

2025-10-17

lunes, 15 de diciembre de 2025 01:16 a. m.

Tema: Fases previas a la Codificación en el SDLC**Objetivos:**

- Comprender la estructura del Ciclo de Vida de Desarrollo de Software (SDLC) y la relevancia de sus fases para asegurar la calidad del producto.
- Ejecutar un ejercicio práctico aplicando el SDLC a una funcionalidad de Inicio de Sesión, abarcando desde la definición de requisitos hasta la planificación de pruebas.
- Redactar historias de usuario claras que especifiquen los requerimientos funcionales y sus respectivos criterios de aceptación.
- Crear prototipos visuales que permitan validar el diseño de la interfaz antes de escribir el código.

Desarrollo:**El Ciclo de Vida (SDLC)**

Se analizó el SDLC como el marco metodológico que orquesta la creación de software, abarcando desde la concepción de la idea hasta su mantenimiento operativo. Este proceso se fundamenta en cuatro etapas críticas para asegurar la calidad: Análisis (definición del problema y requisitos), Diseño (arquitectura y planos del sistema), Codificación (construcción lógica) y Pruebas (validación funcional).

Fase de Análisis: Historia de Usuario

Para la funcionalidad de "Login", se capturaron los requisitos mediante una Historia de Usuario (HU-01) bajo la metodología ágil. Se estableció la necesidad del usuario registrado de acceder mediante credenciales (correo y contraseña). Un componente vital fue la definición de los Criterios de Aceptación, los cuales estipulan que el sistema debe validar la existencia del usuario, manejar errores con mensajes claros, redirigir al dashboard tras un éxito y ofrecer una ruta de recuperación de contraseña.

Fase de Diseño: Prototipado

Antes de escribir cualquier línea de código, se procedió a la creación de un Mockup o prototipo visual de la interfaz. Este diseño incluyó la disposición estratégica de los elementos definidos en el análisis, como las cajas de texto y los botones de acción principal. Esta etapa es crucial para validar visualmente la solución y prevenir retrabajos, asegurando que el diseño satisfaga la experiencia de usuario antes de la implementación técnica.

Fase de Pruebas: Definición de Casos

Para garantizar la integridad y robustez del software antes de su liberación, se diseñó una estrategia de validación compuesta por cinco escenarios de prueba específicos:

Caso de Prueba 1: Validación de acceso exitoso

- *Objetivo: Verificar el flujo normal de autenticación.*
- *Precondición: El usuario debe existir en la base de datos.*
- *Datos de prueba: Credenciales válidas (usuario@ejemplo.com / Contra123).*
- *Procedimiento: Ingresar al formulario, completar los campos con datos correctos y ejecutar la acción de ingreso.*
- *Resultado esperado: El sistema autentica al usuario y lo redirige correctamente a la vista principal (dashboard).*

Caso de Prueba 2: Gestión de credenciales inválidas

- Objetivo: Validar la seguridad ante intentos de acceso erróneos.
- Datos de prueba: Correo válido pero contraseña incorrecta (aaaa@gmail.com / *****).
- Procedimiento: Intentar iniciar sesión con la combinación errónea.
- Resultado esperado: El sistema deniega el acceso y presenta una notificación de error clara ("Correo o contraseña incorrectos").

Caso de Prueba 3: Control de campos obligatorios

- Objetivo: Verificar la integridad de los datos de entrada.
- Procedimiento: Dejar los campos de correo o contraseña vacíos.
- Resultado esperado: El botón de "Iniciar sesión" debe permanecer deshabilitado o inactivo hasta que se complete la información requerida.

Caso de Prueba 4: Usabilidad en visibilidad de contraseña

- Objetivo: Comprobar la funcionalidad de privacidad del campo de contraseña.
- Procedimiento: Ingresar texto en el campo de contraseña e interactuar con el ícono de visibilidad (ojito).

- Resultado esperado: El texto debe alternar entre formato legible y oculto (enmascarado) con cada interacción.

Caso de Prueba 5: Navegación de recuperación

- Objetivo: Validar el flujo de soporte al usuario.
- Procedimiento: Acceder al hipervínculo "¿Olvidaste tu contraseña?".
- Resultado esperado: La aplicación debe redirigir al usuario al formulario específico para el restablecimiento de credenciales.

Definition of ready

Este concepto se definió como el estado de "preparación" de un requerimiento. Una historia de usuario cumple con el Definition of Ready cuando ha sido completamente analizada y cuenta con todos los criterios y contextos necesarios para que el equipo de desarrollo pueda iniciar la codificación sin ambigüedades.

Definition of done

Se estableció como el criterio de finalización estricto. Una tarea solo alcanza el Definition of Done cuando ha superado el ciclo completo: codificación finalizada, revisión de código (code review), cumplimiento de todos los criterios de aceptación y aprobación exitosa de las pruebas en un entorno controlado.

Conclusiones:

- La implementación de fases estructuradas (Análisis, Diseño, Codificación, Pruebas) otorga gobernanza al proyecto, asegurando que cada etapa contribuya valor tangible al objetivo final.
- La inversión de tiempo en la planificación temprana, mediante historias de usuario detalladas y prototipos, reduce drásticamente la deuda técnica y los errores en fases avanzadas.
- La coherencia integral entre el diseño visual, los requisitos funcionales y los casos de prueba garantiza la entrega de un software confiable y alineado con las necesidades reales del negocio.

Bibliografía:

- K. Schwaber y J. Sutherland, "La Guía de Scrum: La guía definitiva de Scrum: Las reglas del juego," Scrum.org, nov. 2020. [En línea]. Disponible: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>
- Atlassian, "Historias de usuario | Ejemplos y plantilla," Atlassian Agile Coach. [En línea]. Disponible: <https://www.atlassian.com/es/agile/project-management/user-stories>