DESARROLLO DE INTERFACES
2º DAM

Unidad 9

ALFONSO RINCÓN | JOSÉ CORROCHANO | GUILLERMO CECA

Nombre: Alfonso Rincón Cuerva, José Corrochano Pardo, Guillermo Ceca Illán

ÍNDICE

Nombre: Alfonso Rincón Cuerva, José Corrochano Pardo, Guillermo Ceca Illán

ACTIVIDAD 1

¿Qué tareas son, a tu juicio, las que se llevan a cabo en las distintas fases de un proyecto de implantación?

¿En qué fase se encuadraría la depuración del código?

1. *Análisis y Planificación*: en esta fase se definen los objetivos del proyecto y se identificar los requisitos.

2. *Diseño del Sistema:* se define la arquitectura, la base de datos y los diagramas necesarios.

3. *Desarrollo y Codificación*: se escribe el código y se realizan pruebas unitarias.

4. *Pruebas y Depuración:* se trata de identificar y corregir los errores del código. También se realizan pruebas unitarias, de integración y funcionales.

5. *Implementación y Despliegue:* se configuran los servidores y los entornos de producción.

6. *Mantenimiento y Soporte:* consiste en la optimización y mejoras continuas.

DEPURACIÓN: esta se lleva a cabo principalmente en las fases de pruebas y depuración, la cual viene después del desarrollo. También podemos hacerla en la fase de mantenimiento, en el caso de que se detecten errores y tengamos que corregirlos.

ACTIVIDAD 2

¿QUÉ TIPOS DE PRUEBAS DE REGRESIÓN CONOCES? INDICA ALGÚN EJEMPLO ILUSTRATIVO DE LAS MISMAS.

Las pruebas de regresión son un tipo de prueba de software que verifica que las modificaciones recientes en el código no han afectado negativamente las funcionalidades existentes del sistema. Se utilizan para asegurar que las nuevas actualizaciones, correcciones de errores o mejoras no introduzcan defectos en partes del software que ya funcionaban correctamente.

Pruebas de Regresión Unitarias

Se enfocan en probar módulos individuales después de que se han realizado cambios en el código. Por jemplo, un desarrollador actualiza una función matemática en una calculadora para mejorar la precisión. Se ejecutan nuevamente las pruebas unitarias para verificar que las operaciones básicas siguen funcionando correctamente.

Pruebas de Regresión Parcial

Solo se prueban los módulos o funcionalidades específicas donde se hicieron cambios, junto con las áreas más cercanas que puedan verse afectadas. Por ejemplo, Se cambia la interfaz de usuario en una aplicación de comercio electrónico, específicamente en la

página de pagos. Se realizan pruebas en la página de pagos y en las funcionalidades relacionadas, como la confirmación de compra y la generación de facturas.

Pruebas de Regresión Completa

Se ejecutan todas las pruebas del sistema para verificar que ningún cambio ha afectado al software en general. Por ejemplo, después de una actualización importante en el núcleo de un sistema operativo, se vuelven a ejecutar todas las pruebas automatizadas y manuales para asegurarse de que todo el sistema sigue funcionando correctamente.

Pruebas de Regresión Automatizadas

Se utilizan herramientas de automatización para ejecutar casos de prueba previamente definidos, lo que ahorra tiempo y esfuerzo en comparación con las pruebas manuales. Por ejemplo, en un sitio web de banca en línea, se utiliza Selenium para automatizar pruebas de inicio de sesión, transferencias y pagos después de actualizar el módulo de seguridad.

Pruebas de Regresión Manuales

Los testers ejecutan manualmente los casos de prueba críticos para verificar que el sistema funciona correctamente después de un cambio. Por ejemplo, después de modificar la lógica de descuentos en un software de ventas, un tester manualmente realiza pruebas comprando diferentes productos con descuentos para verificar que se aplican correctamente.

Pruebas de Regresión Basadas en Caso de Uso

Se prueban escenarios específicos de uso del software para asegurarse de que las funciones clave siguen funcionando correctamente. Por ejemplo, en una aplicación de reservas de vuelos, se prueba el flujo completo de un usuario desde la selección de destino hasta la confirmación del boleto después de modificar la integración con una nueva pasarela de pago.

ACTIVIDAD 3
¿QUÉ CARACTERÍSTICAS DEBEN CUMPLIR LAS PRUEBAS FUNCIONALES SOBRE UNA APLICACIÓN? DESCRÍBELAS E INDICA ALGÚN EJEMPLO

Las pruebas funcionales tienen como objetivo verificar que una aplicación cumpla con los requerimientos especificados y que todas sus funciones trabajen correctamente. Se centran en qué hace el sistema, sin importar cómo lo hace. Las principales características de estas pruebas son las siguientes:

- Exactitud. Se verifica que la aplicación haga exactamente lo que se espera según los requisitos. En una aplicación de banca móvil, si un usuario transfiere $100 a otra cuenta, el sistema debe reflejar exactamente esa transacción en ambos balances.
- Integridad. Se asegura que todas las funciones de la aplicación estén presentes y operativas. En una tienda en línea, la función de agregar productos al carrito, aplicar descuentos y procesar pagos debe funcionar correctamente sin omitir ningún paso.

- Interfaz de Usuario. Se revisa que los elementos gráficos sean visibles, accesibles y funcionales según lo esperado. En una aplicación de reservas de vuelos, los botones de selección de fechas y confirmación deben ser claramente visibles y operar sin fallos.
- Compatibilidad. La aplicación debe funcionar en diferentes dispositivos, navegadores y sistemas operativos. Un servicio de mensajería debe ser probado en iOS, Android y versión web para verificar que los mensajes se envían y reciben sin problemas en todas las plataformas.
- Seguridad. Se comprueba que la aplicación proteja los datos del usuario y prevenga accesos no autorizados. En una aplicación bancaria, al ingresar la contraseña incorrecta tres veces, el sistema debe bloquear la cuenta temporalmente para evitar intentos de acceso no autorizado.
- Manejo de Errores. La aplicación debe gestionar los errores de manera adecuada, mostrando mensajes claros sin afectar el funcionamiento general. Si un usuario introduce un número de tarjeta inválido al realizar una compra, el sistema debe mostrar un mensaje como "Número de tarjeta incorrecto. Intente nuevamente." en lugar de bloquear la aplicación.
- Flujo de Trabajo. Se evalúa que los procesos dentro de la aplicación sigan una lógica clara y secuencial. En una aplicación de citas médicas, el usuario debe poder seleccionar un médico, elegir una fecha y confirmar la cita sin saltarse pasos o encontrar interrupciones en el proceso.
- Funcionalidad Bajo Diferentes Condicionales. Se prueba el comportamiento de la aplicación en distintas situaciones, como una conexión a internet inestable o datos incorrectos. En una aplicación de mapas, si un usuario pierde la señal de internet mientras busca una ruta, el sistema debe mostrar la última información disponible en lugar de cerrarse inesperadamente.

ACTIVIDAD 4
SEGÚN TU CRITERIO, ¿POR QUÉ ES IMPORTANTE QUE SE SIGA EL DESARROLLO DE PRUEBAS INDICADO EN LA FIGURA ANTERIOR?

Es importante porque permite detectar errores en diferentes etapas del ciclo de vida del software, minimizando el impacto de fallos en producción. Algunas de las razones claves son:

- Detección temprana de errores. Permite identificar y corregir fallos antes de que afecten al producto final.
- Reducción de costos. Solucionar problemas en fases avanzadas del desarrollo suele ser mas caro que hacerlo en las etapas iniciales.
- Mejora en la calidad del software. Un proceso de pruebas bien definido garantiza que el software funcione correctamente en distintos escenarios.
- Cumplimiento de lo requisitos. Se asegura que el producto final cumple con las especificaciones y necesidades del cliente.

ACTIVIDAD 5

¿Has utilizado alguna vez alguna versión beta de alguna aplicación?

¿Conoces algún ejemplo de uso de estas?

Si que he utilizado la versión beta de aplicaciones como: Brawl Stars, call of duty: black ops 4, Fortnite.

- Brawl Stars Beta. Supercell lanzó una versión beta del juego antes de su lanzamiento global en 2018. En esta fase, se probaron diferentes modos de juego, mecánicas de combate y ajustes de balance de personajes.
- Call of Duty: Black Ops 4 beta. Activision lanzó una beta cerrada para el multijugador y una beta abierta para el modo **Blackout** (su versión de battle royale).
- Fortnite Beta. **Fortnite Battle Royale** se lanzó originalmente en fase beta, lo que permitió a Epic Games recopilar datos sobre el rendimiento del juego, equilibrar armas y mecánicas, y mejorar la optimización en distintas plataformas.

ACTIVIDAD 6

Es completamente imprescindible para abordar cualquier proyecto el realizar una secuencia de fases bien delimitadas que tengan como resultado un producto exitoso o el menos adecuado a las especificaciones del cliente. En este ejercicio se pide que se desarrolle un manual de pasos en el que describas la estrategia de diseño de la aplicación, analizando los diferentes pasos o etapas que lo componen.

El manual de pasos para la estrategia de diseño de una aplicación es el siguiente:

1. Análisis y planificación
    - Definir el objetivo de la aplicación.
    - Identificar los requisitos funcionales y no funcionales.
    - Analizar el público objetivo.
    - Crear un plan de desarrollo con plazos y recursos.
2. Diseño del sistema.
    - Especificación de la arquitectura del software.
    - Diseño de la interfaz de usuario y experiencia de usuario.
    - Creación de diagramas de flujo y casos de uso.
    - Selección de tecnologías y herramientas.
3. Desarrollo
    - Programación de las funcionalidades principales siguiendo metodologías ágiles.
    - Implementación de bases de datos y conexiones necesarias.
    - Integración de APIs o servicios externos.
    - Creación de pruebas unitarias durante el desarrollo.
4. Pruebas y depuración
    - Realización de pruebas unitarias para validar componentes individuales.
    - Pruebas de integración para verificar la comunicación entre módulos.
    - Pruebas de sistema para evaluar la funcionalidad global

- Depuración y corrección de errores.

5. Implementación y despliegue
   - Configuración del entorno de producción.
   - Pruebas finales en el entorno real.
   - Despliegue de la aplicación en servidores o tiendas de aplicaciones.
   - Monitoreo inicial para detectar posibles fallos.
6. Mantenimiento y actualizaciones
   - Corrección de errores detectados tras el lanzamiento.
   - Optimización del rendimiento y seguridad.
   - Implementación de mejoras y nuevas funcionalidades.
   - Soporte técnico y atención a los usuarios.

ACTIVIDAD 7

Tras describir la secuencia de fases necesarias para el desarrollo de una aplicación, es necesario centrarse en la fase de pruebas o de evaluación de la aplicación. Se pide diseñar un plan de pruebas en el que indiques de forma estimada la carga aproximada de tiempo que se van a asignar a cada una de las mismas en un desarrollo de una aplicación software. Esto es importante, puesto que siempre se va a trabajar con plazos de entrega, que han de estar convenientemente estimados y acotados.

Este desarrollo irá entorno una duración de 3-6 meses. Seguiremos la siguiente distribución:

1. *Pruebas Unitarias (2-3 semanas):* se realizan sobre componentes individuales (clases, métodos, funciones), utilizando Frameworks como **JUnit**.
2. *Pruebas de Integración (1-2 semanas):* se evaluará la interacción entre módulos o servicios.
3. *Pruebas de Sistema (1-2 semanas):* en estas pruebas, se va a evaluar el comportamiento completo del sistema.
4. *Pruebas de Rendimiento (1-2 semanas):* en esta fase, vamos a medir los tiempos de respuesta y comportamiento bajo carga de nuestra aplicación, como por ejemplo podría ser simular que 1000 usuarios están accediendo a la aplicación al mismo tiempo.
5. *Pruebas de Seguridad (1-2 semanas):* se buscarán vulnerabilidades en el sistema, como un usuario intentando acceder a otro sin permisos.
6. *Pruebas de Usabilidad (1-2 semanas):* se evaluará la experiencia del usuario.
7. *Pruebas de Aceptación del Usuario (2-3 semanas):* finalmente, una vez se hayan revisado todos los aspectos de la aplicación, se hará una validación final por el cliente.
8. *Pruebas de Regresión (1 semana):* se repetirán pruebas después de las correcciones para asegurar que todo está en orden.

ACTIVIDAD 8

Se propone realizar un listado de todas las pruebas que se van a realizar para evaluar una aplicación web que hemos desarrollado recientemente, según el encargo realizado por una compañía de venta online de material de oficina, con el fin de poder vender sus productos también de manera online. Lo aconsejable en este caso sería realizar un listado de todas las pruebas que se van a llevar a cabo para evaluar la aplicación (unitarias, de integración, de sistema). Posteriormente, se elaborará un calendario de tiempos de ejecución, que se adapte a las características de cada prueba.

Para garantizar el correcto funcionamiento de la aplicación web, se realizarán diferentes tipos de pruebas, desde pruebas unitarias hasta pruebas de sistema. A continuación, se presenta un listado detallado de las pruebas a ejecutar:

Pruebas Unitarias

Las pruebas unitarias se centran en evaluar componentes individuales de la aplicación para garantizar su correcto funcionamiento. Se suele evaluar lo siguiente:

- Gestión de usuarios: Registro, inicio de sesión, recuperación de contraseña.
- Gestión de productos: Creación, edición, eliminación y visualización de productos.
- Carrito de compras: Agregar, modificar y eliminar productos del carrito.
- Procesamiento de pedidos: Generación de órdenes, cálculo de totales, aplicación de descuentos y métodos de pago.
- Notificaciones: Envío de correos electrónicos y confirmaciones de pedidos.
- Búsqueda y filtrado de productos: Verificación de funcionalidades de búsqueda, filtros por categoría y rango de precios.

Pruebas de Integración

Estas pruebas validan la interacción entre diferentes módulos de la aplicación. Se suele evaluar lo siguiente:

- Inicio de sesión e interacción con la base de datos: Validación de credenciales y acceso seguro a los datos del usuario.
- Flujo de compra completo: Desde la selección de productos hasta la confirmación del pedido.
- Integración con pasarelas de pago: PayPal, tarjetas de crédito/débito, transferencias bancarias.
- Interacción con proveedores y stock: Sincronización de inventario con proveedores externos.
- Envío de correos automáticos: Confirmación de pedidos, notificaciones de entrega y actualización del estado del pedido.
- Integración con sistemas de facturación: Generación de facturas electrónicas.

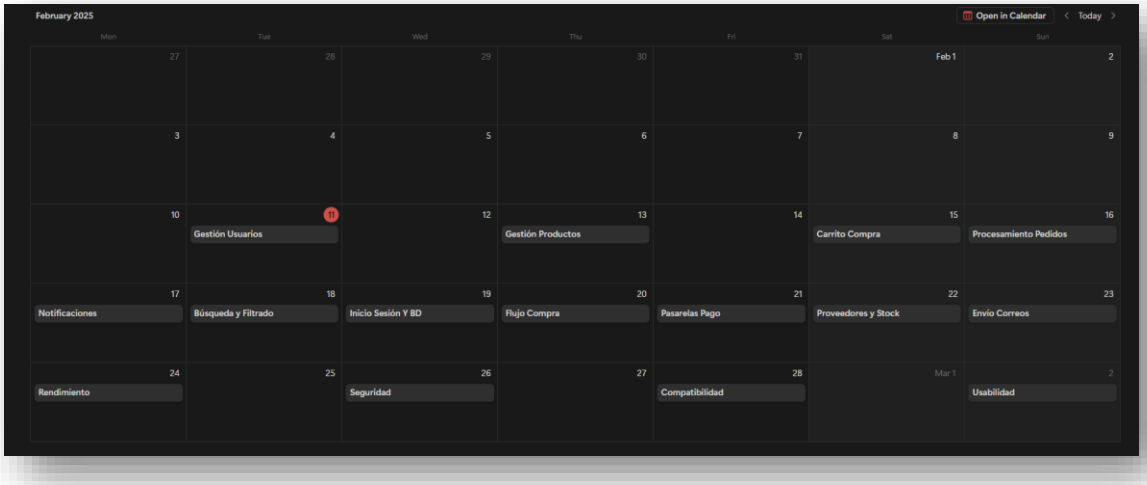
Pruebas de Sistema

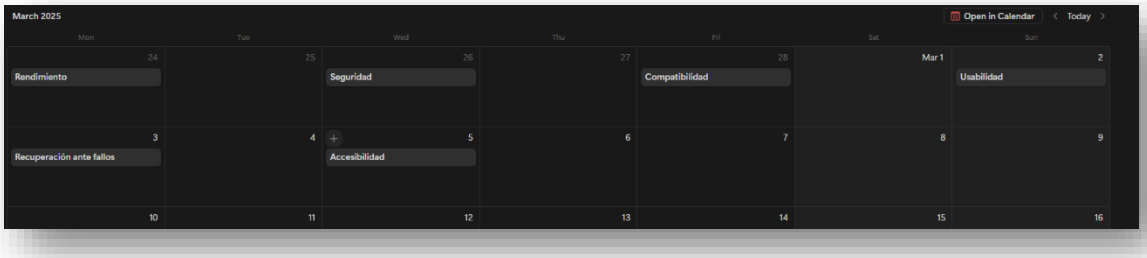
Se evalúa el sistema en su conjunto para verificar el correcto funcionamiento en un entorno real. Se suele evaluar lo siguiente:

- Pruebas de rendimiento. Evaluación de carga con múltiples usuarios simultáneos. Tiempo de respuesta del servidor.

- Pruebas de seguridad. Ataques de inyección SQL. Pruebas de fuerza en autenticación. Protección de datos personales y cumplimiento de normativas.
- Pruebas de compatibilidad. Navegadores y distintos dispositivos móviles con resoluciones específicas.
- Pruebas de usabilidad. Evaluación de experiencia de usuario. Pruebas con usuarios reales.
- Pruebas de recuperación ante fallos. Simulación de caída del servidor y recuperación de datos. Manejo de errores en base de datos y tiempo de inactividad.
- Pruebas de accesibilidad. Evaluación de cumplimiento con estándares WCAG. Uso de lectores de pantalla y navegación con teclado.

A continuación, muestro un ejemplo de las distintas pruebas que he mencionado, organizadas en una línea de tiempo o calendario:





Con esta planificación estaríamos casi un mes realizando las distintas pruebas necesarias.

BIBLIOGRAFÍA

Candil Rodríguez, R. G. (2024). Unidad 9: Desarrollo de interfaces. IES Ribera del Tajo.