UNIVERSIDAD TECNOLÓGICA DE PANAMÁ FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE INGENIERÍA DE SOFTWARE

LICENCIATURA EN DESARROLLO DE SOFTWARE
CALIDAD DEL SOFTWARE

PROYECTO FINAL

PLAN DE EVALUACIÓN – MÓDULO DE AUTENTICACIÓN (LOGIN, LOGOUT, RECUPERACIÓN DE CONTRASEÑA, EMAIL, BLOQUEO)

Prof. Teodolinda Briceño

Pinel, Alexis

Grupo: 1LS142

CIUDAD DE PANAMÁ, 12 DE DICIEMBRE DE 2025

ÍNDICE

INTRODUCCIÓN

El presente documento describe el plan de pruebas integral desarrollado para validar el módulo de autenticación de la plataforma web del restaurante Wushi. Este módulo constituye un componente crítico del sistema, ya que gestiona el acceso de usuarios, la seguridad de credenciales y la recuperación de cuentas.

El objetivo principal de este plan de pruebas es garantizar que el módulo de autenticación funcione de manera correcta, segura y confiable, cumpliendo con los requisitos funcionales establecidos y los estándares de calidad de software. Para ello, se ha desarrollado una estrategia de pruebas basada en los estándares IEEE 829 e ISO/IEC 29119-3, que abarca desde la especificación de casos de prueba hasta la ejecución y el registro de resultados.

El alcance del plan incluye la validación de cinco funcionalidades principales: registro de usuarios, inicio de sesión (login), cierre de sesión (logout), bloqueo por intentos fallidos y recuperación de contraseña. Adicionalmente, se realizaron pruebas de análisis estático utilizando SonarQube Cloud para evaluar la calidad y seguridad del código fuente.

La estrategia de pruebas aplicada combina técnicas de caja negra (clases de equivalencia, valores límite, tablas de decisión y pruebas de estados) con análisis estático de código, permitiendo una cobertura completa tanto de la funcionalidad como de los aspectos de seguridad del sistema.

Este documento se estructura en las siguientes secciones: Plan de Pruebas, Diseño de Pruebas, Especificación de Casos de Prueba, Procedimientos de Prueba, Ejecución y Registro de Resultados, Métricas, y Conclusiones. Los resultados obtenidos proporcionan una evaluación objetiva del estado actual del módulo de autenticación y recomendaciones para su mejora antes del despliegue en producción.

PLAN DE PRUEBAS (TEST PLAN)

Alcance

El presente plan de pruebas abarca la validación completa del módulo de autenticación de la plataforma web del restaurante sushi. Las pruebas se enfocan tanto en el correcto funcionamiento a nivel funcional como en aspectos de seguridad básica, usabilidad y control de errores.

Objetivos

El objetivo principal del presente plan de pruebas es asegurar que el módulo de autenticación funcione de manera correcta, segura y confiable, garantizando que:

- Cumpla correctamente con todos los requisitos funcionales.
- Responda adecuadamente ante entradas válidas e inválidas.
- Mantenga un comportamiento lógico, seguro y estable.
- Gestione correctamente los intentos fallidos de acceso.
- Ejecute de forma completa el proceso de recuperación de contraseña.

Funcionalidades Incluidas:

Las siguientes funcionalidades serán probadas:

- **Login de usuarios**: Acceso al sistema con credenciales válidas.
- **Logout de usuarios**: Cierre de sesión correcto.
- **Recuperación de contraseña**: Proceso completo de recuperación vía email.
- **Bloqueo por intentos fallidos**: Sistema de seguridad ante accesos indebidos.
- **Validación de Email**: Verificación de formato y existencia de correos electrónicos.

Funcionalidades Excluidas:

Quedan fuera del alcance de este plan de pruebas los siguientes elementos:

- Pruebas relacionadas con pasarelas de pago, pedidos en línea u otros módulos distintos a la autenticación.

- Pruebas de carga extrema o estrés del servidor.
- Auditorías de seguridad avanzadas como pruebas de penetración (pentesting).
- Validaciones internas de infraestructura, como configuración de servidores o bases de datos a nivel de administrador del sistema.

Estrategias de Pruebas

Tipos de Pruebas

- Pruebas funcionales
- Validaciones de datos
- Pruebas de control de flujo
- Pruebas de seguridad básica
- Pruebas de estados del sistema
- Pruebas estáticas de calidad de código con SonarQube Cloud

Técnicas de Prueba Aplicadas de Caja Negra

- Clases de Equivalencia
- Valores Límite
- Tablas de Decisión
- Pruebas de Estados

Caja Blanca:

No aplica. Se seleccionaron únicamente técnicas de caja negra debido a que las pruebas se realizaron desde la perspectiva del usuario final, validando el comportamiento funcional del módulo de autenticación sin acceso directo al código fuente. Las pruebas estáticas con SonarQube complementan el análisis de calidad del código.

Criterios de Aceptación

Una prueba será considerada **aprobada** si:

El sistema presenta el resultado exactamente como se define en el resultado esperado.	✓
No se generan errores del sistema.	✓
Se cumplen las reglas de negocio definidas.	✓

Una prueba será considerada **rechazada** si:

El resultado obtenido no coincide con el esperado.	✗
El sistema permite accesos indebidos.	✗
El bloqueo no se ejecuta correctamente.	✗
El proceso de recuperación no finaliza correctamente.	✗

Recursos del Proyecto

Recurso	Detalles	Tipo de Recurso
Tester / Analista de QA	Alexis Pinel	R. Humano
Desarrollador	Alexis Pinel	R. Humano
Usuario de Pruebas	Josué Figueroa	R. Humano
Computadora (Laptop/Desktop)	Con S.O. Windows	R. Técnico
Navegador Web	Mozilla Firefox	R. Técnico
Servidor Local	Apache con Docker	R. Técnico
Base de datos de usuarios	MySQL y PhpMyAdmin	R. Técnico
Herramientas para documentación	Word/Excel	R. Técnico
Herramienta de análisis estático	SonarQube Cloud	R. Técnico

Roles y Responsabilidades

Roles del Equipo de Pruebas:

- **Tester/Analista QA - Elena Wu**: Diseño, ejecución y documentación de casos de prueba.
- **Desarrolladores - Mack Torres y Alexis Pinel**: Corrección de defectos y soporte técnico.
- **Usuario de Pruebas - Josué Figueroa**: Ejecución de pruebas de usabilidad.

Ambiente de Pruebas

Las pruebas se ejecutarán en:

- Ambiente local
- Base de datos de pruebas
- Usuarios de prueba
- Correos electrónicos simulados
- Tokens de recuperación generados automáticamente

Riesgos del Proyecto

Riegos	Impacto
Fallo del Servidor local	**Alto**
Token de recuperación no funcional	**Alto**
Base de datos corrupta	**Alto**
Errores de validación	**Medio**
Datos de prueba incorrectos	**Medio**

Criterios de Entrada y Salida

Criterios de Entrada

- Módulo instalado correctamente
- Base de datos funcional
- Usuarios de prueba creados
- Acceso al sistema habilitado

Criterios de Salida

- 100% de los casos ejecutados
- Resultados documentados
- Defectos registrados
- Métricas calculadas

DISEÑO DE PRUEBAS (TEST DESIGN)

Identificación del Test Design

Test Design ID	TD-AUTH-01
Módulo	Autenticación
Fecha	Diciembre 2025
Responsable	Grupo 142-2
Versión	1.0

Características a Probar

Se evaluarán las siguientes funcionalidades:

Funcionalidades	
Registro de Nuevos Usuarios	Creación de cuenta con datos válidos
	Validación de campos obligatorios
	Validación de formato de email
	Validación de fortaleza de contraseña
	Detección de correos ya registrados
	Confirmación de registro exitoso
Login	Acceso con credenciales válidas
	Rechazo con credenciales inválidas
	Manejo de usuario inexistente
	Manejo de contraseña incorrecta
Logout	Cierre correcto de sesión activa

	Redirección posterior al logout
Recuperación de Contraseña	Solicitud de recuperación
	Validación de correo existente
	Envío de enlace o token
	Restablecimiento de contraseña
Bloqueo por Intentos Fallidos	Conteo de intentos
	Bloqueo automático
	Mensaje de cuenta bloqueada
	Restauración del acceso tras recuperación
Validación de Email	Formato correcto
	Formato incorrecto
	Email existente
	Email inexistente

Características No a Probar

Las siguientes funcionalidades del módulo de autenticación quedan excluidas de este diseño de pruebas:

- Integración con servicios de autenticación de terceros (Google, Facebook, Microsoft).
- Autenticación de dos factores (2FA) o autenticación multifactor (MFA).
- Cambio de contraseña desde el perfil del usuario autenticado (solo se prueba recuperación de contraseña).
- Gestión avanzada de roles y permisos de usuario.
- Sesiones concurrentes desde múltiples dispositivos.
- Recordar sesión ("Remember me").
- Límite de tiempo de sesión por inactividad.

- Registro mediante redes sociales.

Enfoque de Diseño de Pruebas de Caja Negra para Validación Funcional

Técnica	Aplicación
Clases de Equivalencia	Datos Válidos o Inválidos
Valores Límite	Longitud de Campos
Tablas de Decisión	Bloqueo de Cuenta
Pruebas de Estados	Activo, bloqueado, recuperación

Enfoque de Pruebas Estáticas

Técnica	Aplicación
Seguridad	Vulnerability + OWASP + CWE
Fiabilidad	Bugs
Mantenibilidad	Code Smells
Accesibilidad	WCAG
Portabilidad	ES2020

Criterios de Selección de Casos

Los casos de prueba fueron seleccionados aplicando los siguientes criterios:

- **Cobertura funcional completa:** Se incluyen todos los flujos principales y alternativos de cada funcionalidad.
- **Escenarios de error críticos:** Se priorizan los casos de error más comunes y de mayor impacto en la experiencia del usuario.
- **Validación de límites:** Se incluyen casos en los valores límite de todos los campos de entrada.
- **Seguridad:** Se incluyen casos que validan el comportamiento del sistema ante intentos de acceso no autorizado.
- **Reglas de negocio:** Cada regla de negocio definida tiene al menos un caso de prueba asociado.

- **Alto riesgo:** Se priorizan funcionalidades críticas como el bloqueo de cuenta y recuperación de contraseña.

Datos de Prueba Preliminares o Reglas de Derivación

Clases de Equivalencia

Registro de Usuarios

Campo	Clase Válida	Clase Inválida
Nombre de Usuario	Texto Válido	Vacío
Email	Formato Válido	Sin @, sin dominio
Contraseña	≥ 8 caracteres	< 8 caracteres
Confirmación de contraseña	coincide	no coincide

Login

Campo	Clase Válida	Clase Inválida
Nombre de Usuario	Existente	No Registrado
Email	Correcto	Incorrecto

Recuperación

Campo	Clase Válida	Clase Inválida
Nombre de Usuario/Email	Registrado	No Registrado

Valores Límites

Campo	Valor Mínimo	Límite Inferior	Límite Superior	Valor Máximo	Inválido Bajo	Inválido Alto
Contraseña	8 caracteres	8	20	20 caracteres	7 caracteres	21 caracteres
Nombre Usuario	3 caracteres	3	20	20 caracteres	2 caracteres	21 caracteres
Email	test@a.co (9 caracteres)	9	100	usuario@dominio.com	sin @	>100 caracteres

Tabla de Decisión - Bloqueo de Cuenta

Intentos Fallidos	Usuario Válido	Resultado Esperado
1	Sí	Acceso denegado
2	Sí	Acceso denegado
3	Sí	Cuenta Bloqueada
3	No	Acceso denegado

Pruebas de Estado del Sistema

Estado	Descripción
Activo	Usuario puede iniciar sesión
Bloqueado	Usuario no puede acceder
En Recuperación	Usuario puede cambiar contraseña
Cerrado	Sesión Finalizada

Requisitos de Ambiente y Herramientas

Para ejecutar correctamente el presente diseño de pruebas se requiere:

Ambiente de Software:

- Sistema Operativo: Windows 10 o superior.
- Navegador Web: Mozilla Firefox (versión 120 o superior).
- Servidor Local: Apache con Docker configurado.
- Base de Datos: MySQL 8.0 con PhpMyAdmin instalado.
- Herramienta de análisis estático: SonarQube Cloud con acceso configurado.

Ambiente de Datos:

- Base de datos de pruebas poblada con usuarios de prueba (mínimo 10 usuarios).
- Usuarios en diferentes estados: activos, bloqueados, en proceso de recuperación.
- Servidor de correo electrónico configurado para envío de tokens de recuperación.
- Correos electrónicos de prueba accesibles para validación.

Herramientas de Documentación:

- Microsoft Word o Excel para registro de resultados.
- Acceso a sistema de gestión de defectos (si aplica).

Dependencias

El presente diseño de pruebas tiene las siguientes dependencias que deben cumplirse antes de iniciar la ejecución:

Dependencias Técnicas:

- Módulo de autenticación completamente desarrollado y desplegado en ambiente de pruebas.
- Base de datos con esquema de usuarios, contraseñas y tokens implementado.
- Servicio de envío de correos electrónicos funcional y configurado.
- Mecanismo de generación de tokens de recuperación operativo.

Dependencias de Datos:

- Usuarios de prueba creados en base de datos con credenciales conocidas.
- Al menos 3 usuarios en estado "bloqueado" para pruebas de recuperación.
- Correos electrónicos de prueba válidos y accesibles.

Dependencias de Equipo:

- Disponibilidad del desarrollador para corrección de defectos críticos.
- Acceso del tester a los logs del sistema para diagnóstico de errores.
- Disponibilidad del usuario de pruebas para validaciones de usabilidad.

Riesgos Asociados

Riesgo	Probabilidad	Impacto	Mitigación
Tokens de recuperación no llegan al correo electrónico	Media	Alto	Verificar configuración SMTP y realizar prueba previa de envío antes de iniciar ejecución
Base de datos no refleja cambios en tiempo real	Baja	Medio	Validar conexión y permisos de escritura antes de cada sesión de pruebas
Mecanismo de bloqueo no se activa después de 5 intentos	Media	Alto	Probar el mecanismo de bloqueo de forma aislada en primera sesión
Contraseñas no se encriptan correctamente en BD	Baja	Crítico	Revisar con el desarrollador la implementación de hash antes de pruebas
Sesiones no se cierran correctamente al hacer logout	Media	Medio	Incluir validación de tokens de sesión en cada caso de logout
Falta de usuarios de prueba con datos variados	Alta	Medio	Crear un conjunto completo de usuarios de prueba antes de iniciar
Incompatibilidad del navegador con funcionalidades	Baja	Bajo	Mantener Firefox actualizado y tener Chrome como alternativa

CASOS DE PRUEBAS (TEST CASES)

Caso de Prueba: Registro de Usuarios

Registro exitoso con datos válidos

ID	TC-REG-01
Precondición	El sistema está operativo
Datos de Entrada	Nombre válido email válido contraseña válida
Pasos	1. Acceder al formulario de registro 2. Ingresar datos válidos 3. Presionar “Registrar”
Resultado Esperado	Usuario registrado correctamente

Registro con email inválido

ID	TC-REG-02
Precondición	Formulario activo
Datos de Entrada	Email sin “@”
Pasos	1. Acceder al formulario de registro 2. Ingresar dato 3. Presionar “Registrar”
Resultado Esperado	Mensaje de formato incorrecto

Registro con contraseña menor a 8 caracteres

ID	TC-REG-03
Precondición	Sistema activo
Datos de Entrada	Contraseña = "12345"
Pasos	1. Acceder al formulario de registro 2. Ingresar dato 3. Presionar “Registrar”
Resultado Esperado	Advertencia de contraseña con menos caracteres de los aceptados

Registro con correo ya existente

ID	TC-REG-04
Precondición	Correo existente
Datos de Entrada	Email ya registrado
Pasos	1. Acceder al formulario de registro 2. Ingresar dato 3. Presionar “Registrar”
Resultado Esperado	Mensaje: “Usuario ya existe”

Caso de Prueba: Login

Login exitoso

ID	TC-LOG-01
Precondición	Usuario registrado
Datos de Entrada	Usuario válido / contraseña correcta
Pasos	1. Acceder al formulario de inicio de sesión 2. Ingresar dato 3. Presionar “Iniciar Sesión”
Resultado Esperado	Acceso permitido

Login con contraseña incorrecta

ID	TC-LOG-02
Precondición	Usuario registrado
Datos de Entrada	Contraseña incorrecta
Pasos	1. Acceder al formulario de inicio de sesión 2. Ingresar dato 3. Presionar “Iniciar Sesión”
Resultado Esperado	Acceso denegado

Login con usuario inexistente

ID	TC-LOG-03
Precondición	Usuario inexistente
Datos de Entrada	Usuario no registrado
Pasos	1. Acceder al formulario de inicio de sesión 2. Ingresar dato 3. Presionar “Iniciar Sesión”
Resultado Esperado	Mensaje “Usuario no existe”

Caso de Prueba: Bloqueo por Intentos

Bloqueo tras 3 intentos fallidos

ID	TC-BLQ-01
Precondición	Usuario Activo
Datos de Entrada	3 contraseñas incorrectas
Pasos	1. Acceder al formulario de inicio de sesión 2. Ingresar dato incorrecto 3 veces 3. Presionar “Iniciar Sesión”
Resultado Esperado	Cuenta bloqueada

Login con cuenta bloqueada

ID	TC-BLQ-02
Precondición	Cuenta bloqueada
Datos de Entrada	Credenciales de cuenta

Pasos	1. Acceder al formulario de inicio de sesión 2. Ingresar datos 3. Presionar “Iniciar Sesión”
Resultado Esperado	Mensaje “Cuenta bloqueada”

Caso de Prueba: Recuperación de Contraseña

Recuperación con cuenta no registrada

ID	TC-REC-01
Precondición	Sistema activo y formulario de recuperación disponible
Datos de Entrada	Email: correo_noexiste@test.com (no registrado)
Pasos	1. Acceder al formulario de recuperación de contraseña. 2. Ingresar email no registrado. 3. Presionar "Recuperar contraseña".
Resultado Esperado	Mensaje “Correo no registrado” o “El correo ingresado no existe en el sistema”

Solicitud de recuperación válida

ID	TC-REC-02
Precondición	Usuario registrado en el sistema con email verificado
Datos de Entrada	Email: usuario@test.com (registrado)
Pasos	1. Acceder al formulario de recuperación de contraseña. 2. Ingresar email registrado. 3. Presionar "Recuperar contraseña".
Resultado Esperado	Envío exitoso de correo electrónico con token de recuperación

Cambio de contraseña exitoso con token válido

ID	TC-REC-03
Precondición	Token de recuperación generado, válido y no expirado
Datos de Entrada	- Token: 908353 (válido) - Nueva contraseña: NuevaPass123! - Confirmar contraseña: NuevaPass123!
Pasos	1. Ingresar el campo de token con token válido. 2. Ingresar nueva contraseña. 3. Confirmar nueva contraseña. 4. Presionar "Cambiar contraseña".
Resultado Esperado	Contraseña actualizada correctamente y mensaje de confirmación

Cambio de contraseña con token inválido o expirado

ID	TC-REC-04
Precondición	Token generado pero inválido, expirado o ya utilizado
Datos de Entrada	- Token: 908353 (inválido o expirado) - Nueva contraseña: NuevaPass123! - Confirmar contraseña: NuevaPass123!
Pasos	1. Ingresar el campo de token con token inválido, expirado o ya utilizado. 2. Ingresar nueva contraseña. 3. Confirmar nueva contraseña. 4. Presionar "Cambiar contraseña".
Resultado Esperado	Mensaje "Token inválido o expirado" y la contraseña NO se actualiza

Caso de Prueba: LOGOUT

Cierre de sesión exitoso

ID	TC-OUT-01
Precondición	Sesión activa de usuario autenticado
Datos de Entrada	Usuario con sesión activa: usuario@test.com
Pasos	1. Localizar botón "Cerrar sesión". 2. Presionar botón "Cerrar sesión".
Resultado Esperado	Sesión cerrada correctamente y redirección a página de login.

Caso de Prueba: Validación de seguridad en conexión a base de datos

ID	TC-SQ-01
Tipo de Prueba	Estática
Herramienta	SonarQube Cloud
Precondición	Código fuente disponible y SonarQube Cloud configurado
Datos de Entrada	Archivo: src/includes/db.php Regla: CWE-521 – Weak Password
Pasos	1. Ejecutar análisis estático con SonarQube Cloud. 2. Revisar regla CWE-521 (Weak Password Requirements). 3. Validar credenciales de conexión a BD.
Resultado Esperado	La conexión debe usar credenciales seguras. Sin vulnerabilidades detectadas en CWE-521.

PROCEDIMIENTOS DE PRUEBA (TEST PROCEDURES)

Identificación General

Test Design ID	TD-AUTH-01
Módulo	Autenticación
Fecha	Diciembre 2025
Responsable	Grupo 142-2
Versión	1.0

Datos Iniciales de Prueba

Elemento	Valor
Usuario Válido	tester1
Usuario Inválido	fakeTester
Contraseña Válida	Tester123*
Contraseña Inválida	12345
Email Existente	wushitester2025@gmail.com
Email Inexistente	fakeemail@gmail.com
Token Válido	Generado por el sistema
Token Inválido	Expirado o alterado

Configuración General del Entorno

Elemento	Configuración Requerida
Sistema Operativo	Windows 10 o superior
Navegador Web	Mozilla Firefox (última versión)
Servidor	Apache con Docker en ejecución
Base de Datos	MySQL con PhpMyAdmin accesible
URL Base	http://localhost/restaurante/
Estado Inicial BD	Tabla usuarios creada y funcional

Procedimientos por Fase de Ejecución

Procedimiento 1

Procedimiento	Registro de Usuarios
Test Procedure ID	TP-REG-01
Propósito	Validar que el sistema registre correctamente usuarios con datos válidos y rechace registros con datos inválidos mostrando mensajes de error apropiados.
Casos Incluidos	• TC-REG-01 • TC-REG-02 • TC-REG-03 • TC-REG-04
Orden de Ejecución	1. Acceder al formulario de registro 2. Ejecutar TC-REG-01 (Registro válido) 3. Ejecutar TC-REG-02 (Email inválido) 4. Ejecutar TC-REG-03 (Contraseña corta) 5. Ejecutar TC-REG-04 (Correo existente)
Resultado Global Esperado	• Solo el TC-REG-01 debe finalizar en PASS • Los demás deben ser rechazados con mensajes correctos

Configuración del Entorno:

- Base de datos sin usuario "tester1" registrado

- Tabla usuarios activa con permisos de inserción

Datos Iniciales:

- Usuario a registrar: tester1
- Email: wushitester2025@gmail.com
- Contraseña: Tester123*

Pasos Detallados:

1. Abrir Firefox y navegar a http://localhost/restaurante/registro.php
2. **TC-REG-01:** Ingresar datos válidos (tester1, wushitester2025@gmail.com, Tester123*) → Presionar "Registrar" → Verificar mensaje de éxito
3. **TC-REG-02:** Ingresar email sin "@" (emailinvalido.com) → Presionar "Registrar" → Verificar mensaje "Formato de email incorrecto"
4. **TC-REG-03:** Ingresar contraseña corta "12345" → Presionar "Registrar" → Verificar mensaje de longitud mínima
5. **TC-REG-04:** Ingresar email ya registrado (wushitester2025@gmail.com) → Presionar "Registrar" → Verificar mensaje "Usuario ya existe"

Criterios de Aceptación Globales:

- ✓ Solo TC-REG-01 debe completarse exitosamente
- ✓ Los casos TC-REG-02, TC-REG-03 y TC-REG-04 deben mostrar mensajes de error claros
- ✓ Solo debe existir 1 registro del usuario "tester1" en la base de datos

Dependencias:

- Ninguna (este es el primer procedimiento a ejecutar)
- Requiere base de datos MySQL activa

Reinicios:

- **Para repetir el procedimiento:** Eliminar manualmente el usuario "tester1" de la tabla usuarios en PhpMyAdmin antes de volver a ejecutar.
- **Entre ciclos de prueba:** Vaciar la tabla usuarios para garantizar un estado inicial limpio.

Procedimiento 2

Procedimiento	Login de Usuario
Test Procedure ID	TP-LOG-01
Propósito	Verificar que el sistema permita el acceso solo a usuarios con credenciales válidas y rechace intentos con credenciales incorrectas o usuarios inexistentes.
Casos Incluidos	• TC- LOG -01 • TC- LOG -02 • TC-LOG-03
Orden de Ejecución	1. Acceder al formulario de login 2. Ejecutar TC-LOG-01 (Login correcto) 3. Ejecutar TC-LOG-02 (Contraseña incorrecta) 4. Ejecutar TC-LOG-03 (Usuario inexistente)
Resultado Global Esperado	• Solo el TC-LOG-01 debe iniciar sesión • Los demás deben mostrar mensajes de error correctos

Configuración del Entorno:

- Usuario "tester1" debe existir en la base de datos
- Campo "bloqueado" debe estar en 0 (cuenta activa)

Datos Iniciales:

- Usuario existente: tester1
- Contraseña correcta: Tester123*
- Usuario inexistente: fakeTester

Pasos Detallados:

1. Navegar a http://localhost/restaurante/login.php
2. **TC-LOG-01:** Ingresar usuario "tester1" y contraseña "Tester123*" → Presionar "Iniciar Sesión" → Verificar acceso exitoso al sistema
3. Cerrar sesión presionando botón "Logout"
4. **TC-LOG-02:** Ingresar usuario "tester1" con contraseña incorrecta → Presionar "Iniciar Sesión" → Verificar mensaje de error "Contraseña incorrecta"
5. **TC-LOG-03:** Ingresar usuario "fakeTester" → Presionar "Iniciar Sesión" → Verificar mensaje "Usuario no existe"

Criterios de Aceptación Globales:

- ✓ Solo TC-LOG-01 debe permitir el acceso al sistema
- ✓ TC-LOG-02 y TC-LOG-03 deben denegar acceso con mensajes de error claros
- ✓ El contador de intentos fallidos debe incrementarse en TC-LOG-02

Dependencias:

- Requiere que TP-REG-01 se haya ejecutado previamente (usuario "tester1" debe existir)
- Usuario "tester1" debe estar desbloqueado

Reinicios:

- **Después de TC-LOG-02:** Reiniciar el contador de intentos fallidos del usuario "tester1" a cero en PhpMyAdmin para evitar bloqueos accidentales.
- **Si hay sesión activa:** Cerrar sesión o limpiar cookies del navegador antes de repetir.

Procedimiento 3

Procedimiento	Bloqueo por Intentos Fallidos
Test Procedure ID	TP-BLQ-01
Propósito	Validar que el sistema bloquee automáticamente una cuenta después de 3 intentos fallidos de login y que una vez bloqueada no permita acceso ni con credenciales correctas.
Casos Incluidos	• TC- BLQ -01 • TC- BLQ -02
Orden de Ejecución	1. Ingresar contraseña incorrecta 3 veces 2. Ejecutar TC-BLQ-01 3. Intentar login nuevamente 4. Ejecutar TC-BLQ-02
Resultado Global Esperado	• La cuenta debe quedar bloqueada • No debe permitir más accesos

Configuración del Entorno:

- Usuario "tester1" activo con campos: bloqueado = 0, intentos_fallidos = 0
- Mecanismo de bloqueo automático implementado

Datos Iniciales:

- Usuario: tester1
- Contraseña correcta: Tester123*
- Contraseña incorrecta: ContraseñaErronea123

Pasos Detallados:

1. Navegar a http://localhost/restaurante/login.php
2. Ingresar contraseña incorrecta 5 veces consecutivas (usuario "tester1", contraseña "ContraseñaErronea123")
3. **TC-BLQ-01:** Al 5to intento → Verificar mensaje "Cuenta bloqueada por múltiples intentos fallidos"

4. Verificar en PhpMyAdmin que campo "bloqueado" = 1
5. **TC-BLQ-02:** Intentar login con contraseña CORRECTA (Tester123*) → Verificar que el sistema sigue bloqueando el acceso con mensaje "Cuenta bloqueada"

Criterios de Aceptación Globales:

- ✓ La cuenta debe bloquearse exactamente después del 5to intento fallido
- ✓ Una vez bloqueada, no debe permitir acceso ni con la contraseña correcta
- ✓ El campo "bloqueado" en la base de datos debe cambiar a 1

Dependencias:

- Requiere que TP-REG-01 se haya ejecutado (usuario "tester1" debe existir)
- Usuario debe estar inicialmente desbloqueado

Reinicios:

- **CRÍTICO - Después de ejecutar:** Desbloquear manualmente la cuenta "tester1" en PhpMyAdmin cambiando los campos "bloqueado" a 0 y "intentos_fallidos" a 0. Sin este reinicio, los procedimientos posteriores fallarán.
- **Antes de otros procedimientos:** Verificar en PhpMyAdmin que el usuario esté activo antes de ejecutar TP-REC-01 o TP-LOG-01.

Procedimiento 4

Procedimiento	Recuperación de Contraseña
Test Procedure ID	TP-REC-01
Propósito	Validar que el sistema permita recuperar contraseñas mediante token enviado por email y rechace solicitudes con emails no registrados o tokens inválidos.
Casos Incluidos	• TC-REC-01 • TC-REC-02 • TC-REC-03 • TC-REC-04
Orden de Ejecución	1. Acceder a "Olvidé mi contraseña" 2. Ejecutar TC-REC-01 3. Ejecutar TC-REC-02 4. Usar token válido → TC-REC-03 5. Usar token inválido → TC-REC-04
Resultado Global Esperado	• El cambio de contraseña solo debe permitirse con token válido • Los tokens inválidos deben ser rechazados

Configuración del Entorno:

- Servidor de correo SMTP configurado y funcional
- Tabla tokens_recuperacion creada en base de datos
- Acceso al email wushitester2025@gmail.com

Datos Iniciales:

- Email registrado: wushitester2025@gmail.com
- Email no registrado: fakeemail@gmail.com
- Nueva contraseña: NuevaPass123!

Pasos Detallados:

1. Navegar a http://localhost/restaurante/login.php → Clic en "¿Olvidaste tu contraseña?"

2. **TC-REC-01:** Ingresar email no registrado (fakeemail@gmail.com) → Presionar "Recuperar" → Verificar mensaje "Correo no registrado"
3. **TC-REC-02:** Ingresar email registrado (wushitester2025@gmail.com) → Presionar "Recuperar" → Verificar envío de correo con token
4. Abrir bandeja de entrada y copiar el token recibido
5. **TC-REC-03:** Ingresar token válido + nueva contraseña (NuevaPass123!) → Presionar "Cambiar contraseña" → Verificar mensaje de éxito
6. Probar login con nueva contraseña para confirmar el cambio
7. **TC-REC-04:** Ingresar token inválido (xyz789invalido) + contraseña → Presionar "Cambiar" → Verificar mensaje "Token inválido o expirado"

Criterios de Aceptación Globales:

- ✓ Solo emails registrados deben recibir correo de recuperación
- ✓ El cambio de contraseña solo debe completarse con token válido
- ✓ Los tokens inválidos o expirados deben ser rechazados
- ✓ Después del cambio exitoso, el usuario debe poder acceder con la nueva contraseña

Dependencias:

- Requiere que TP-REG-01 se haya ejecutado (usuario debe existir)
- Servidor de correo debe estar configurado
- Email de prueba debe ser accesible

Reinicios:

- **Después de TC-REC-03:** Restaurar la contraseña original del usuario "tester1" a "Tester123*" en PhpMyAdmin si se planea ejecutar otros procedimientos.
- **Limpiar tokens usados:** Eliminar registros de la tabla tokens_recuperacion asociados al email de prueba para evitar conflictos.
- **Verificar correo:** Eliminar emails de prueba de la bandeja de entrada para evitar confusiones en ejecuciones futuras.

Procedimiento 5

Procedimiento	Logout
Test Procedure ID	TP-OUT-01
Propósito	Verificar que el sistema cierre correctamente la sesión de usuario, elimine tokens de sesión y redirija a la página de login.
Casos Incluidos	• TC-OUT-01
Orden de Ejecución	1. Iniciar sesión correctamente 2. Presionar “Cerrar sesión” 3. Ejecutar TC-OUT-01
Resultado Global Esperado	• La sesión debe cerrarse • El sistema debe redirigir al login

Configuración del Entorno:

- Usuario "tester1" debe estar desbloqueado
- Sistema de sesiones PHP funcional

Datos Iniciales:

- Usuario: tester1
- Contraseña: Tester123* (o NuevaPass123! si se ejecutó TP-REC-01)

Pasos Detallados:

1. Navegar a http://localhost/restaurante/login.php
2. Iniciar sesión con usuario "tester1" y contraseña válida
3. Verificar acceso exitoso al sistema
4. **TC-OUT-01:** Presionar botón "Cerrar Sesión" → Verificar redirección automática a página de login
5. Intentar acceder directamente a una página protegida (ej: dashboard.php) → Verificar que redirige a login o muestra error de "No autorizado"

Criterios de Aceptación Globales:

- ✓ La sesión debe cerrarse completamente
- ✓ El sistema debe redirigir automáticamente a la página de login
- ✓ No debe ser posible acceder a páginas protegidas después del logout
- ✓ Las cookies de sesión deben ser eliminadas o invalidadas

Dependencias:

- Requiere que TP-LOG-01 funcione correctamente (mecanismo de login)
- Usuario "tester1" debe existir y estar desbloqueado

Reinicios:

- **No requiere reinicios:** El procedimiento limpia automáticamente la sesión al ejecutarse.
- **Si la sesión no se cierra:** Limpiar manualmente las cookies del navegador (Ctrl + Shift + Delete) y volver a intentar.
- **Para repetir:** Simplemente volver a iniciar sesión con TP-LOG-01.

Orden de Ejecución Recomendado

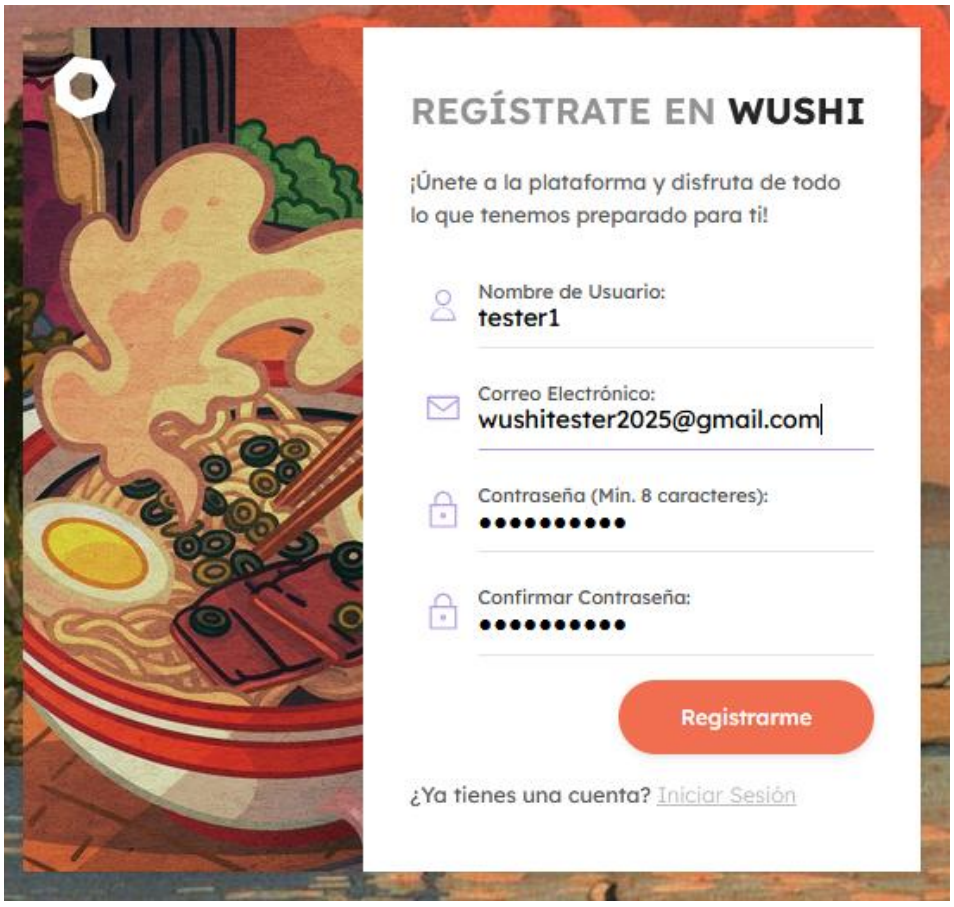
1. **TP-REG-01** - Registro de Usuarios
2. **TP-LOG-01** - Login de Usuario
3. **TP-OUT-01** - Logout
4. **TP-BLQ-01** - Bloqueo por Intentos Fallidos
5. **TP-REC-01** - Recuperación de Contraseña

EJECUCIÓN DE PRUEBAS Y REGISTRO DE RESULTADOS

Caso de Prueba: Registro de Usuarios

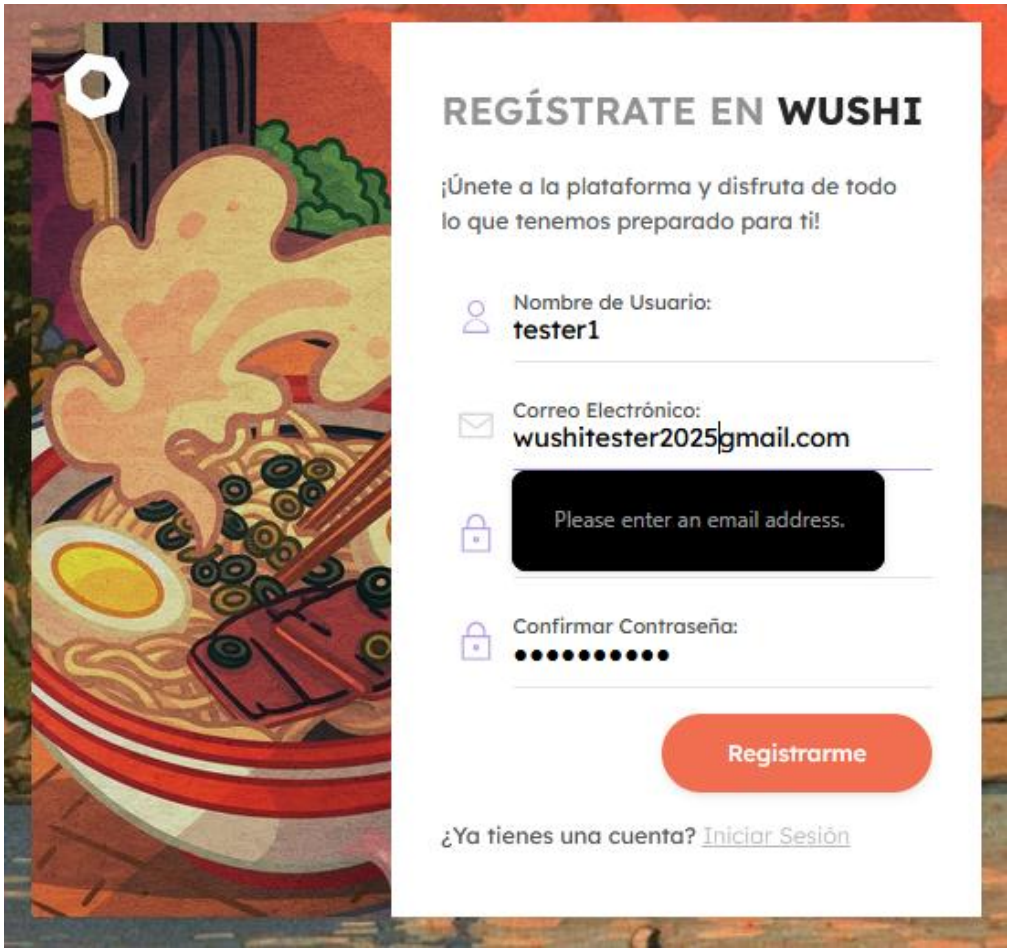
Registro exitoso con datos válidos

ID	TC-REG-01	**DEFECTO ASOCIADO**
Resultado Esperado	Usuario registrado correctamente	-
Resultado Obtenido	Usuario Registrado Correctamente	
Estado	**PASS**	



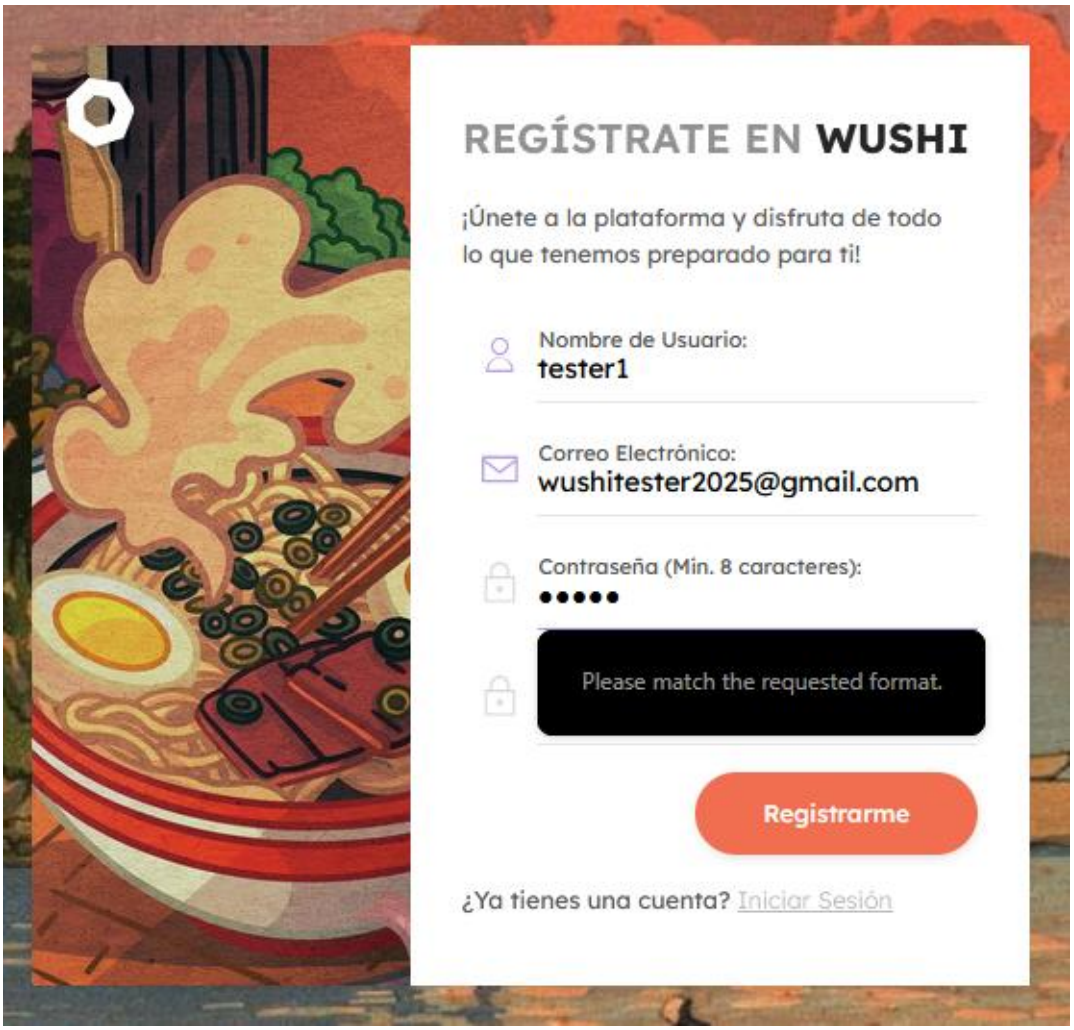
Registro con email inválido

ID	TC-REG-02	DEFECTO ASOCIADO
Resultado Esperado	Mensaje de formato incorrecto	-
Resultado Obtenido	Mensaje de formato incorrecto	
Estado	**PASS**	



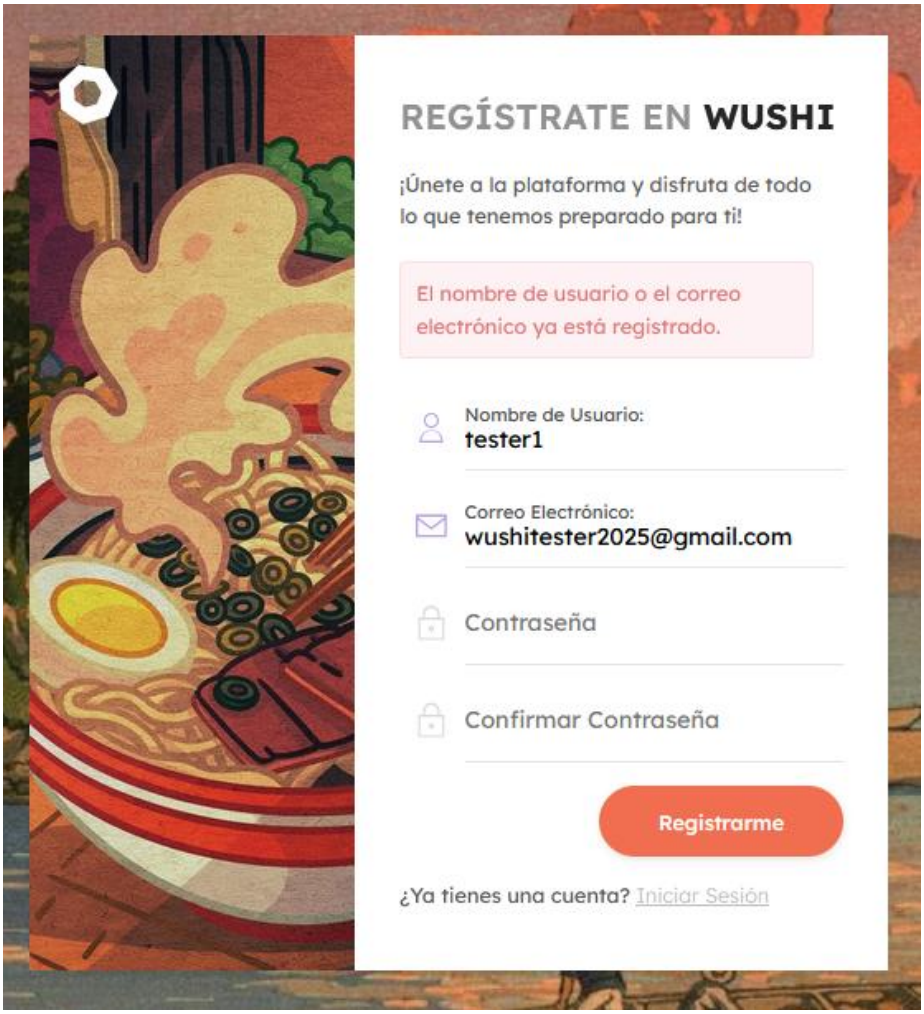
Registro con contraseña menor a 8 caracteres

ID	TC-REG-03	**DEFECTO ASOCIADO**
Resultado Esperado	Advertencia de contraseña con menos caracteres de los aceptados	-
Resultado Obtenido	Advertencia de contraseña con menos caracteres de los aceptados	
Estado	**PASS**	



Registro con correo ya existente

ID	TC-REG-04	DEFECTO ASOCIADO
Resultado Esperado	Mensaje: “Usuario ya existe”	-
Resultado Obtenido	El usuario pudo registrarse	
Estado	**PASS**	



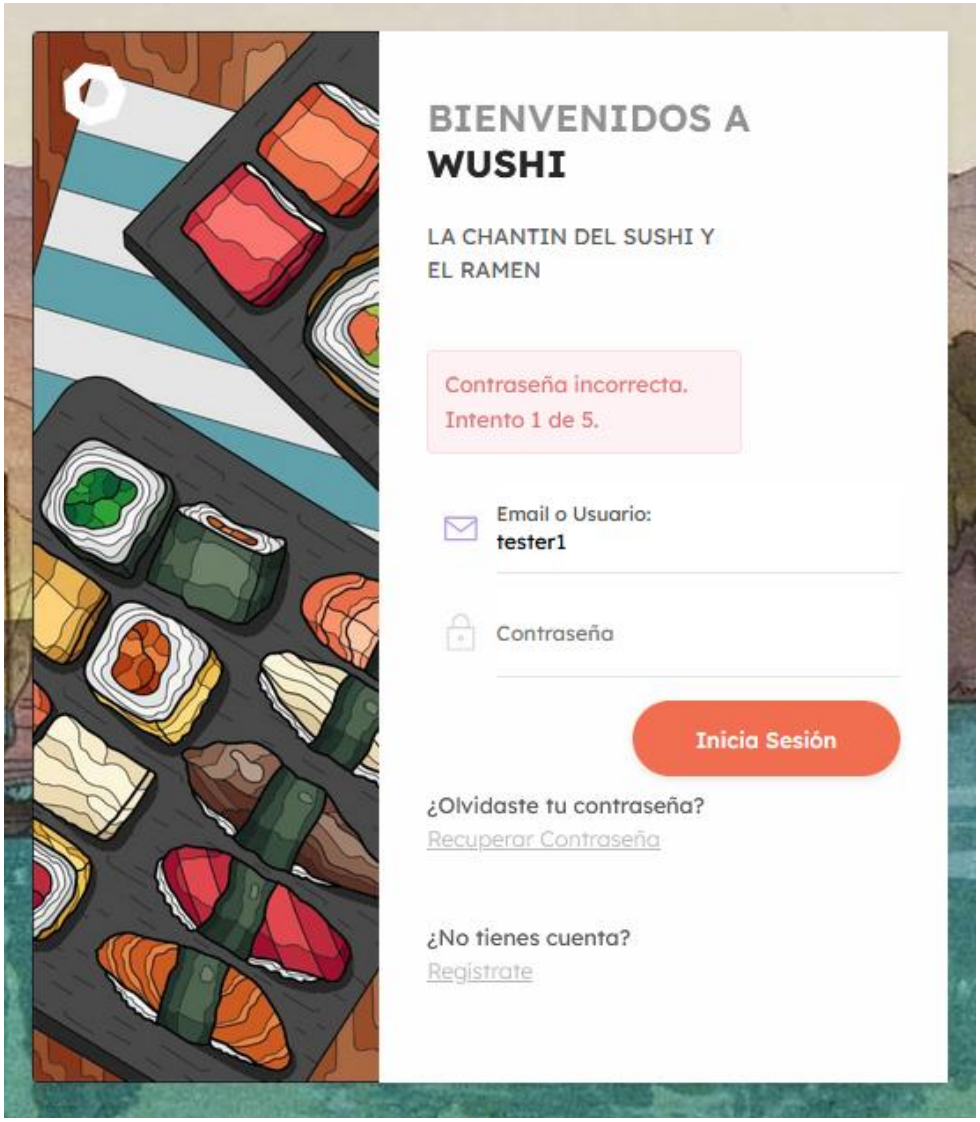
Caso de Prueba: Login

Login exitoso

ID	TC-LOG-01	**DEFECTO ASOCIADO**
Resultado Esperado	Acceso permitido	-
Resultado Obtenido	Acceso permitido	
Estado	**PASS**	

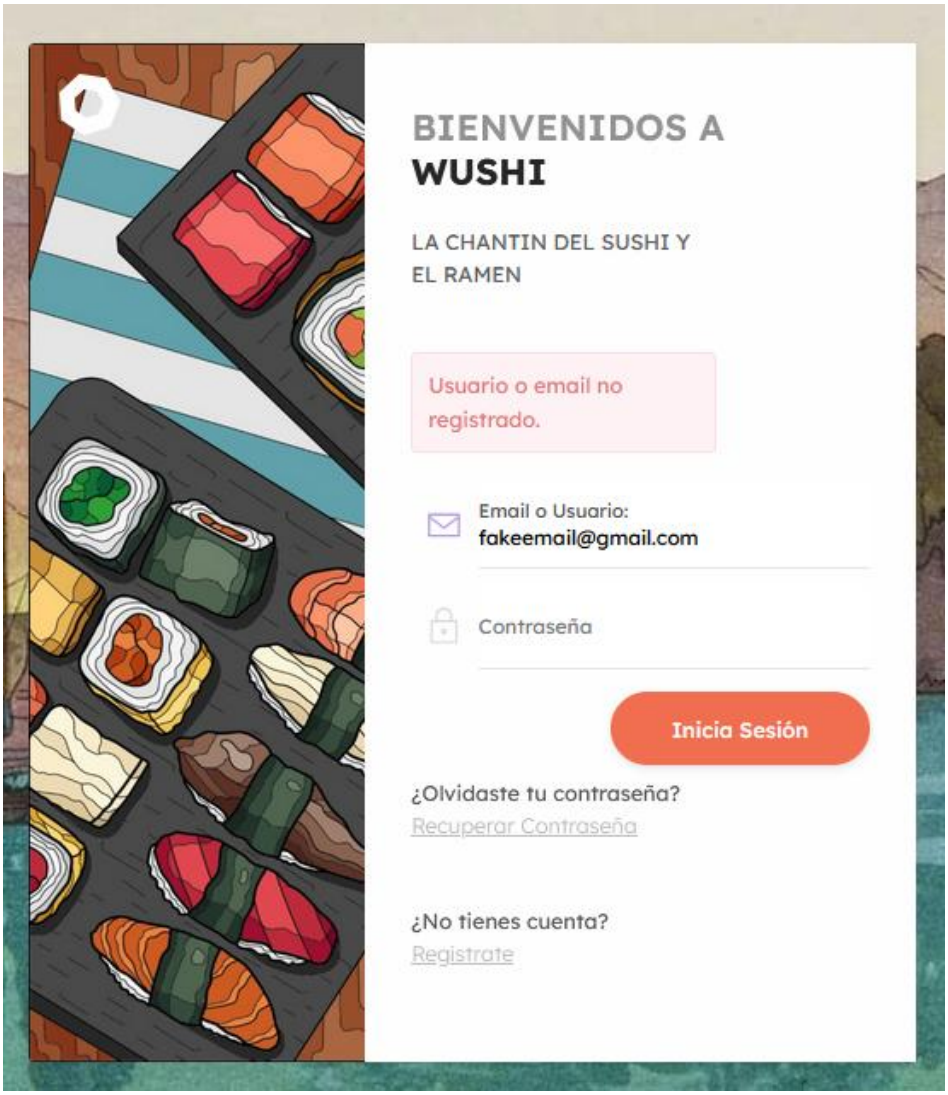
Login con contraseña incorrecta

ID	TC-LOG-02	**DEFECTO ASOCIADO**
Resultado Esperado	Acceso denegado	-
Resultado Obtenido	Acceso denegado	
Estado	**PASS**	



Login con usuario inexistente

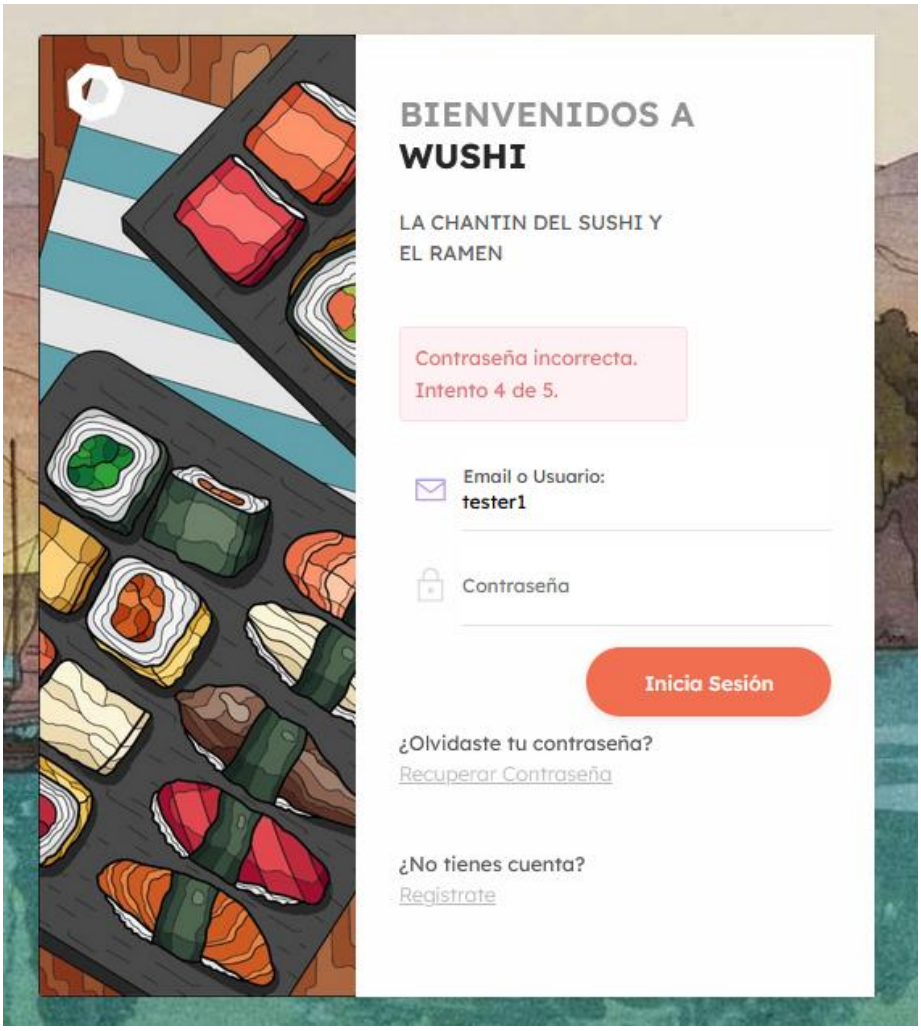
ID	TC-LOG-03	**DEFECTO ASOCIADO**
Resultado Esperado	Mensaje “Usuario no existe”	-
Resultado Obtenido	Mensaje “Usuario no existe”	
Estado	**PASS**	



Caso de Prueba: Bloqueo por Intentos

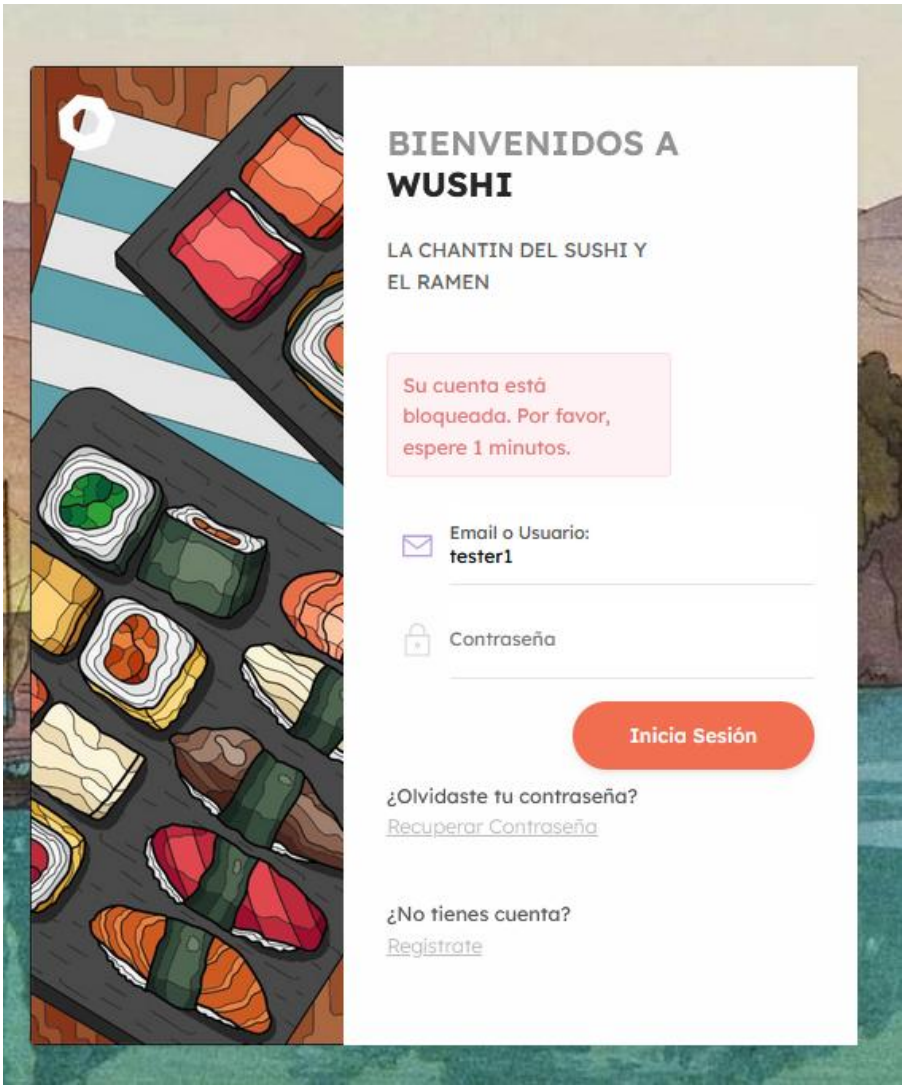
Bloqueo tras 5 intentos fallidos

ID	TC-BLQ-01	**DEFECTO ASOCIADO**
Resultado Esperado	Cuenta bloqueada	-
Resultado Obtenido	Cuenta bloqueada	
Estado	**PASS**	



Login con cuenta bloqueada

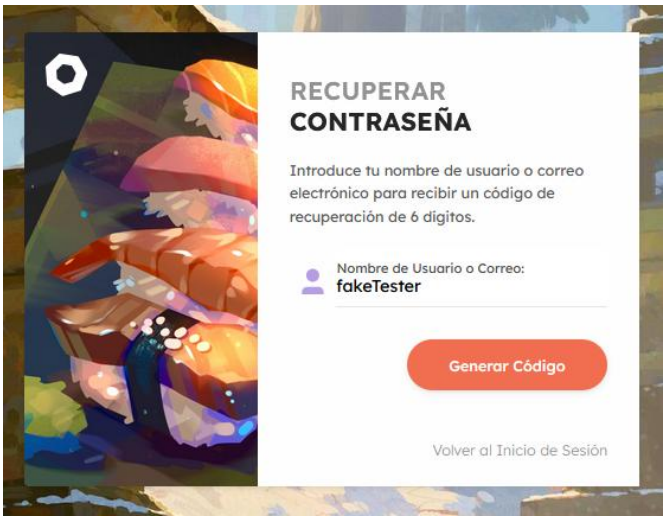
ID	TC-BLQ-02	**DEFECTO ASOCIADO**
Resultado Esperado	Mensaje “Cuenta bloqueada”	-
Resultado Obtenido	Mensaje “Cuenta bloqueada”	
Estado	**PASS**	

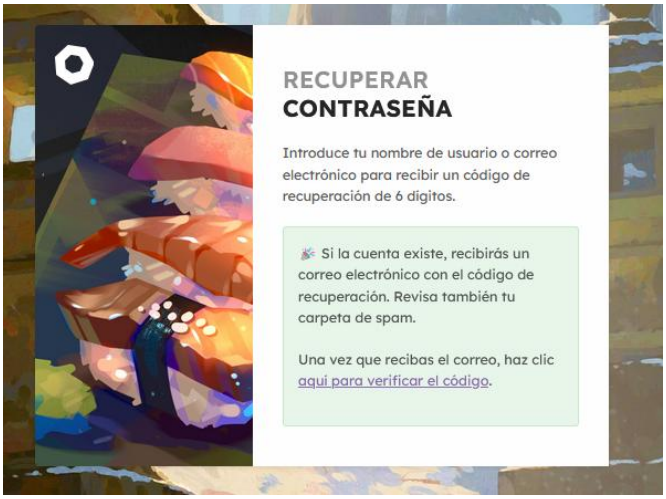


Caso de Prueba: Recuperación de Contraseña

Recuperación con cuenta no registrada

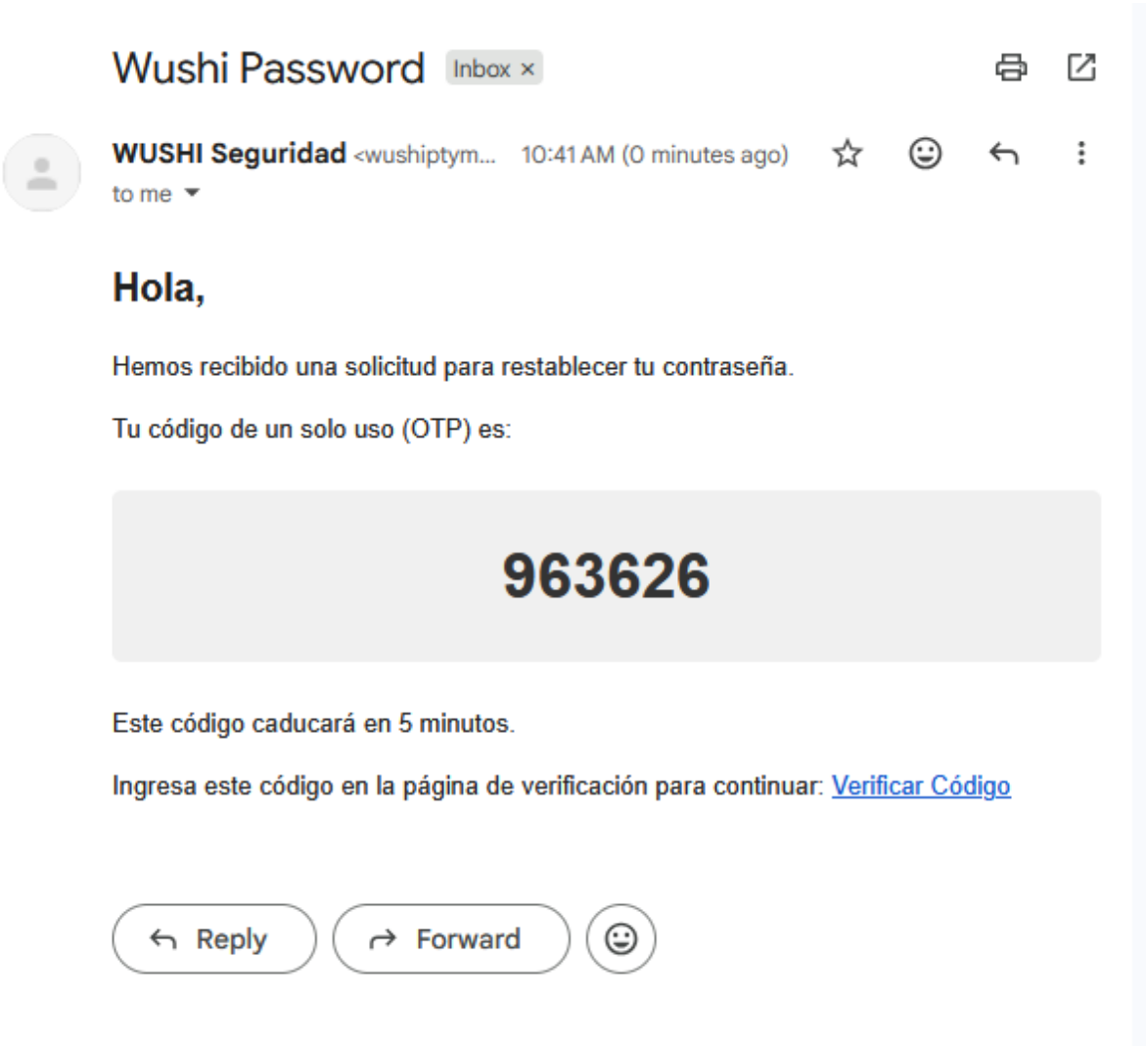
ID	TC-REC-01	**DEFECTO ASOCIADO**
Resultado Esperado	Mensaje “Cuenta no registrada”	DF - 01
Resultado Obtenido	Envía mensaje como si la cuenta estuviera registrada	
Estado	**FAIL**	





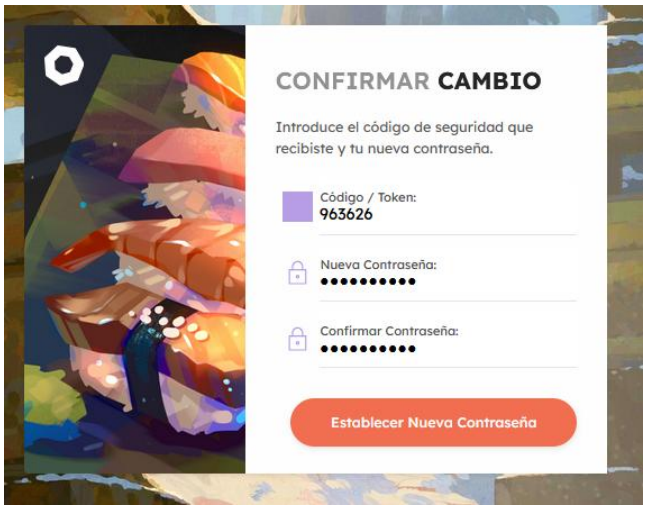
Solicitud de recuperación válida

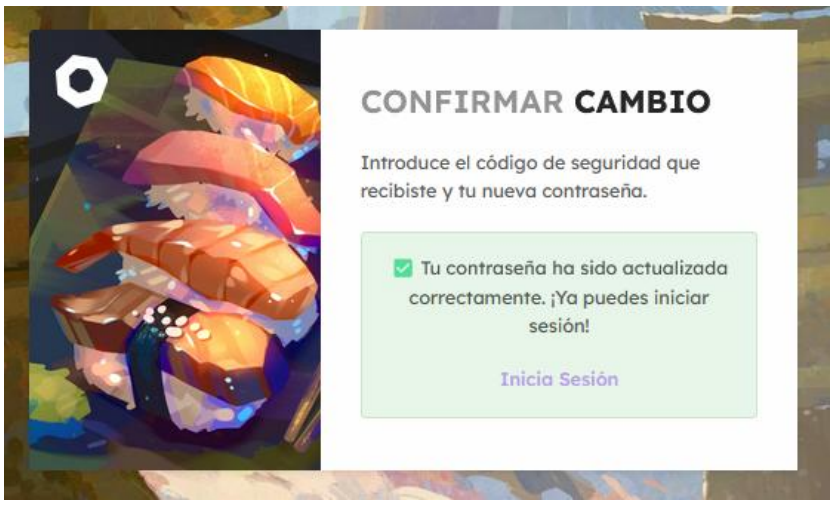
ID	TC-REC-02	**DEFECTO ASOCIADO**
Resultado Esperado	Envío de correo con token	-
Resultado Obtenido	Envío de correo con token	
Estado	**PASS**	



Cambio de contraseña exitoso

ID	TC-REC-03	**DEFECTO ASOCIADO**
Resultado Esperado	Contraseña actualizada	-
Resultado Obtenido	Contraseña actualizada	
Estado	**PASS**	

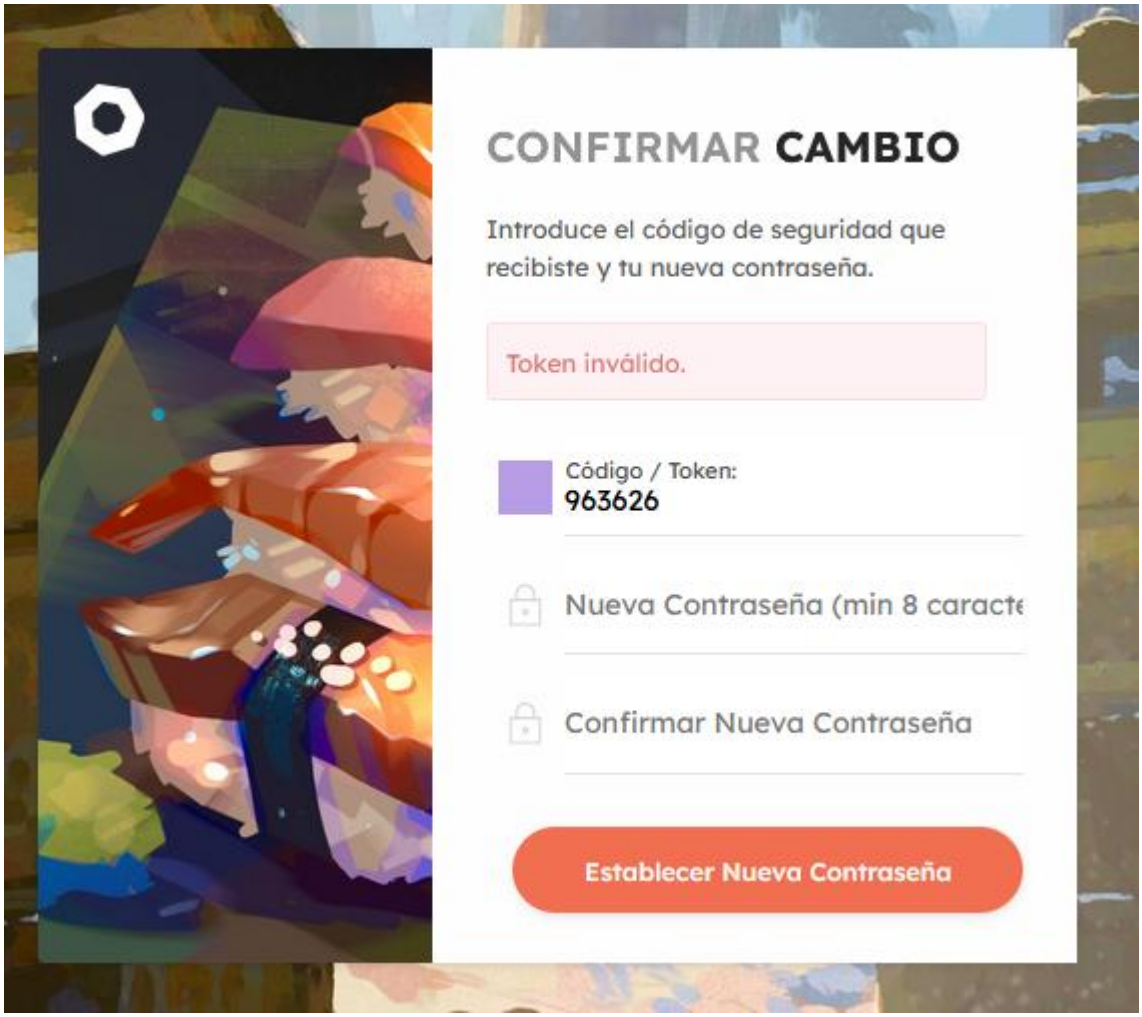




CONTRASEÑA NUEVA: admin123

Cambio de contraseña exitoso

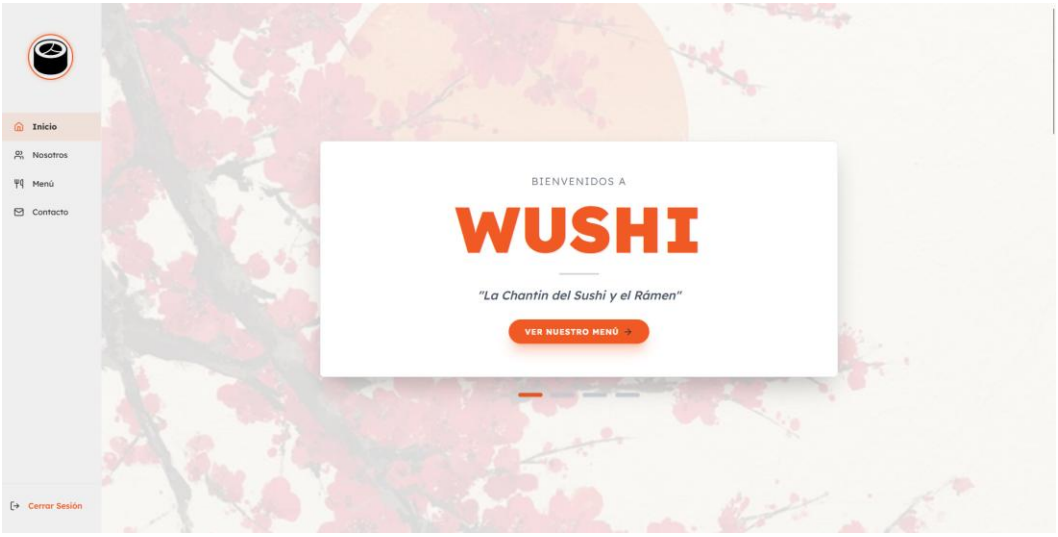
ID	TC-REC-04	**DEFECTO ASOCIADO**
Resultado Esperado	Mensaje “Token inválido o expirado”	-
Resultado Obtenido	Mensaje “Token inválido o expirado”	
Estado	**PASS**	

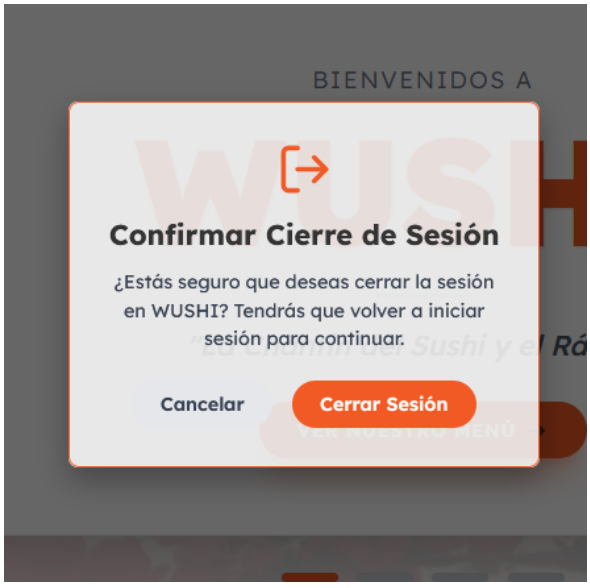


Caso de Prueba: LOGOUT

Cierre de sesión exitoso

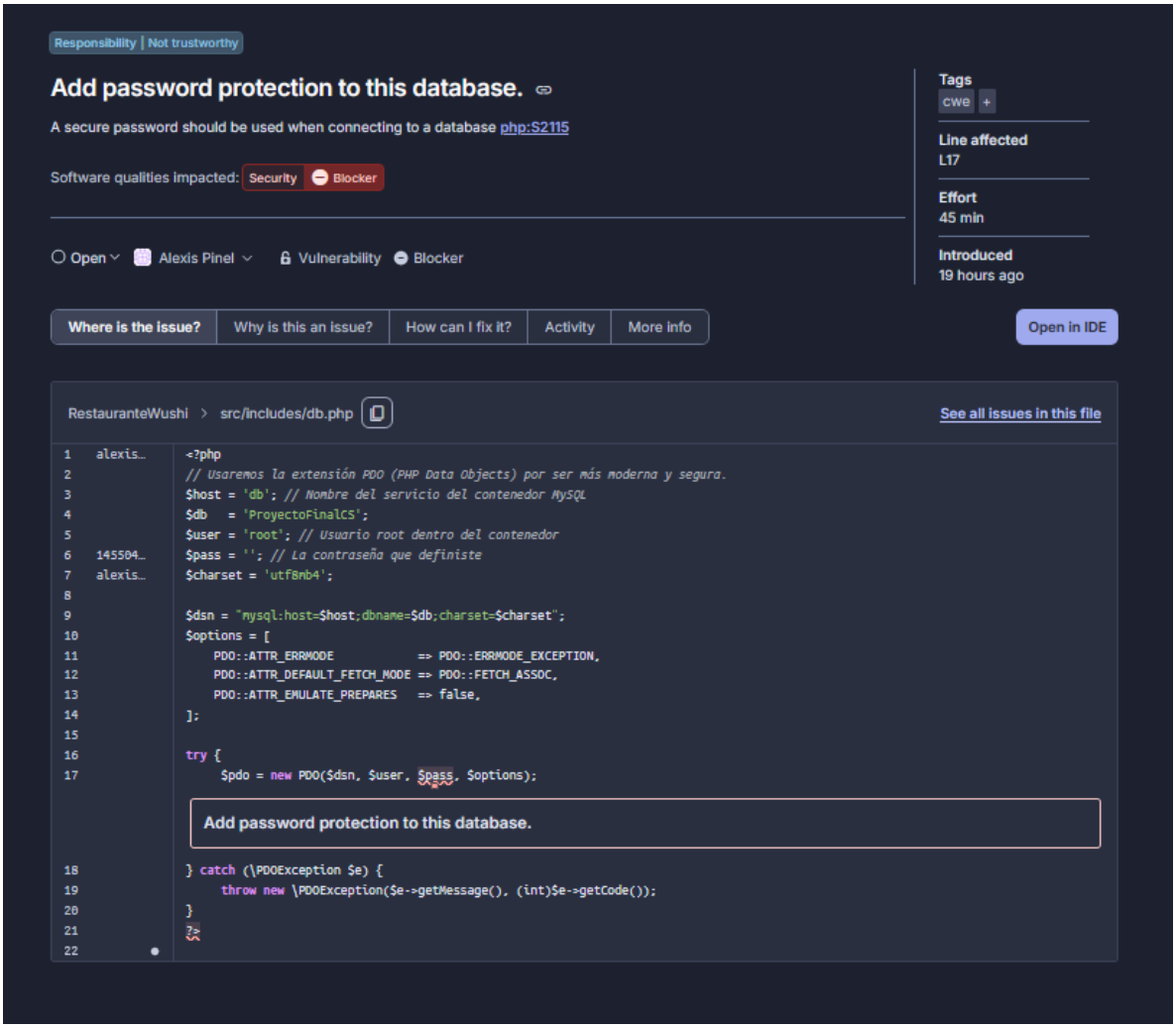
ID	TC-OUT-01	**DEFECTO ASOCIADO**
Resultado Esperado	Sesión cerrada y redirección	-
Resultado Obtenido	Sesión cerrada y redirección	
Estado	**PASS**	





Caso de Prueba: Validación de seguridad en conexión a base de datos

ID	TC-SQ-01	DEFECTO ASOCIADO
Resultado Esperado	La conexión debe usar credenciales seguras	**DF-SQ-01**
Resultado Obtenido	No tiene contraseña	
Estado	**FAIL**	



REGISTRO DE DEFECTOS (BUG REPORT)

ID Defecto (bug)	DF-01
Caso	TC-REC-01
Descripción	Al ingresar un correo electrónico que no está registrado en el sistema, el sistema envía el mensaje de recuperación como si la cuenta existiera, cuando debería mostrar un mensaje de error indicando que la cuenta no se encuentra registrada.
Funcionalidad	Recuperación de Contraseña
Severidad	**ALTA**
Prioridad	**ALTA**
Estado	Abierta

ID Defecto (bug)	DF-SQ-01
Caso	TC-SQ-01
Origen	SonarQube Cloud
Problema	Conexión sin contraseña
Tipo	Vulnerabilidad
Severidad	**BLOCKER**
Estándar	OWASP A7 / CWE-521
Estado	Abierta

MÉTRICAS

Porcentaje De Casos Ejecutados

$$\%\boldsymbol{Casos\ Ejecutados} = \left(\frac{\boldsymbol{Casos\ Ejcutado}}{\boldsymbol{Casos\ Definidos}}\right) \times \mathbf{100}$$

Casos Definidos	14
Casos Ejecutados	14
Resultado	**100%**

Explicación: *El 100% de los casos de prueba definidos fueron ejecutados exitosamente. Esto indica una cobertura completa de ejecución sobre todos los escenarios de prueba planificados para el módulo de autenticación.*

Porcentaje De Casos Aprobados

$$\%\boldsymbol{Casos\ Aprobados} = \left(\frac{\boldsymbol{Casos\ Aprobados}}{\boldsymbol{Casos\ Ejecutados}}\right) \times \mathbf{100}$$

Casos Aprobados	13
Casos Ejecutados	14
Resultado	**92.86%**

Explicación: *El 92.8% de los casos ejecutados fueron aprobados correctamente, cumpliendo con el comportamiento esperado según las especificaciones funcionales.*

Porcentaje De Casos Fallidos

$$\%\boldsymbol{Casos\ Fallidos} = \left(\frac{\boldsymbol{Casos\ Fallidos}}{\boldsymbol{Casos\ Ejecutados}}\right) \times \mathbf{100}$$

Casos Fallidos	1
Casos Ejecutados	14
Resultado	**7.14%**

Explicación: *El 7.14% de los casos de prueba presentaron fallas. Este porcentaje corresponde a 1 caso funcional (TC-REC-01) que no cumplió con el comportamiento esperado.*

Cobertura Por Requisitos Funcionales

Requisito	Casos Definidos	Casos Ejecutados	Cobertura
Registro de Usuarios	4	4	100%
Inicio de Sesión	3	3	100%
Bloqueo por Intentos Fallidos	2	2	100%
Recuperación de Contraseña	4	4	100%
Cierre de Sesión	1	1	100%
Total	14	14	100%

Explicación: *Se logró una cobertura del 100% en todos los requisitos funcionales definidos para el módulo de autenticación. Cada funcionalidad fue validada mediante al menos un caso de prueba, garantizando que todos los aspectos críticos fueron evaluados.*

MÉTRICAS DE SONARQUBE CLOUD

Métrica SonarQube	Valor
Issues Totales	107
Vulnerabilidades	1
Bugs	7
Code Smells	99
Blocker	1
Major	12
Minor	94
Esfuerzo Técnico	3 h 54 min

Defectos Detectados

ID Defecto	Caso Asociado	Funcionalidad	Tipo de Defecto	Severidad	Origen	Estado
DF-01	TC-REC-01	Recuperación de contraseña	Funcional	Alta	Prueba funcional	Abierto
DF-SQ-01	TC-SQ-01	Seguridad Base de Datos	Vulnerabilidad	Blocker	SonarQube Cloud	Abierto

Distribución de Defectos por Severidad

Severidad	Cantidad	Porcentaje
Blocker	1	50%
Alta	1	50%
Media	0	0%
Baja	0	0%
Total	14	100%

Tasa de Defectos por Requisito

Requisito Funcional + Seguridad	Casos Ejecutados	Defectos Detectados	Tasa de Defectos
Registro de Usuarios	4	0	0 %
Inicio de Sesión	3	0	0 %
Bloqueo por Intentos Fallidos	2	0	0 %
Recuperación de Contraseña	4	1	25 %
Cierre de Sesión	1	0	0 %
Seguridad (Análisis Estático)	1	1	100 %
TOTAL	**15**	**2**	**13.33 %**

Explicación:

La tasa promedio de defectos del sistema es del **13.33%** (2 defectos en 15 casos de prueba totales).

Análisis por requisito:

- **Recuperación de Contraseña:** Presenta una tasa de defectos del 25% (1 defecto en 4 casos), específicamente en la validación de correos no registrados (TC-REC-01).
- **Seguridad (Análisis Estático):** Tasa del 100% (1 vulnerabilidad crítica detectada por SonarQube en la conexión a base de datos).

- **Resto de requisitos funcionales:** 0% de defectos, indicando una implementación estable.

ÍNDICE DE ESTABILIDAD DEL SISTEMA

Estabilidad Funcional

Casos Aprobados	13
Casos Ejecutados	14
Índice de Estabilidad Funcional	**92.86%**

Interpretación: *El sistema presenta una alta estabilidad funcional con un 92.86% de casos aprobados. Esto indica que la mayoría de las funcionalidades operan correctamente según lo especificado.*

Estabilidad de Seguridad

Vulnerabilidades Críticas (Blocker)	1
Vulnerabilidades de Seguridad Detectadas	1
Índice de Estabilidad de Seguridad	**0% (Crítico)**

Interpretación: *La presencia de 1 vulnerabilidad de severidad BLOCKER (CWE-521: conexión a base de datos sin contraseña) representa un riesgo crítico de seguridad que compromete completamente la estabilidad del sistema en este aspecto.*

VEREDICTO FINAL

No Apto Para Producción

Aunque el sistema demuestra un **desempeño funcional sólido** (92.86% de casos aprobados), la existencia de **1 vulnerabilidad crítica de severidad BLOCKER** (DF-SQ-01: conexión a base de datos sin contraseña) representa un **riesgo inaceptable de seguridad** que debe resolverse **obligatoriamente** antes de cualquier despliegue en ambiente productivo.

CONCLUSIONES Y REFLEXIONES

El módulo de autenticación de Wushi presenta un buen desempeño funcional con un 92.86% de casos aprobados; sin embargo, durante la revisión se identificó un hallazgo crítico de seguridad: la base de datos está configurada sin contraseña de acceso. Esto evidencia que un sistema puede "funcionar" correctamente en la interfaz y aun así mantener debilidades graves a nivel de infraestructura, lo que compromete la confidencialidad e integridad de la información. En consecuencia, la calidad del sistema no debe medirse únicamente por el cumplimiento de requisitos funcionales, sino también por su seguridad, confiabilidad y robustez.

Además, esta evaluación demostró que las pruebas funcionales por sí solas no siempre permiten detectar vulnerabilidades críticas. El problema fue identificado mediante análisis estático con SonarQube Cloud, lo que refuerza la necesidad de complementar el plan de pruebas con herramientas automatizadas y criterio técnico. Por ello, se concluye que el sistema no está listo para producción hasta implementar acciones inmediatas como asegurar el acceso a la base de datos, corregir la recuperación de contraseña para evitar enumeración de usuarios y realizar una auditoría general de configuración. A futuro, se recomienda integrar revisión de código, checklists de seguridad y análisis estático continuo desde las primeras etapas del desarrollo. La seguridad no puede ser una consideración posterior, sino un principio rector desde el diseño inicial del sistema.