



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



**UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
INVESTIGACION EN INGENIERIA DE SOFTWARE**

Nombre: Danna Valentina Andrade Lucio.

Fecha: 03/02/2026.

NRC: 28482

Carrera: Ingeniería en Software.

INTRODUCCIÓN

Este documento recopila la interacción con el Asistente de IA (Gemini) utilizada para el análisis, desarrollo y corrección del ejercicio de replicación C1. Se detallan los prompts utilizados para estructurar las pruebas y solucionar errores de entorno.

INTERACCIÓN 1: Análisis del Planteamiento y Estrategia

PROMPT DEL ESTUDIANTE:

Se va a realizar la siguiente actividad con la instrucción (Confeccione las pruebas de unidad usando JUnit. Para ello, debe utilizar el LLM indicado por el profesor. Utilice el interfaz web, ya que deberá aportar una copia de la conversación que ha mantenido con el LLM.), para la cual se necesita cumplir con lo que se especifica dentro del archivo adjunto(Se adjunto el documento de especificaciones junto con la imagen del diagrama de clases)

RESPUESTA GENERADA POR LA IA:

He desarrollado un conjunto completo de pruebas unitarias para la clase File usando JUnit 5:

Estructura de Tests:

1. Tests del Constructor:

- testConstructor_ContentNotNull: Verifica que content no sea null después de la construcción
- testConstructor_ContentIsEmpty: Verifica que content esté vacío después de la construcción

2. Tests de addProperty():

- testAddProperty_ValidContent: Verifica que se añaden caracteres correctamente a un archivo PROPERTY
- testAddProperty_NullContent: Verifica que se lanza InvalidContentException cuando content es null
- testAddProperty_ImageType: Verifica que se lanza WrongFileTypeException cuando el tipo es IMAGE
- testAddProperty_MultipleAdditions: Verifica que múltiples llamadas acumulan el contenido
- testAddProperty_EmptyArray: Verifica el comportamiento con un array vacío

3. Tests de getCRC32():

- testGetCRC32_EmptyContent: Verifica que devuelve 0 cuando content está vacío
- testGetCRC32_WithContent: Verifica que calcula CRC32 con contenido válido
- testGetCRC32_ByteConversion: Verifica la conversión correcta de caracteres a bytes
- testGetCRC32_UTF8Range: Prueba con caracteres en el rango [0, 255]

4. Tests de Integración:

- testIntegration_PropertyFileWorkflow: Prueba un flujo completo de trabajo con archivos PROPERTY
- testSetType_Property: Verifica el comportamiento con tipo PROPERTY
- testSetType_Image: Verifica el comportamiento con tipo IMAGE

Estrategia de Testing:

1. Cobertura Completa: Se cubren casos normales, casos límite y casos de error
2. Pruebas Unitarias: Cada método se prueba de forma aislada
3. Pruebas de Integración: Se verifican flujos completos de trabajo
4. Manejo de Excepciones: Se valida que se lancen las excepciones correctas en los escenarios apropiados

Técnicas Utilizadas:

- @BeforeEach: Para inicializar un objeto File fresco antes de cada test
- assertThrows: Para verificar que se lancen las excepciones esperadas
- assertEquals/assertTrue/assertNotNull: Para verificaciones de estado
- assertDoesNotThrow: Para verificar operaciones que deben ejecutarse sin errores

Casos de Prueba Importantes:

1. Validación de null: Asegura que no se permitan valores null
2. Validación de tipo: Asegura que solo archivos PROPERTY acepten propiedades
3. Conversión de bytes: Verifica el correcto manejo del byte menos significativo
4. Contenido vacío: Maneja correctamente el caso especial de CRC32 con contenido vacío

Cobertura de Código:

Las pruebas cubren:

- Constructor: 100%
- addProperty(): 100% (incluyendo todas las ramas de validación)
- getCRC32(): 100% (incluyendo caso especial de contenido vacío)
- Manejo de excepciones: 100%

Archivos Creados:

- `FileTest.java` - Suite completa de pruebas unitarias con 18 tests