ESTUDIA EN EL INSTITUTO TECNOLÓGICO DE LAS AMÉRICAS (ITLA)

Tarea: Proyecto Final

Materia: Programación III

Alumno: Gabriel Arnaldo Pozo Tavarez

Matricula: 2022 – 0416

Fecha: 17 – 04 – 2024

Cuatrimestre: Enero – abril 2024

Maestro: Kelyn Tejada Belliard

SecureLoginSystem

Surgió para satisfacer la creciente demanda de sistemas de autenticación seguros en un mundo conectado digitalmente. Este proyecto se centra en desarrollar un sistema de registro e inicio de sesión de usuarios altamente confiable utilizando tecnologías avanzadas como ASP.NET, C# y SQL Server.

Centrándose en la seguridad y la privacidad, SecureLoginSystem se esfuerza por establecer nuevos estándares para la verificación de identidad en línea para brindar a los usuarios una experiencia segura y sin complicaciones en sus interacciones digitales.

tecnología aplicada

El desarrollo front-end se basará en ASP.NET, una tecnología potente y ampliamente utilizada para crear interfaces web dinámicas y responsivas. Para implementar la lógica de back-end se utilizará el lenguaje de programación C#, conocido por su potencia y versatilidad en el desarrollo de aplicaciones y servicios web.

Para la gestión de datos se utilizará SQL Server, que es una plataforma confiable para el almacenamiento y recuperación eficiente de información en bases de datos relacionales. Todo el desarrollo se llevará a cabo dentro del entorno de desarrollo integrado de Visual Studio, proporcionando una experiencia de desarrollo completa y unificada.

El objetivo

El objetivo principal del proyecto SecureLoginSystem es desarrollar un sistema de inicio de sesión y registro de usuarios que garantice un alto grado de seguridad y eficiencia. El sistema se diseñará cuidadosamente para satisfacer las necesidades de autenticación específicas de las aplicaciones, brindando a los usuarios una experiencia segura y fluida al interactuar con la plataforma.

Al implementar las mejores prácticas de ciberseguridad, SecureLoginSystem tiene como objetivo proteger la información confidencial del usuario, como contraseñas y datos personales, de las amenazas cibernéticas. Además, intentaremos optimizar la eficiencia del proceso de autenticación, reducir al mínimo el tiempo de espera y los posibles errores.

El proyecto se centrará en el desarrollo de funciones básicas como el registro seguro de nuevos usuarios, un flujo de inicio de sesión sólido y gestión de perfiles de usuario. También prestaremos especial atención a la usabilidad y la experiencia del usuario, asegurando que la interfaz sea intuitiva y fácil de usar para personas de todos los niveles. En general, SecureLoginSystem tiene como objetivo proporcionar una solución integral y confiable para las necesidades de autenticación de aplicaciones que garantice la seguridad, la eficiencia y la disponibilidad en cada paso del proceso de inicio de sesión y registro del usuario.

alcance

El alcance del proyecto SecureLoginSystem se centra en implementar una funcionalidad básica para garantizar una autenticación de usuario segura y eficiente en las aplicaciones. Estas características incluyen:

Registro de Usuario: Se desarrollará un proceso de registro seguro para permitir a los usuarios crear nuevas cuentas proporcionando de manera confiable la información requerida y protegiendo los datos ingresados durante el proceso. Inicio de sesión: se implementará un mecanismo de inicio de sesión sólido para autenticar a los usuarios de manera segura y eficiente, verificar sus credenciales y brindar acceso a funciones protegidas de la aplicación. Además de estas características principales, el proyecto también se centrará en el diseño y desarrollo de una interfaz de usuario intuitiva y receptiva que facilite la navegación y la interacción para los usuarios finales. Prestaremos especial atención al diseño de los elementos visuales, la disposición de los elementos y el flujo de navegación para garantizar una experiencia consistente y placentera en todas las pantallas y dispositivos. Por último, pero no menos importante, se incorporarán fuertes medidas de seguridad en todas las áreas del sistema, desde el almacenamiento seguro de contraseñas hasta la protección de datos confidenciales en tránsito. Se utilizarán tecnologías y prácticas de seguridad avanzadas para proteger la información del usuario de las amenazas cibernéticas y garantizar la integridad y confidencialidad de los datos en todo momento.

Cronograma

Día 1: Planificación y Definición de Requisitos

• Definir los requisitos del proyecto en detalle.

• Diseñar la estructura de la base de datos y establecer relaciones entre entidades.

• Planificar las tareas necesarias para cada fase del proyecto.

Día 2-5: Desarrollo

• Implementar el frontend utilizando ASP.NET y diseñar las interfaces de usuario.

• Desarrollar la lógica del backend utilizando C# y establecer la comunicación con la base de datos.

• Integrar la lógica de negocio para gestionar el registro, inicio de sesión y seguridad del sistema.

Día 6-7: Pruebas y Despliegue

• Realizar pruebas de unidad para garantizar el funcionamiento correcto de cada componente.

• Realizar pruebas de integración para verificar la interacción entre frontend y backend.

• Ejecutar pruebas de aceptación para validar la funcionalidad global del sistema.

• Preparar el entorno de producción y realizar el despliegue de la aplicación en un servidor web.

• Realizar pruebas finales en el entorno de producción para asegurar la estabilidad y seguridad del sistema antes de lanzarlo al público.

Primer lanzamiento

La primera versión del proyecto SecureLoginSystem se centrará en proporcionar la funcionalidad básica que los usuarios necesitan para registrarse e iniciar sesión de forma segura y eficiente. La primera versión incluirá:

Registro de usuario: los usuarios pueden crear nuevas cuentas proporcionando la información requerida (como nombre de usuario, dirección de correo electrónico y contraseña). Se implementarán medidas de validación para garantizar la integridad de los datos de entrada.

Inicio de sesión: se desarrollará un mecanismo de inicio de sesión para permitir a los usuarios autenticarse de forma segura utilizando credenciales registradas previamente. Se utilizará tecnología de cifrado para proteger las contraseñas almacenadas en la base de datos.

Interfaz de usuario básica: contiene una interfaz de usuario funcional básica que proporciona a los usuarios un fácil acceso a las funciones de registro e inicio de sesión. La interfaz es intuitiva y fácil de usar, con elementos de diseño mínimos para una experiencia de usuario fluida.

Funciones mínimas de seguridad: Se implementarán funciones de seguridad básicas, como la prevención de ataques de fuerza bruta durante el proceso de inicio de sesión y la prevención de ataques de inyección SQL en el formulario de registro. Estas medidas proporcionarán una capa adicional de seguridad para proteger la integridad de los datos del usuario.

Requisitos del sistema para la primera versión.

La primera versión del proyecto SecureLoginSystem debería implementar las siguientes funciones y características:

Funcionalidad de registro e inicio de sesión: el sistema permitirá a los usuarios registrarse de forma segura proporcionando la información requerida, como nombre de usuario, dirección de correo electrónico y contraseña. Además, se desarrollará un mecanismo de inicio de sesión para autenticar a los usuarios utilizando credenciales registradas previamente.

Capacidad para cifrar contraseñas: se implementarán métodos de cifrado seguros para proteger las contraseñas almacenadas en la base de datos. Esto garantizará que las contraseñas de los usuarios estén protegidas contra el acceso no autorizado y la piratería.

Interfaz de usuario intuitiva y amigable: Se desarrollará una interfaz de usuario intuitiva y fácil de usar para brindar a los usuarios un fácil acceso a las funciones de registro e inicio de sesión. La interfaz será amigable y responsiva con elementos de diseño cuidadosamente seleccionados para mejorar la experiencia del usuario.

Además de estos requisitos principales, se prestará especial atención a la seguridad y estabilidad del sistema, garantizando que todas las funciones implementadas cumplen con los

estándares de calidad y seguridad establecidos. Además, antes del lanzamiento de la primera versión, se realizará una prueba exhaustiva para detectar y corregir posibles errores o vulnerabilidades.

Conclusiones

SecureLoginSystem es un hito importante en el desarrollo de sistemas de autenticación de seguridad para entornos web. Al implementar tecnologías líderes como ASP.NET, C#, SQL Server y Visual Studio, hemos creado una plataforma poderosa y confiable que brinda a los usuarios una experiencia de autenticación segura y fluida.

El proyecto aborda una necesidad fundamental del mundo digital actual: la protección de la identidad y la privacidad del usuario. A medida que la gente está cada vez más preocupada por la seguridad de la red y la propagación de amenazas en línea, SecureLoginSystem se ha convertido en una solución potente y eficaz para proteger la información confidencial de los usuarios.

A lo largo del desarrollo de SecureLoginSystem, hemos enfatizado la seguridad, la eficiencia y la facilidad de uso. Desde el cifrado de contraseñas hasta la implementación de medidas de seguridad avanzadas, cada aspecto del sistema está diseñado para garantizar la integridad y confidencialidad de los datos del usuario. Además, nos esforzamos por crear una interfaz intuitiva y fácil de usar que permita a personas de todos los niveles navegar e interactuar fácilmente. Creemos que una experiencia de usuario positiva es esencial para el éxito de cualquier aplicación y SecureLoginSystem no es una excepción.

B-Equipo Metodología Scrum:

Definir tareas a ejecutar:

- **Crear base de datos:** Diseñar la estructura de la base de datos para almacenar la información de usuarios.
- **Desarrollar frontend:** Implementar la interfaz de usuario utilizando ASP.NET para permitir el registro e inicio de sesión de usuarios.
- **Implementar backend:** Desarrollar la lógica de negocio y la capa de acceso a datos utilizando C# para gestionar el registro e inicio de sesión de usuarios.
- **Integrar seguridad:** Aplicar medidas de seguridad como encriptación de contraseñas y protección contra ataques de inyección SQL.
- **Realizar pruebas:** Probar todas las funcionalidades del sistema para asegurar su correcto funcionamiento y seguridad.
- **Desplegar aplicación:** Preparar el entorno de producción y lanzar la aplicación para su uso público.

Definir el equipo de trabajo:

- **Product Owner:** Kelyn Tejada Encargado de definir los requisitos del sistema.

- **Scrum Master:** Gabriel Pozo Responsable de facilitar el proceso Scrum y eliminar obstáculos que puedan afectar al equipo.

- **Equipo de Desarrollo:** Gabriel Pozo Integrado por desarrolladores frontend y backend.

Herramientas que usarían:

- **ZenHub**: Para gestionar el backlog y el progreso de las tareas durante el sprint, Para el seguimiento de las historias de usuario, asignación de tareas y registro de avances.
- **Visual Studio**: Para el desarrollo del frontend y backend utilizando ASP.NET y C# respectivamente.
- **SQL Server Management Studio**: Para el diseño y administración de la base de datos.
- **Github**: Para el control de versiones del código fuente y la colaboración en el desarrollo del software.

Definir las épicas:

- **Registro de usuarios:** Implementar la funcionalidad que permita a los usuarios crear cuentas en la plataforma.
- **Inicio de sesión:** Desarrollar el mecanismo de autenticación que permita a los usuarios acceder a sus cuentas.
- **Interfaz de usuario:** Crear una interfaz intuitiva y funcional que facilite el registro e inicio de sesión de los usuarios.
- **Seguridad:** Integrar medidas de seguridad robustas para proteger la información de los usuarios y prevenir ataques cibernéticos.

2- Plan de Pruebas

Realizar un plan de pruebas que contenga lo siguiente puntos, con al menos 20 casos de pruebas que sean obligatorios dentro de cualquier tipo de sistema (ejemplo Inicio de sesión de usuario, creación de usuarios, creación de roles, etc.): Importante (los casos de pruebas deben estar relacionados con las HU que fueron definidas para este proyecto.

1- Lista de requerimientos funcionales y no funcionales de acorde a las historias de usuarios. (mínimo de 10 HU).

Requerimientos Funcionales:

Requerimientos	Descripción
Los usuarios deben poder registrar nuevas cuentas proporcionando un nombre de usuario, dirección de correo electrónico y contraseña segura.	Los usuarios deben tener la capacidad de crear nuevas cuentas en la plataforma ingresando información básica como nombre de usuario, dirección de correo electrónico y una contraseña segura.
Los usuarios deben poder iniciar sesión en la plataforma utilizando su dirección de correo electrónico y contraseña.	Se requiere que los usuarios puedan acceder a sus cuentas utilizando su dirección de correo electrónico y la contraseña asociada.
Se deben implementar medidas de seguridad para proteger las contraseñas almacenadas en la base de datos.	El sistema debe utilizar técnicas de encriptación y almacenamiento seguro de contraseñas para proteger la información de los usuarios contra accesos no autorizados y ataques cibernéticos.
La interfaz de usuario debe ser intuitiva y fácil de usar, con elementos de diseño mínimos para una experiencia fluida.	La interfaz de usuario debe diseñarse de manera que sea fácil de entender y navegar para los usuarios, con un diseño limpio y funcional que garantice una experiencia de usuario fluida y agradable.
El sistema debe proteger contra ataques de fuerza bruta durante el inicio de sesión.	Deben implementarse medidas de seguridad para prevenir y mitigar posibles ataques de fuerza bruta, como limitar el número de intentos de inicio de sesión y bloquear temporalmente las cuentas después de múltiples intentos fallidos.
Se debe validar que los campos del formulario de registro se completen correctamente y que los datos ingresados sean válidos.	Antes de procesar el registro, el sistema debe validar que todos los campos del formulario estén completos y que los datos ingresados sean válidos y cumplan con los criterios especificados (por ejemplo, formato de correo electrónico válido).

Requerimientos No Funcionales:

Requerimientos No Funcionales	**Descripción**
El sistema debe ser seguro y resistente a ataques externos.	Se requiere que el sistema implemente medidas de seguridad robustas para proteger los datos de los usuarios y garantizar su integridad ante posibles amenazas y ataques externos.
La interfaz de usuario debe ser receptiva y compatible con diferentes dispositivos y navegadores web.	La interfaz de usuario debe diseñarse y desarrollarse de manera que sea compatible con una variedad de dispositivos (móviles, tabletas, computadoras) y navegadores webs populares para garantizar una experiencia consistente para todos los usuarios.
El sistema debe ser escalable para manejar un gran número de usuarios concurrentes.	El sistema debe estar diseñado y configurado para escalar eficientemente y manejar un alto volumen de tráfico y usuarios concurrentes sin degradación significativa del rendimiento o tiempos de respuesta.
El tiempo de respuesta del sistema debe ser rápido, garantizando una experiencia de usuario fluida.	Se espera que el sistema tenga tiempos de respuesta rápidos y eficientes para todas las interacciones de los usuarios, asegurando una experiencia de usuario fluida y sin demoras significativas.

2- Definir cuáles son los criterios de aceptación de las pruebas.

Criterios de Aceptación de las Pruebas:

1- Los casos de prueba deben cubrir todas las funcionalidades requeridas para el inicio de sesión y la creación de usuarios según las historias de usuario definidas en el proyecto SecureLoginSystem.

- Por ejemplo, los casos de prueba deben verificar que los usuarios puedan registrarse proporcionando un nombre de usuario único, una dirección de correo electrónico válida y una contraseña segura, y que puedan iniciar sesión utilizando sus credenciales previamente registradas.

2- Las pruebas deben garantizar que el sistema cumpla con todos los requisitos funcionales y no funcionales especificados en el proyecto SecureLoginSystem.

- Por ejemplo, las pruebas deben verificar que el sistema cumpla con los requisitos funcionales como la validación de datos de entrada en los formularios de registro e inicio de sesión, y con los requisitos no funcionales como la seguridad y el rendimiento del sistema.

3- Los casos de prueba deben cubrir exhaustivamente todas las funcionalidades requeridas para el inicio de sesión y la creación de usuarios según las historias de usuario definidas en el proyecto SecureLoginSystem.

- Cada caso de prueba debe verificar la funcionalidad principal de registro de usuario, incluyendo diferentes escenarios como registro exitoso, registro con campos inválidos, y registro con información duplicada.

- Cada caso de prueba debe abordar diferentes aspectos del inicio de sesión, como iniciar sesión con credenciales válidas, con credenciales inválidas, y con campos de formulario vacíos.

4- Las pruebas deben garantizar exhaustivamente que el sistema cumpla con todos los requisitos funcionales y no funcionales especificados en el proyecto SecureLoginSystem.

- Los casos de prueba deben validar todas las funciones requeridas en las historias de usuario, asegurando que cada característica funcione según lo esperado y que cumpla con los estándares de calidad definidos.

- Las pruebas deben evaluar el cumplimiento de los requisitos no funcionales como seguridad, rendimiento y usabilidad, verificando que el sistema sea seguro contra vulnerabilidades conocidas, tenga tiempos de respuesta aceptables y proporcione una experiencia de usuario intuitiva.

3- Definir cuáles son los criterios de rechazo en las pruebas.

Criterios de Rechazo en las Pruebas:

1- Si una funcionalidad no cumple con los criterios de aceptación especificados en las historias de usuario del proyecto SecureLoginSystem, la prueba correspondiente será rechazada.

- Por ejemplo, si una prueba de registro de usuario falla porque no se valida correctamente la dirección de correo electrónico, la prueba será rechazada hasta que se corrija este problema.

2- Si se identifican vulnerabilidades de seguridad o problemas de rendimiento durante las pruebas, el caso de prueba será rechazado hasta que se resuelvan los problemas.

- Por ejemplo, si una prueba de inicio de sesión revela una vulnerabilidad de seguridad como una posible inyección SQL, la prueba será rechazada y se requerirá una revisión adicional y corrección del código antes de volver a ejecutar la prueba.

3- Si una funcionalidad no cumple con los criterios de aceptación especificados en las historias de usuario del proyecto SecureLoginSystem, la prueba correspondiente será rechazada y se requerirá una revisión y corrección.

   - Si un caso de prueba de registro de usuario falla al validar la fortaleza de la contraseña, la prueba será rechazada y se requerirá una corrección para garantizar que se cumplan los criterios de aceptación definidos.

4- Si se identifican vulnerabilidades de seguridad o problemas de rendimiento durante las pruebas, el caso de prueba será rechazado y se requerirá una acción correctiva inmediata.

   - Si una prueba de inicio de sesión revela una vulnerabilidad de seguridad como una inyección de SQL, la prueba será rechazada y se requerirá una revisión y corrección del código antes de volver a ejecutar la prueba.

4- Comentar sobre las herramientas de pruebas que estarían usando y justificar respuesta.

Herramientas de Pruebas:

Utilizaremos las siguientes herramientas de pruebas para asegurar la calidad del proyecto SecureLoginSystem:

- **Selenium:** Se utilizará Selenium para realizar pruebas de interfaz de usuario automatizadas. Con Selenium, podemos simular acciones de usuarios reales en el navegador web, como hacer clic en botones, ingresar texto en campos de formulario y verificar elementos en la página.

- **Microsoft UnitTesting:** Se empleará este método para poder realizar dichas pruebas de automatización, probar diferentes endpoints, enviar diferentes tipos de datos y verificar las respuestas recibidas.

- `Selenium Edge`**:** Se utilizará el navegador de Microsoft Edge para poder poner a prueba nuestro sistema de automatización de registro de usuario e inicio de sesión.

5- Estimar el tiempo que duraría la ejecución de pruebas.

Basándonos en la complejidad del sistema y la cantidad de casos de prueba a ejecutar, estimamos que la ejecución de las pruebas tomará aproximadamente 1 semana.

Durante este período, se elaborará un cronograma detallado de trabajo para distribuir las pruebas de manera eficiente. Este cronograma incluirá la asignación de recursos, la programación de pruebas específicas para cada herramienta.

Cronograma

Día 1: Definición de Requerimientos y Configuración del Entorno

- Revisión detallada de los requerimientos del sistema.
- Configuración del entorno de desarrollo en Visual Studio.
- Instalación y configuración de Selenium para pruebas automatizadas.
- Desarrollo del esquema de base de datos en SQL Server y configuración inicial.

Día 2-3: Desarrollo del Sistema y Pruebas de Interfaz de Usuario

- Desarrollo del frontend utilizando ASP.NET para la interfaz de usuario de registro e inicio de sesión.
- Desarrollo del backend utilizando C# para la lógica de negocio.
- Implementación de la integración con la base de datos en SQL Server.
- Creación de casos de prueba automatizados con Selenium para verificar la funcionalidad del registro de usuario e inicio de sesión.
- Ejecución de las pruebas automatizadas y revisión de los resultados.

Día 4: Ajustes y Correcciones

- Corrección de errores y ajustes necesarios en el código para garantizar el funcionamiento correcto de la interfaz de usuario.
- Optimización del rendimiento y seguridad del sistema.
- Revisión de la integración de la base de datos y ajustes necesarios.

Día 5: Pruebas de Aceptación y Revisión Final

- Ejecución de pruebas manuales adicionales para validar el sistema.
- Revisión final de todas las funcionalidades del sistema y corrección de últimos detalles.
- Preparación del entorno de producción para el despliegue.

Día 6: Documentación y Preparación para Despliegue

- Documentación de los resultados de las pruebas y preparación de informes de calidad.
- Despliegue del sistema SecureLoginSystem en el entorno de producción y lanzamiento para uso público.

Día 7: Entrega del Proyecto y Soporte Inicial

- Preparación final para el despliegue del sistema en el entorno de producción.
- Documentación completa del proyecto, incluyendo manuales de usuario y detalles de implementación.
- Despliegue del sistema SecureLoginSystem en el entorno de producción y lanzamiento para uso público.
- Ofrecimiento de soporte técnico inicial para resolver posibles problemas post-despliegue.
- Evaluación de retroalimentación inicial de los usuarios y ajustes necesarios.

6- Elaborar plantillas para cada caso de pruebas.

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_01	Prueba de Registro de Usuario Exitoso	1. Acceder a la página de registro. 2.Completar el formulario con datos válidos. 3. Enviar el formulario de registro.	Nombre de usuario: "ejemplo_usuario", Correo electrónico: "usuario@example.com", Contraseña: "contraseña_segura"	El usuario se registra exitosamente y se redirige a la página de inicio con un mensaje de confirmación.	El usuario se registra Con los datos correctos lo envía directamente a hacer el login como lo esperado.

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_02	Prueba de Registro de Usuario con Datos Incompletos	1. Acceder a la página de registro. 2.Completar solo algunos campos del formulario. 3. Enviar el formulario de registro.	Nombre de usuario: "ejemplo_usuario", Correo electrónico: "", Contraseña: "contraseña_segura"	El sistema muestra mensajes de error indicando que los campos obligatorios están vacíos y no se registra el usuario.	Al momento de iniciar sesión con correo no creado, notifica que no se encontraron coincidencias

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_03	Prueba de Inicio de Sesión Exitoso	1. Acceder a la página de inicio de sesión. 2. Ingresar credenciales válidas (nombre de usuario y contraseña). 3. Hacer clic en el botón de inicio de sesión.	Nombre de usuario: "usuario_existente", Contraseña: "contraseña_correcta"	El usuario inicia sesión correctamente y se redirige a la página de inicio con un mensaje de bienvenida.	Al insertar el correo y contraseña correcta ingresa a la pagina de inicio como lo esperado.

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_04	Prueba de Inicio de Sesión con Credenciales Incorrectas	1. Acceder a la página de inicio de sesión. 2. Ingresar credenciales inválidas (nombre de usuario y contraseña). 3. Hacer clic en el botón de inicio de sesión.	Nombre de usuario: "usuario_existente", Contraseña: "contraseña_incorrecta"	El sistema muestra un mensaje de error indicando que las credenciales son incorrectas y no se permite el inicio de sesión.	Al intentar iniciar sesión con los datos erroneo le presenta una alerta de que están incorrectos los datos como se esperaba.

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_005	Prueba de Recuperación de Contraseña	1. Acceder a la página de recuperación de contraseña. 2. Ingresar correo electrónico registrado. 3. Hacer clic en el botón de enviar.	Correo electrónico: "usuario@example.com"	El sistema envía un correo electrónico al usuario con instrucciones para restablecer la contraseña.	Al momento de recuperar contraseña presenta error porque no tiene esa disponibilidad el sistema.

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_006	Prueba de Recuperación de Contraseña con Correo Electrónico No Registrado	1. Acceder a la página de recuperación de contraseña. 2. Ingresar un correo electrónico no registrado. 3. Hacer clic en el botón de enviar.	Correo electrónico: "correo_no_registrado@example.com"	El sistema muestra un mensaje de error indicando que el correo electrónico no está asociado a ninguna cuenta.	Al intentar esta acción no es permitida ya que ese método no se encuentra en el sistema.

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_007	Prueba de Longitud Máxima de Nombre de Usuario	1. Acceder a la página de registro. 2. Completar el formulario con un nombre de usuario que exceda la longitud máxima permitida. 3. Enviar el formulario de registro.	Nombre de usuario: "nombre_de_usuario_con_longitud_maxima_que_excede_el_limite_permitido_para_registrar"	El sistema muestra un mensaje de error indicando que el nombre de usuario excede el límite de caracteres permitidos y no se registra el usuario.	Al intentar crear un usuario y pone muchos caracteres hasta llegar al límite no le permitirá poner mas y eso es lo esperado.

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_008	Prueba de Longitud Máxima de Contraseña	1. Acceder a la página de registro. 2. Completar el formulario con una contraseña que exceda la longitud máxima permitida. 3. Enviar el formulario de registro.	Contraseña: "contraseña_con_longitud_maxima_que _excede_el_limite_permitido_para_registrar"	El sistema muestra un mensaje de error indicando que la contraseña excede el límite de caracteres permitidos y no se registra el usuario.	Al intentar crear un usuario y pone muchos caracteres hasta llegar al límite no le permitirá poner más y eso es lo esperado.

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_009	Prueba de Validación de Correo Electrónico	1. Acceder a la página de registro. 2. Completar el formulario con un correo electrónico inválido. 3. Enviar el formulario de registro.	Correo electrónico: "correo_invalido @ejemplo"	El sistema muestra un mensaje de error indicando que el correo electrónico ingresado no es válido y no se registra el usuario.	Al momento de registrar un usuario con un contexto de correo electrónico invalido, entra el error porque no se registra, pero no muestra el error de que no acepta ese tipo de correo.

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_010	Prueba de Seguridad de Contraseña	1. Acceder a la página de registro. 2. Completar el formulario con una contraseña débil. 3. Enviar el formulario de registro.	Contraseña: "123456"	El sistema muestra un mensaje de error indicando que la contraseña no cumple con los requisitos de seguridad y no se registra el usuario.	Este método da error porque el sistema no te valida ni te sugiere poner una contraseña más segura.

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_011	Prueba de Longitud Mínima de Nombre de Usuario	1. Acceder a la página de registro. 2. Completar el formulario con un nombre de usuario que no cumpla con la longitud mínima permitida. 3. Enviar el formulario de registro.	Nombre de usuario: "usr"	El sistema muestra un mensaje de error indicando que el nombre de usuario no cumple con el requisito de longitud mínima y no se registra el usuario.	El sistema en este caso funciona como lo esperado porque el mínimo de caracteres es 1 y por ende lo registra.

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_012	Prueba de Caracteres Especiales en Nombre de Usuario	1. Acceder a la página de registro. 2. Completar el formulario con un nombre de usuario que contenga caracteres especiales. 3. Enviar el formulario de registro.	Nombre de usuario: "user@123"	El sistema muestra un mensaje de error indicando que el nombre de usuario no puede contener caracteres especiales y no se registra el usuario.	El sistema no registra el usuario ya que contiene caracteres especiales, pero hay un error porque no notifica por qué pasa eso.

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_013	Prueba de Registro de Usuario con Correo Electrónico Duplicado	1. Acceder a la página de registro. 2. Completar el formulario con un correo electrónico que ya esté registrado en el sistema. 3. Enviar el formulario de registro.	Correo electrónico: "usuario@example.com"	El sistema muestra un mensaje de error indicando que el correo electrónico ya está registrado y no se registra el usuario nuevamente.	El sistema registra al usuario porque no tiene una validación de correo y no notifica que el correo este duplicado.

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_014	Prueba de Inicio de Sesión con Campos Vacíos	1. Acceder a la página de inicio de sesión. 2. Dejar los campos de nombre de usuario y contraseña vacíos. 3. Hacer clic en el botón de inicio de sesión.	Nombre de usuario: "", Contraseña: ""	El sistema muestra mensajes de error indicando que los campos obligatorios están vacíos y no se permite el inicio de sesión.	El sistema le arroja mensajes indicando que tiene que rellenar los campos vacíos como se esperaba.

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_015	Prueba de Cierre de Sesión	1. Iniciar sesión en el sistema. 2. Hacer clic en el botón de cierre de sesión.	Presionar el botón de cerrar sesión	El sistema cierra la sesión del usuario y lo redirige a la página de inicio con un mensaje de confirmación.	El sistema cierra la sesión y manda al usuario, pero no le genera un mensaje de confirmación.

Nombre del Caso de Prueba	Descripción	Pasos Para Seguir	Datos de Entrada	Resultados Esperados	Resultados Reales
CP_016	Prueba de Visualización de Contraseña	1. Acceder a la página de registro o inicio de sesión. 2. Verificar la opción para mostrar la contraseña. 3. Ingresar la contraseña.	Contraseña: "contraseña_visible"	La contraseña ingresada es visible en el campo correspondiente.	El sistema te permite ver la contraseña al momento de digitarla en el campo correspondiente.

7- Elaborar una plantilla con los equipos de pruebas y sus responsabilidades.

Equipo de Pruebas y Responsabilidades:

Rol	Responsabilidades
QA Engineer	- Diseñar casos de prueba detallados para cubrir todas las funcionalidades del sistema SecureLoginSystem.
	- Desarrollar y mantener scripts de prueba automatizados utilizando Selenium Web Driver para la prueba de interfaz de usuario.
	- Ejecutar pruebas manuales y automatizadas para identificar defectos y asegurar la calidad del software.
	- Documentar exhaustivamente los resultados de las pruebas, incluyendo informes de errores y mejoras sugeridas.
	- Colaborar estrechamente con el desarrollador para comprender los requisitos y el diseño del sistema y asegurar una cobertura completa de pruebas.
	- Realizar pruebas de regresión para garantizar que los cambios no introduzcan nuevos problemas en el sistema.
	- Proporcionar soporte técnico y solucionar problemas relacionados con la calidad del software durante todo el ciclo de vida del proyecto.

8- Elaborar un plan de automatización de pruebas.

Plan de Automatización de Pruebas:

Definición de Alcance:

- Se automatizarán las pruebas de interfaz de usuario y se enfocarán en verificar las funcionalidades principales del registro de usuario e inicio de sesión.
- No se automatizarán las pruebas de API ni de rendimiento en esta etapa debido a las limitaciones del proyecto y la priorización de las pruebas de interfaz de usuario.

Selección de Herramientas:

- Se utilizará Selenium Web Driver para automatizar las pruebas de interfaz de usuario. Esta herramienta es adecuada para simular las acciones de un usuario real en un navegador web y verificar la funcionalidad de la aplicación web.

Definición de Casos de Prueba:

- Se identificarán los casos de prueba más críticos y frecuentemente ejecutados para la automatización.
- Los casos de prueba se centrarán en validar la funcionalidad de registro de usuario e inicio de sesión, incluyendo la verificación de la validación de datos, la seguridad de la contraseña y la navegación entre páginas.

Desarrollo de Scripts de Prueba:

- Se escribirán scripts de prueba utilizando el lenguaje de programación C# y el framework de Selenium Web Driver.
- Los scripts de prueba se desarrollarán para ejecutar las acciones necesarias en la interfaz de usuario, como completar formularios, hacer clic en botones y verificar elementos en la página.

Integración en el Proceso de Desarrollo:

- Los scripts de prueba automatizados se integrarán en el proceso de desarrollo utilizando un enfoque de desarrollo dirigido por pruebas (TDD) o integración continua (CI).
- Los scripts de prueba se ejecutarán automáticamente como parte de las compilaciones regulares del código para detectar y corregir rápidamente cualquier regresión en la funcionalidad.

Ejecución y Mantenimiento:

- Los scripts de prueba automatizados se ejecutarán regularmente como parte de las pruebas de regresión para garantizar que el sistema continúe funcionando correctamente después de cada cambio.
- Se realizarán actualizaciones y mantenimiento continuo de los scripts de prueba para mantener su relevancia y eficacia a lo largo del ciclo de vida del proyecto.