

INFORME DE PRUEBAS

A. Perez Pedraza, 160004829 - J. Andrés Ariza, 160004904 - C. Andres Cordoba, 160004914 - Y. Fabio Rey, 160004928 - N. Antonio Royert, 160004933 - O. Ignacio Galindo, 160005113 Universidad de los Llanos, Villavicencio, Colombia

Introducción.

El presente informe tiene por objetivo validar, mediante pruebas funcionales y no funcionales, si el sistema satisface las necesidades del usuario expresadas en requisitos, historias de usuario y casos de uso implementados en el producto. Además, documenta los defectos y observaciones identificados y evalúa si el sistema cumple los criterios de aceptación para su entrega.

Evaluación de requisitos.

En esta sección se presentan los análisis de los requerimientos funcionales y no funcionales, junto con sus respectivas historias de usuario, con el fin de verificar su cumplimiento dentro del producto final.

Requerimientos Funcionales

Código: HU-01: Iniciar sesión y acceder según rol

Descripción: El sistema debe permitir al usuario autenticarse con sus credenciales y mostrar únicamente las funciones correspondientes a su rol.

Reporte: El inicio de sesión funciona correctamente con credenciales válidas y el sistema restringe las funcionalidades según el rol. Sin embargo, la opción de recuperación de contraseña presenta fallas y no se realiza como debe ser.

Código: HU-02: Crear y actualizar lotes

Descripción: La aplicación debe permitir al operario crear y actualizar lotes, registrando especie, fecha de inicio, cantidad de ejemplares y estado, con el fin de mantener un control de la producción.

Reporte: La aplicación permite crear y actualizar lotes y obliga al completado de los campos requeridos. Si falta información obligatoria, el sistema muestra un error y no permite finalizar la creación del lote..

Código: HU-03: Registrar muestreo de un lote

Descripción: La aplicación debe permitir al inspector registrar muestreos indicando fecha, tamaño de muestra y método, para iniciar la evaluación de la calidad del lote.

Reporte: La aplicación si permite registrar muestreos manualmente, además de que también se registran automáticamente al analizar un pez usando la IA, no obstante, no

implementa herramientas para indicar el tamaño de la muestra o un método para iniciar la evaluación de calidad del lote.

Código: HU-04: Adjuntar fotografías al muestreo

Descripción: La aplicación debe permitir al inspector adjuntar una o varias fotografías al muestreo, junto con metadatos (parte, marca de tiempo, etc.), para evidenciar la evaluación.

Reporte: La aplicación permite adjuntar fotografías y clasificarlas por parte (por ejemplo: ojo, branquias, piel). Además, detecta imágenes borrosas o no relacionadas con peces y muestra una advertencia. Las imágenes se almacenan con metadatos asociados.

Código: HU-05: Obtener resultado IA por fotografía

Descripción: La aplicación debe permitir al usuario subir una fotografía que sea clasificada automáticamente por la IA, entregando puntajes por criterio, categoría y nivel de confianza, para reducir tiempo y subjetividad.

Reporte: La aplicación envía la imagen a la API de IA, recibe la clasificación y muestra el resultado con puntajes y nivel de confianza. Se implementan reintentos en caso de fallo o demora en la respuesta.

Código: HU-06: Generar evaluación de calidad del muestreo

Descripción: La aplicación debe permitir al inspector generar una evaluación de calidad aplicando las reglas activas (p. ej., NTC 1443) para emitir una decisión sobre el lote (apto / reproceso / descarte).

Reporte: Mediante la integración con la IA y la aplicación de la normativa NTC 1443, la aplicación calcula un ponderado global de calidad de carne utilizando ponderaciones fijas definidas por el equipo del proyecto. La versión actual no contempla edición manual de ponderaciones desde la interfaz; el recálculo de la evaluación se realiza siempre aplicando la misma versión de reglas almacenada para garantizar la trazabilidad de la decisión.

Código: HU-07: Generar reportes

Descripción: La aplicación debe permitir al administrador generar reportes por fecha, especie y lote, que incluyan la distribución de clases y enlaces a las imágenes para auditoría y clientes.

Reporte: La aplicación genera reportes automáticamente ordenados por fecha, especie, lote, además de contar con enlaces a las imágenes.

Código: HU-08: Ver trazas de registro y supervisión

Descripción: La aplicación debe permitir al administrador consultar quién registró cada muestreo y quién supervisó la evaluación, con fecha y hora, para garantizar trazabilidad.

Reporte: Se visualizan la fecha y la hora de cada evaluación junto con el estanque y la estación correspondientes. Sin embargo, la información sobre el usuario que ejecutó o supervisó la evaluación (identificador o rol) no se muestra actualmente.

Código: HU-09: Gestión de estaciones y tanques

Descripción: El sistema debe permitir al administrador y al operario gestionar el registro de estaciones piscícolas y sus tanques asociados, de modo que la estructura del sistema refleje la organización real de la producción y facilite la gestión de lotes.

Reporte: El sistema permite al administrador y al operario registrar y administrar estaciones y tanques, incluyendo nombre, ubicación, capacidad y tipo de pez.

REQUERIMIENTOS NO FUNCIONALES

RNF01: Rendimiento

Descripción: El sistema deberá clasificar una imagen mediante el servicio de IA en < 2 s en el hardware objetivo. Además, la generación de reportes debe completarse en ≤ 5 s para conjuntos de hasta 100 registros

Análisis: Durante las pruebas observadas, los tiempos de clasificación y de generación de reportes se ajustan a los valores solicitados en la especificación. Las mediciones registradas muestran respuestas rápidas del servicio de clasificación y tiempos de exportación de reportes aceptables según la condición de ≤ 5 s para 100 registros.

RNF02: Disponibilidad

Descripción: El sistema debe funcionar el 99% del horario laboral

Análisis: Durante el periodo de pruebas no se registraron caídas ni interrupciones que afectaran las tareas de validación, lo que sugiere alta disponibilidad en el entorno evaluado..

RNF03: Seguridad

Descripción: El sistema debe estar constituido por credenciales cifradas; control de acceso por rol; registro de auditoría (quién/qué/cuándo).

Análisis: Se comprobó la existencia de autenticación adecuada, control de acceso basado en roles y registros de auditoría que documentan acciones relevantes dentro del sistema. Estos mecanismos permiten garantizar la integridad y la trazabilidad de la información.

RNF04: Protección de datos

Descripción: El sistema debe permitir almacenamiento de imágenes con control de acceso y eliminación segura (retención configurable).

Análisis: El sistema restringe correctamente el acceso a los datos asociados a estaciones, tanques e imágenes, evitando modificaciones o eliminaciones no autorizadas. El control de acceso funciona conforme a lo establecido en el requerimiento.

RNF05: Escalabilidad

Descripción: El sistema debe soportar más de 10 mil imágenes almacenadas sin degradación perceptible en consulta.

Análisis: Aunque no se alcanzó la cifra exacta de 10 000 imágenes, se realizaron pruebas con un volumen considerable de datos y no se evidenció degradación en los tiempos de consulta. El sistema mostró un comportamiento estable ante cargas elevadas..

RNF06: **Interoperabilidad** **API** **IA**

Descripción: El sistema debe permitir la comunicación entre el frontend/backend desarrollado en Java y el módulo de clasificación IA en Python a través de una API REST documentada y segura.

Análisis: Se verificó comunicación correcta entre los componentes Java y el módulo de IA en Python. La API procesó las solicitudes adecuadamente, devolvió resultados en el formato esperado y aceptó únicamente imágenes en formatos permitidos (jpg, png, jpeg).

ANALISIS DE DIAGRAMAS UML

Diagrama de Casos de Uso

El diagrama de casos de uso define los requisitos funcionales mostrando cómo los actores (usuarios y sistemas externos) interactúan con el sistema. Por ejemplo, el diseño propone casos de uso como “Iniciar sesión”, “Crear/editar lote” y “Registrar muestreo”. En el código se observa correspondencia con estos casos: existen módulos/clases asociados a la autenticación de usuarios, al manejo de *lotes* y *muestreos*, etc., lo que sugiere que se implementaron las funcionalidades previstas. Esto cumple con la idea de que el diagrama de casos de uso define lo que el sistema **debe hacer** y guía el desarrollo de esas funciones. En resumen, la implementación refleja el diseño propuesto en el diagrama de casos de uso.

Diagrama de Clases

El diagrama de clases muestra la estructura estática del sistema: clases, atributos, métodos y relaciones. Es el “plano” que define qué clases deben existir en la implementación. Por ejemplo, el diagrama incluye clases como *Lote*, *Tanque*, *Usuario* y *Muestreo*, significa que en el código deberían aparecer clases o componentes equivalentes con esos nombres y responsabilidades. Efectivamente, el repositorio está organizado en paquetes y clases que reflejan estos conceptos de negocio, lo que indica que se respetó el diseño. Los diagramas de clases describen lo que debe estar presente en el sistema y sus elementos suelen corresponder directamente con las clases programadas. La correspondencia entre el modelo UML y la implementación confirma que la estructura de código implementa el diseño previsto.

Diagrama de Secuencia

El diagrama de secuencia modela la interacción dinámica en el tiempo entre objetos para realizar un caso de uso. Describe “cómo y en qué orden” los objetos colaboran durante un

proceso. Por ejemplo, un diagrama de secuencia de “Crear lote” mostraría mensajes desde la interfaz de usuario hacia el *LoteController*, luego hacia el *LoteService* y al repositorio de datos, en ese orden. Al revisar el código, se confirma la implementación de dicha secuencia: existen componentes (controladores, servicios, repositorios) que encadenan llamadas en el orden previsto. El cumplimiento del diseño se refleja si, por ejemplo, el controlador llama al servicio y éste al repositorio en la misma secuencia ordenada por el diagrama.

Validación de Requerimientos

La **Tabla I** presenta la matriz de validación de requerimientos, en la cual se resume el nivel de cumplimiento observado para cada historia de usuario y requisito no funcional, indicando si fue satisfecho total o parcialmente por el sistema.

Tabla I. Matriz de Validación de Requerimientos

Historia / Requisito	Estado	Observaciones (resumen)
HU-01 — Iniciar sesión y roles	Cumple	Inicio de sesión funciona; recuperación de contraseña envía token de restablecimiento.
HU-02 — Crear/actualizar lotes	Cumple	Permite crear/editar lotes; se sugiere usar lista desplegable para ubicación en vez de entrada libre.
HU-03 — Registrar muestreo	Parcial	Registro manual y automático vía IA funcionan; falta opción para especificar tamaño de muestra.
HU-04 — Adjuntar fotografías	Cumple	Subida y clasificación por parte (ojos/branchias/piel) y detección de imágenes borrosas funcionan.

HU-05 — Resultado IA por foto	Parcial	Clasificación por IA funcional; actualmente cubre pocas especies (p.ej. tilapia).
HU-06 — Generar evaluación de calidad	Cumple	Evaluación con NTC 1443 implementada; se calcula puntaje global y decisión aplicando ponderaciones fijas definidas por el proyecto y se registra la versión de reglas utilizada. La edición manual de ponderaciones no hace parte del alcance actual del sistema.
HU-07 — Generar reportes	Cumple	Reportes por fecha/especie/lote con enlaces a imágenes generados correctamente.
HU-08 — Trazabilidad / Logs	Parcial	Fecha/hora, estanque y estación se registran; no se muestra el usuario/rol que ejecutó/supervisó.
HU-09 — Gestión estaciones y tanques	Cumple	Registro y gestión de estaciones/tanques (nombre, ubicación, capacidad, tipo de pez) presentes.
RNF01 — Rendimiento (clasificación / reportes)	Cumple	Respuestas de clasificación rápidas; generación de reportes aceptable en entorno de prueba.
RNF02 — Disponibilidad	Parcial	No se observaron caídas en pruebas; falta prueba de concurrencia (múltiples usuarios simultáneos).
RNF03 — Seguridad	Cumple	Autenticación y control por roles presentes; mecanismos de auditoría activos.

RNF04 — Protección de datos	Cumple	Control de acceso a imágenes y datos funciona según lo esperado.
RNF05 — Escalabilidad ($\geq 10\ 000$ imágenes)	Parcial	No se probó con 10 000 imágenes; con volúmenes inferiores no hubo degradación perceptible.
RNF06 — Interoperabilidad (API IA)	Cumple	Comunicación Java ↔ Python vía API REST funciona y acepta formatos de imagen permitidos.

Conclusión

El sistema cumple con la mayoría de los objetivos funcionales planteados en la especificación: permite a los usuarios gestionar lotes y estaciones, registrar muestreos, adjuntar fotografías con metadatos, obtener clasificaciones automáticas vía IA y generar reportes consolidados con enlaces a la evidencia. La integración entre la interfaz, el backend y el servicio de IA se comporta de forma coherente con el diseño propuesto, lo que garantiza que los flujos críticos (subida de imagen → clasificación → evaluación → reporte) estén disponibles y operativos en el entorno probado.

No obstante, se identificaron áreas que requieren atención antes de un despliegue en producción: la trazabilidad no siempre muestra el usuario/rol responsable de acciones clave; la evaluación automática carece de una opción de ajuste manual de ponderaciones; la cobertura del modelo de IA es limitada a determinadas especies; y no se ejecutaron pruebas de volumen y concurrencia hasta los umbrales especificados, por lo que la escalabilidad y la disponibilidad en condiciones de carga real aún no están validadas.