistema/registro_productos.php");
45
46     cy.get("#codigobarras").click();
47
48     cy.get("#codigobarras").type("7676");
49
50     cy.get("#descripcion").click();
51
52     cy.get("#descripcion").type("Manzana");
53
54     cy.get("#precioventa").click();
55
56     cy.get("#precioventa").type("1500");
57
58     cy.get("#preciocosto").click();
59
60     cy.get("#preciocosto").type("750");
61
62     cy.get("#stock").click();
63

```

```

prueba_componente.cyjs X
proyecto_sergio > cypress > e2e > prueba_componente.cyjs > ...
63
64     cy.get("#stock").type("25");
65
66     cy.get("#idcategoria").select('Frutas');
67
68     cy.get("#idcategoria").type("4");
69
70     cy.get("#idproveedor").select('Proveedor General');
71
72     cy.get("#idproveedor").type("9");
73
74     cy.get("#contenido > div.row > div > form > input").click();
75     cy.location("href").should("eq", "http://localhost:8080/proyecto_sergio/sistema/registro_productos.php");
76
77     cy.get("#menu > ul > li:nth-child(10) > a > span").click();
78     cy.location("href").should("eq", "http://localhost:8080/proyecto_sergio/sistema/listar_productos.php");
79
80     cy.get("#cabecera > div > div:nth-child(2) > div > button > a").click();
81     cy.location("href").should("eq", "http://localhost:8080/proyecto_sergio/");
82
83   });
84 });
85
86 // reviderSourceMap=BCCECGCICKCMOORDUDXCZCbDeDhBDkBCmBCoBCqBCsBCuBCwBCyBC0BC2BC4BC6BC8BC+BCgCDjCDmCD

```



Resultados Nivel de Pruebas de Sistema

