



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

Aplicaciones Distribuidas

DOCENTE: Mgs. Dario Morales

Planificación de pruebas

GRUPO 7

NRC: 14930

Mayo 24 - Sep 24

Tabla de Planificación de Pruebas

Fase	Tipo de Prueba	Objetivo	Componentes Involucrados	Herramientas	Descripción
Fase 1	Pruebas Unitarias	Validar el funcionamiento individual de métodos y funciones	Servicios (CursoServiceImpl), Repositorios (CursoRepository)	JUnit, Mockito	Escribir y ejecutar pruebas unitarias para cada método. Simular dependencias usando mocks.
Fase 2	Pruebas de Integración	Validar la interacción entre módulos	Servicios y Repositorios, Microservicios (UsuarioClientRest)	JUnit, Spring Test, H2 Database	Probar la interacción entre CursoServiceImpl y CursoRepository, y entre CursoServiceImpl y UsuarioClientRest.
Fase 3	Pruebas de Extremo a Extremo (E2E)	Validar el funcionamiento de todo el sistema de principio a fin	Front-End (Angular), Back-End (Spring Boot)	Selenium, Cypress	Probar la interacción completa entre la interfaz Angular y los microservicios de Spring Boot.
Configuración	Entorno de Pruebas	Preparar el entorno para ejecutar las pruebas	Base de Datos, Configuración de Spring Boot	H2 Database, Spring Profiles	Configurar una base de datos en memoria para pruebas, utilizar perfiles de Spring Boot.

Detalle de las Pruebas

Fase 1: Pruebas Unitarias

Método a Probar	Descripción de la Prueba
listar()	Verificar que se lista correctamente todos los cursos.
porId(Long id)	Verificar que se obtiene un curso por su ID.
guardar(Curso curso)	Verificar que se guarda un curso correctamente.
eliminar(Long id)	Verificar que se elimina un curso por su ID.

agregarUsuario(Usuario usuario, Long idCurso)	Verificar que se agrega un usuario a un curso correctamente.
crearUsuario(Usuario usuario, Long idCurso)	Verificar que se crea y agrega un usuario a un curso correctamente.
eliminarUsuario(Usuario usuario, Long idCurso)	Verificar que se elimina un usuario de un curso correctamente.

Fase 2: Pruebas de Integración

Interacción a Probar	Descripción de la Prueba
CursoServiceImpl y CursoRepository	Verificar que los métodos del servicio interactúan correctamente con el repositorio.
CursoServiceImpl y UsuarioClientRest	Verificar que los métodos del servicio interactúan correctamente con los microservicios de usuarios.

Fase 3: Pruebas de Extremo a Extremo (E2E)

Funcionalidad a Probar	Descripción de la Prueba
Inscripción de Usuarios a Cursos	Verificar que un usuario puede inscribirse a un curso correctamente desde la interfaz Angular hasta el backend.
Eliminación de Usuarios de Cursos	Verificar que un usuario puede eliminarse de un curso correctamente desde la interfaz Angular hasta el backend.

Reportes de Pruebas

Fase	Tipo de Reporte	Contenido
Fase 1	Reporte de Pruebas Unitarias	Resultados de las pruebas unitarias, cobertura de código, casos de prueba fallidos y exitosos.
Fase 2	Reporte de Pruebas de Integración	Resultados de las pruebas de integración, interacciones correctas e incorrectas, correcciones necesarias.
Fase 3	Reporte de Pruebas E2E	Resultados de las pruebas de extremo a extremo, flujo completo de la aplicación, casos de prueba fallidos y exitosos.

Conclusión

En conclusión, al realizar una planificación exhaustiva de pruebas para la aplicación, se ha asegurado que cada componente, desde el servicio y el repositorio hasta la interacción completa entre el front-end en Angular y el back-end en Spring Boot, funcione correctamente. Las pruebas unitarias, de integración y de extremo a extremo no solo han permitido identificar y corregir errores en las primeras etapas del desarrollo, sino que también han proporcionado confianza en la robustez y fiabilidad de la aplicación. Esta planificación detallada ha ayudado a cubrir todas las bases necesarias para entregar un producto de alta calidad.