

**SISTEMA DE CONTROL DE CALIDAD EN EL PROCESAMIENTO DE GRANOS**

**Documentación de Proyecto**

Presentado por

**Juan Pablo Mozuca Chaparro**

**Sebastián Camilo Murcia Mateus**

**Nicolas Santiago Ruiz Suarez**

**SERVICIO NACIONAL DE APRENDIZAJE - CENTRO DE BIOTECNOLOGIA AGROPECUARIA**

**TECNOLOGO DE ANALISIS Y DESARROLLO DE SOFTWARE**

Mosquera, Cundinamarca, Colombia

(agosto 2025)

**SISTEMA DE CONTROL DE CALIDAD EN EL PROCESAMIENTO DE GRANOS**

**Juan Pablo Mozuca Chaparro**

**Sebastián Camilo Murcia Mateus**

**Nicolas Santiago Ruiz Suarez**

APROBADO:

**Iván Malaver Fierro**

Mosquera, Cundinamarca 25 de agosto de 2025

# Tabla de Contenido

[Tabla de Contenido iii](#_Toc207554664)

[1.ARQUITECTURA Y LOGICA DEL PROCESO iv](#_Toc207554665)

[1.1 Lógica del Proceso: Flujo Operacional y Componentes Clave v](#_Toc207554666)

[**1.1.1.** **Interfaz de Usuario** v](#_Toc207554667)

[**1.1.2.** **Módulo de Registro de Datos de Control** v](#_Toc207554668)

[**1.1.3.** **Algoritmos de Verificación y Procesamiento** v](#_Toc207554669)

[**1.1.4.** **Generación de Informes y Conectividad Externa** vi](#_Toc207554670)

[1.2. Arquitectura del Sistema: Diseño Monolítico vi](#_Toc207554671)

[**1.2.1.** **Justificación Técnica y Operacional** vi](#_Toc207554672)

[**1.2.2 Beneficios Tangibles del Enfoque Arquitectónico** vii](#_Toc207554673)

[2. Diseño de la Base de Datos Relacional viii](#_Toc207554674)

[2.1 Entidades Principales del Modelo de Datos viii](#_Toc207554675)

[2.2. Relaciones entre Entidades ix](#_Toc207554676)

[2.3. Funcionalidades y Soporte a la Aplicación x](#_Toc207554677)

[3. Requerimientos Funcionales y Metodología de Desarrollo xi](#_Toc207554678)

[3.1 Historias de Usuario: Perspectiva del Cliente xi](#_Toc207554679)

[3.2. Planificación de Tareas (Sprint xii](#_Toc207554680)

[3.3. Caso de Uso: Registrar Inspección xiii](#_Toc207554681)

[3.4. Alineación Estratégica xiv](#_Toc207554682)

[4. Estrategia de Diseño y Plan de Desarrollo xiv](#_Toc207554683)

[4.1 Diseño Arquitectónico: Enfoque Híbrido C4 y UML xiv](#_Toc207554684)

[**4.1.1 Modelado con C4** xiv](#_Toc207554685)

[**4.1.2. Detalle con UML** xv](#_Toc207554686)

[4.2 Plan de Desarrollo: Backlog y Gestión por Sprints xvi](#_Toc207554687)

[**4.2.1. Backlog Inicial: Épicas y Criterios** xvi](#_Toc207554688)

[**4.2.2. Plan de Trabajo: 3 Sprints de una Semana** xvii](#_Toc207554689)

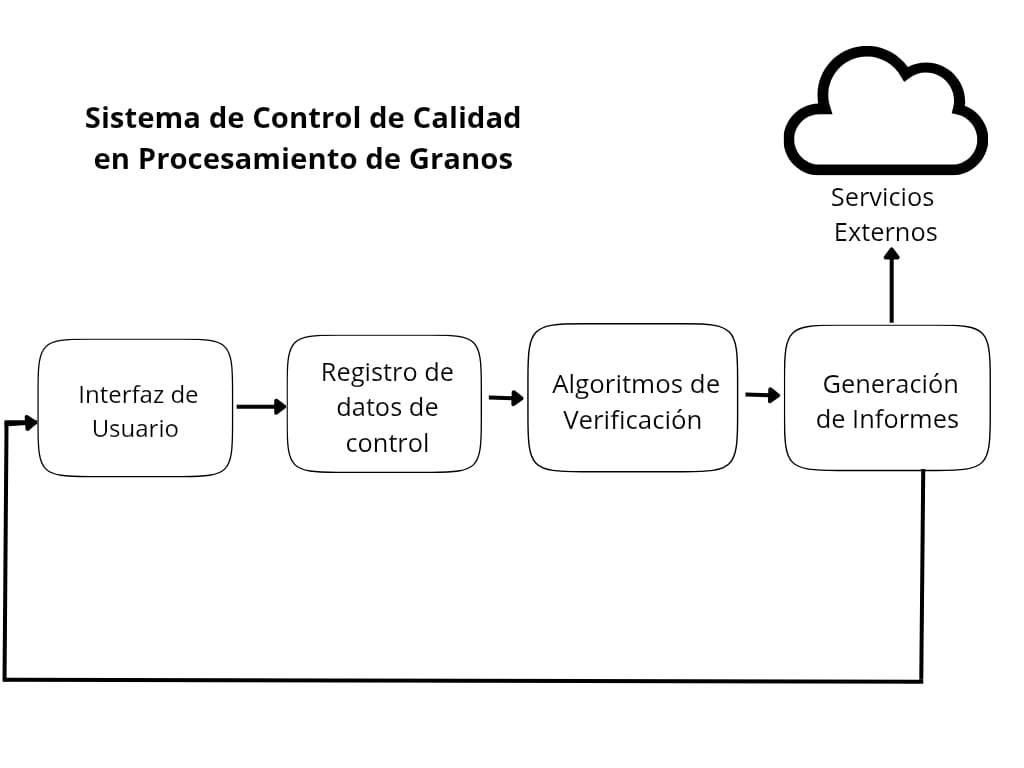
[4.3. Riesgos y Estrategias de Mitigación xvii](#_Toc207554690)

[5. Control de Acceso y Casos de Uso del Sistema xviii](#_Toc207554691)

[5.1 Roles y Responsabilidades Fundamentales xix](#_Toc207554692)

[5.2 Casos de Uso del Sistema xx](#_Toc207554693)

# **ARQUITECTURA Y LOGICA DEL PROCESO**



El presente documento detalla la estructura lógica y la arquitectura tecnológica del sistema de clasificación de granos, diseñado con el objetivo fundamental de establecer un flujo de trabajo claro, seguro y eficiente. La metodología de diseño se basa en la integración de componentes modulares que, en conjunto, garantizan la fiabilidad de los procesos de control de calidad, desde la adquisición inicial de datos hasta la generación de informes finales para la toma de decisiones.

## **Lógica del Proceso: Flujo Operacional y Componentes Clave**

La operatividad del sistema se articula a través de una secuencia de etapas interdependientes, cada una con un rol específico en la validación y clasificación de los granos. Este flujo secuencial asegura la integridad del proceso y el cumplimiento riguroso de los estándares de calidad predefinidos.

* + 1. **Interfaz de Usuario**

Este módulo constituye el punto de interacción principal entre el personal operativo, como analistas y operadores, y el sistema. Su diseño se ha enfocado en la usabilidad, permitiendo de manera intuitiva el ingreso de parámetros, la consulta de resultados y la configuración de los estándares de calidad. La interfaz actúa como el panel de control central para la gestión del proceso.

* + 1. **Módulo de Registro de Datos de Control**

Este componente desempeña una función crítica al centralizar y almacenar de forma segura todos los datos generados por los equipos de laboratorio. La consolidación de la información en una única base de datos facilita su trazabilidad, permite una validación posterior exhaustiva y asegura el cumplimiento de las normativas y requisitos de calidad establecidos para la industria.

* + 1. **Algoritmos de Verificación y Procesamiento**

En esta etapa, el sistema procesa la información recibida del módulo de registro. A través de la aplicación de algoritmos especializados, se cotejan los datos capturados con los estándares de calidad configurados. Este análisis automatizado garantiza que cada lote de granos cumpla con los parámetros de calidad especificados antes de avanzar a la etapa de generación de informes, minimizando el riesgo de errores humanos.

* + 1. **Generación de Informes y Conectividad Externa**

Una vez que los datos han sido validados, este módulo se encarga de la creación automática de informes detallados con los resultados del control de calidad. Estos reportes se convierten en herramientas esenciales para la gerencia y los departamentos de calidad, permitiendo una toma de decisiones informada y ágil. Adicionalmente, este componente está diseñado para interactuar con servicios externos (como plataformas de análisis químico o sistemas de almacenamiento en la nube), facilitando la distribución de los resultados a otras plataformas o stakeholders relevantes.

## **Arquitectura del Sistema: Diseño Monolítico**

La elección de una arquitectura monolítica responde a un análisis exhaustivo de los requerimientos técnicos y las restricciones operativas del proyecto. Este enfoque de diseño, donde todo el sistema reside en un único bloque de aplicación, se justifica por las siguientes consideraciones estratégicas:

* + 1. **Justificación Técnica y Operacional**

Restricción de Infraestructura: El despliegue del sistema en servidores locales, una restricción técnica impuesta por el entorno operativo, se optimiza con una arquitectura centralizada, que es inherente al diseño monolítico.

Simplicidad en el Despliegue y Mantenimiento: La estructura unificada reduce la complejidad técnica, lo que se traduce en un proceso de despliegue más ágil y en un mantenimiento simplificado, ya que todas las funcionalidades están contenidas en un solo repositorio de código.

Escalabilidad no Modular Crítica: A diferencia de los sistemas de microservicios, no existe un requerimiento crítico de escalabilidad diferenciada por módulo. La escalabilidad del sistema en su conjunto puede gestionarse de forma eficiente en la infraestructura local.

Integración de Hardware Específico: La integración directa con el hardware de laboratorio es más eficiente en un sistema monolítico, evitando las complejidades y los puntos de fallo que podría generar un esquema distribuido.

Control y Seguridad de Datos: Un sistema centralizado en un entorno industrial facilita la implementación de políticas de seguridad robustas y un control riguroso sobre los datos, un aspecto fundamental para la protección de la información sensible.

**1.2.2 Beneficios Tangibles del Enfoque Arquitectónico**

Mantenimiento Optimizado: La consolidación de todo el código en un único bloque simplifica las tareas de mantenimiento, depuración y actualización.

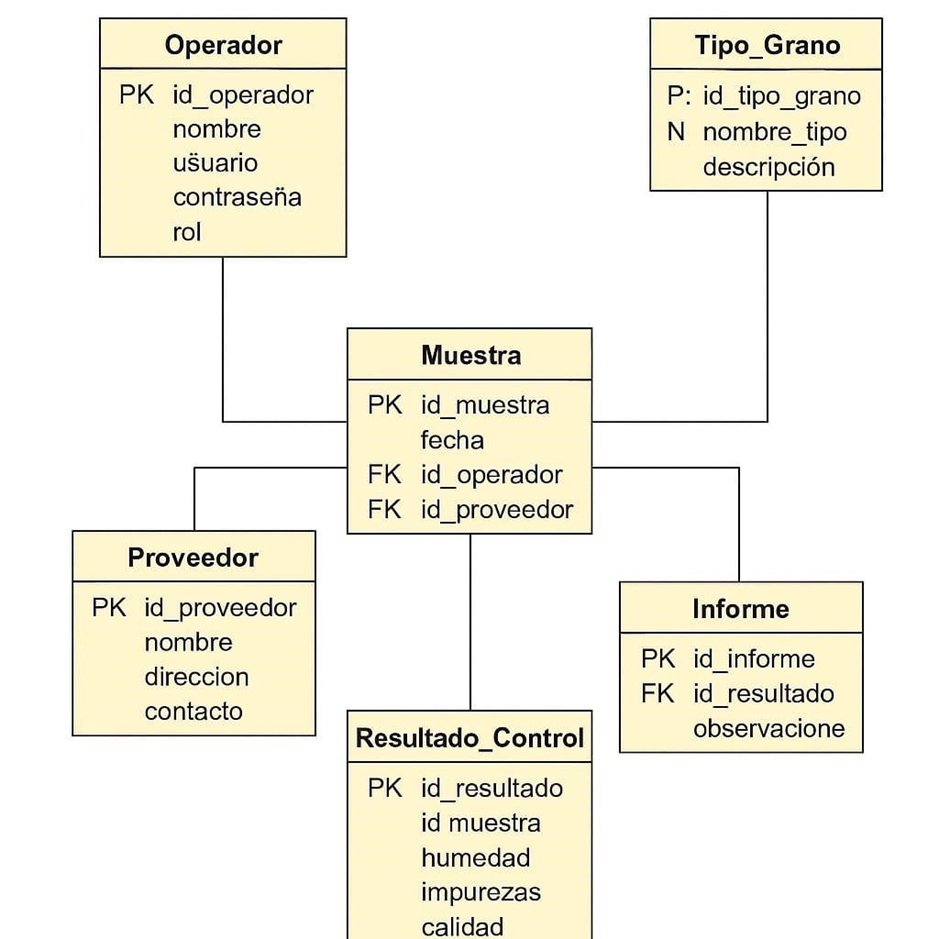
Reducción de la Complejidad: La simplicidad inherente a la arquitectura monolítica reduce los desafíos técnicos y la curva de aprendizaje para los desarrolladores.

Integración Directa y Eficiente: Facilita la comunicación directa entre el software y el hardware de laboratorio, lo que minimiza la latencia y maximiza el rendimiento del sistema.

# **Diseño de la Base de Datos Relacional**

El diseño de la base de datos se basa en un modelo relacional, estructurado para asegurar la integridad, la trazabilidad y la eficiencia en la gestión de la información. Este modelo permite una representación clara de las entidades principales y las relaciones que existen entre ellas, lo que es fundamental para el correcto funcionamiento del sistema de clasificación de granos.

## **2.1 Entidades Principales del Modelo de Datos**



Las entidades representan los objetos clave del sistema y sus atributos. Cada entidad está definida por una clave primaria (PK) que la identifica de forma única.

\* 1. Operador: Representa al usuario del sistema.

\* Campos: id\_operador (PK), nombre, usuario, contraseña, rol.

\* Motivo: Es crucial para la autenticación y para registrar quién realizó cada acción dentro del sistema, proporcionando un control de seguridad esencial.

\* 2. Muestra: La unidad fundamental de análisis.

\* Campos: id\_muestra (PK), fecha, id\_operador (FK), id\_proveedor (FK), id\_tipo\_grano (FK).

\* Motivo: Cada análisis de calidad se realiza sobre una muestra específica, y esta entidad centraliza la información de origen de dicha muestra.

\* 3. Proveedor: La fuente de los granos.

\* Campos: id\_proveedor (PK), nombre, dirección, contacto.

\* Motivo: Permite la trazabilidad del material recibido y el registro del historial de calidad por cada proveedor.

\* 4. Tipo Grano: Categorización del producto.

\* Campos: id\_tipo\_grano (PK), nombre\_tipo, descripción.

\* Motivo: Facilita la clasificación y el análisis de los diferentes tipos de granos, como maíz, trigo o arroz.

\* 5. Resultado Control: Almacena los datos del análisis de calidad.

\* Campos: id\_resultado (PK), id\_muestra (FK), humedad, impurezas, calidad, fecha\_análisis.

\* Motivo: Es el repositorio de los resultados cuantitativos y cualitativos de cada muestra, lo que permite la evaluación del producto.

\* 6. Informe: El documento final.

\* Campos: id\_informe (PK), id\_resultado (FK), fecha\_generación, observaciones.

\* Motivo: Contiene la versión final y consultable de los reportes, listo para su impresión o exportación.

## **2.2. Relaciones entre Entidades**

Las relaciones definen cómo las entidades se conectan entre sí, garantizando la integridad referencial y la coherencia del modelo. Las claves foráneas (FK) son los campos que enlazan una tabla con la clave primaria de otra.

\* Operador → Muestra: Relación uno a muchos. Un operador puede registrar múltiples muestras.

\* Proveedor → Muestra: Relación uno a muchos. Un proveedor puede suministrar varias muestras al sistema.

\* Tipo Grano → Muestra: Relación uno a muchos. Un tipo\_grano puede estar asociado a múltiples muestras.

\* Muestra → Resultado Control: Relación uno a uno. Cada muestra tiene un único resultado\_control de análisis.

\* Resultado Control → Informe: Relación uno a uno. Cada resultado\_control genera un solo informe.

## **2.3. Funcionalidades y Soporte a la Aplicación**

El diseño de esta base de datos es fundamental para las siguientes funcionalidades clave del sistema:

\* Trazabilidad Completa: Permite seguir el recorrido de cada muestra, desde su origen en el proveedor hasta la generación del informe final.

\* Historial de Calidad: Facilita el análisis histórico del rendimiento de los proveedores y de la calidad de los diferentes tipos de granos.

\* Seguridad y Control: El registro de operadores y sus acciones garantiza la seguridad de los datos y permite un control riguroso de las operaciones.

\* Consultas y Auditorías: El modelo relacional optimizado permite la ejecución de consultas rápidas para auditorías, la exportación de informes y el análisis de tendencias de forma eficiente.

# **Requerimientos Funcionales y Metodología de Desarrollo**

Los requerimientos del sistema se definen a través de historias de usuario y casos de uso, que detallan las funcionalidades clave desde la perspectiva del usuario final. Esta metodología de desarrollo garantiza que cada función del sistema responda directamente a una necesidad real y aporte valor al proceso de control de calidad.

## **3.1 Historias de Usuario: Perspectiva del Cliente**

Las historias de usuario representan, en un lenguaje sencillo, lo que los diferentes roles de usuario necesitan hacer con el sistema. Son la base para la planificación y el desarrollo de las funcionalidades.

\* Registro de Inspecciones: Como operador, necesito registrar los datos de inspección de un lote para poder evaluar su calidad.

\* Validación Automática de Parámetros: Como analista de calidad, busco que el sistema valide de forma automática los parámetros contra los estándares establecidos, lo que me permitirá detectar cualquier desviación con rapidez.

\* Generación de Reportes: Como analista, es fundamental poder generar reportes de calidad para cada lote, con el fin de documentar y comunicar los resultados de forma eficiente.

\* Consulta de Historial: Como operador, me interesa consultar inspecciones anteriores para poder comparar resultados y detectar tendencias en la calidad de los lotes.

\* Integración con Equipos de Laboratorio: Como operador, quiero que los datos de los equipos de laboratorio se registren automáticamente en el sistema. Esto me permitirá evitar errores manuales y ahorrar tiempo en el proceso.

**3.2. Planificación de Tareas (Sprint**)

Las funcionalidades se dividen en tareas concretas, organizadas en sprints de desarrollo para asegurar una entrega incremental y ordenada.

\* Para la Historia 1 (Registro de Inspecciones):

\* Crear el formulario de registro de inspecciones.

\* Validar los campos obligatorios del formulario.

\* Programar el guardado de la inspección en la base de datos.

\* Para la Historia 2 (Validación Automática):

\* Definir y configurar los estándares de calidad en la base de datos.

\* Programar el módulo de verificación de parámetros.

\* Implementar la funcionalidad de mostrar alertas cuando un valor está fuera de rango.

\* Para la Historia 3 (Generación de Reportes):

\* Diseñar el formato del reporte en PDF.

\* Programar la exportación de los reportes.

\* Implementar filtros de búsqueda por fecha y por lote.

## **3.3. Caso de Uso: Registrar Inspección**

Los casos de uso detallan la interacción entre el usuario y el sistema, describiendo los pasos necesarios para completar una tarea. El caso de uso “Registrar Inspección” es fundamental para la operación diaria del sistema.

\* Actores Involucrados:

\* Operador (Principal): Inicia y ejecuta la acción.

\* Sistema de Control de Calidad (Secundario): Procesa la información y realiza las validaciones.

\* Equipos de Laboratorio (Opcional): Fuente de datos automática.

\* Flujo Normal del Proceso:

\* El operador accede al sistema e inicia su sesión.

\* Selecciona la opción “Registrar inspección”.

\* Ingresa el código del lote y otros datos requeridos.

\* El sistema obtiene los parámetros del equipo de laboratorio de forma automática. Si la conexión falla, el operador ingresa los datos de forma manual.

\* El sistema valida que todos los campos obligatorios estén completos.

\* El operador guarda la inspección.

\* El sistema confirma que el registro ha sido exitoso.

\* Flujos Alternativos y Excepciones:

\* Flujo Alternativo: En caso de no poder conectar con el equipo de laboratorio, el operador ingresa todos los datos de forma manual.

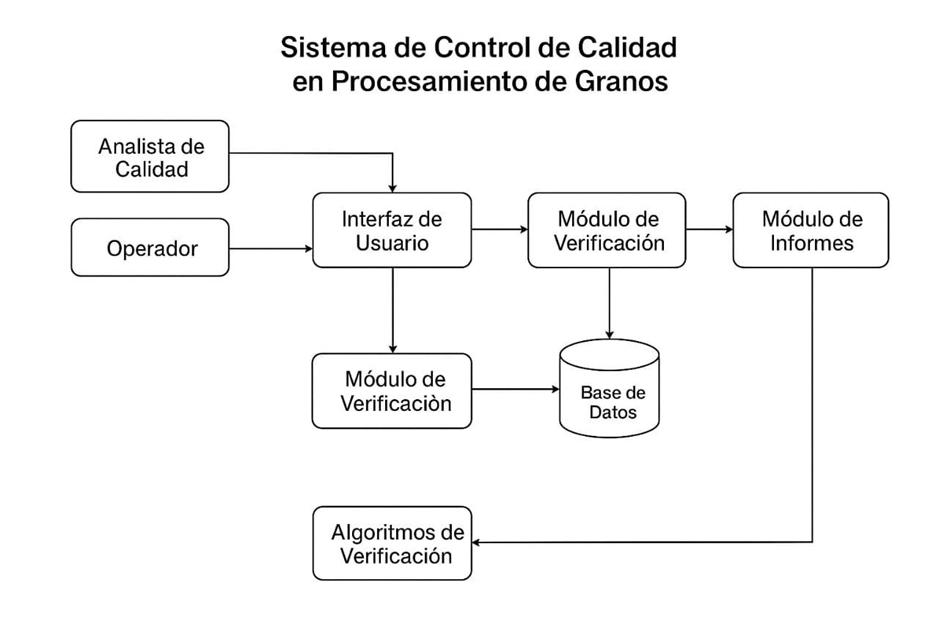
\* Excepción: Si el lote no está registrado previamente, el sistema solicita al operador que lo cree antes de continuar con la inspección.

## **3.4. Alineación Estratégica**

La definición de estos requerimientos a través de historias de usuario y casos de uso asegura que el desarrollo del software se alinee directamente con las necesidades de los roles clave (operador y analista). Este enfoque garantiza que las funcionalidades de registro, validación, generación de reportes, consulta de historial e integración con el laboratorio cubran todos los requerimientos funcionales clave, reflejando el flujo de trabajo real en la planta.

# **Estrategia de Diseño y Plan de Desarrollo**

El desarrollo del sistema de clasificación de granos se abordará con una estrategia robusta que integra un diseño arquitectónico exhaustivo y una metodología de desarrollo ágil. Este enfoque garantiza la alineación del proyecto con los objetivos de negocio, la gestión eficaz de los recursos y una mitigación proactiva de los riesgos.



## **4.1 Diseño Arquitectónico: Enfoque Híbrido C4 y UML**

Para el diseño del sistema, se ha tomado la decisión estratégica de utilizar un enfoque híbrido que combina la simplicidad del modelo C4 con la precisión técnica de UML. Este binomio permite una comunicación clara de la arquitectura en diferentes niveles de detalle, tanto para stakeholders no técnicos como para el equipo de desarrollo.

### **4.1.1 Modelado con C4**

El modelo C4 se utilizará para definir la estructura de alto nivel del sistema, enfocándose en dos niveles clave:

\* Nivel de Contenedor: Aquí se visualizarán las principales unidades implementables. Se han definido una Web UI para la interacción del usuario, un API Backend que servirá como un punto de entrada seguro y estable, y Servicios de Dominio especializados para la Ingesta de Datos, la Validación de Parámetros y la Generación de Reportes. Esta separación en contenedores facilita la comprensión de la arquitectura general y la distribución de responsabilidades.

\* Nivel de Componente: Dentro de cada contenedor, se modelarán los componentes internos y sus interacciones. Por ejemplo, en el servicio de validación, se definirá un componente para las reglas de negocio y otro para el motor de verificación. Este nivel permite una visión detallada de cómo se implementarán las funcionalidades clave.

### **4.1.2. Detalle con UML**

Para los detalles técnicos, se empleará UML, específicamente en los siguientes diagramas:

\* Diagrama de Secuencia: Este diagrama será crucial para modelar el flujo de los casos de uso críticos, como el registro de una inspección. Se mostrará la interacción temporal entre el Operador, la Web UI, el API Backend y los Servicios de Dominio. Este nivel de detalle asegura que las interacciones se planifiquen y ejecuten de manera eficiente, minimizando la latencia y los puntos de fallo.

\* Diagrama de Despliegue: Este diagrama visualiza la topología de la infraestructura física y lógica. Se prioriza un escenario on-premise para cumplir con la restricción de servidores locales. Se representará un App Server .NET que aloja la interfaz de usuario y el API, dos Servicios de Windows (o Workers) para procesos en segundo plano (ingesta, validación) y una base de datos SQL Server. Esta topología minimiza la latencia al mantener los componentes en la misma red local.

Este enfoque híbrido, al combinar la claridad del C4 con la precisión del UML, facilita el desarrollo de un sistema modular con acoplamiento débil. Esto significa que los componentes pueden evolucionar de forma independiente, lo que favorece las pruebas unitarias y la sustitución de adaptadores (ej. Para equipos de laboratorio) sin afectar la lógica de negocio central.

## **4.2 Plan de Desarrollo: Backlog y Gestión por Sprints**

El desarrollo del proyecto se gestionará a través de un backlog inicial, que divide el trabajo en épicas y historias de usuario con criterios de aceptación claros. Este enfoque ágil permite una entrega incremental y una adaptación flexible a los comentarios del cliente.

### **4.2.1. Backlog Inicial: Épicas y Criterios**

\* Épica E1: Registro y Verificación de Datos: Se enfoca en las funcionalidades de entrada de datos.

\* Historia H1.1 (Ingesta): Carga de datos desde equipos de laboratorio. Los criterios de aceptación incluyen el soporte para formatos CSV/JSON, la validación de esquemas y la persistencia de datos con trazabilidad.

\* Historia H1.2 (Validación): Motor de validación. Los criterios exigen la capacidad de un analista para configurar reglas, la validación automática y la entrega de resultados claros (Conforme/No Conforme).

\* Épica E2: Informes y Visualización: Cubre las funcionalidades de salida.

\* Historia H2.1 (Informes): Generación de reportes. Se debe generar un PDF con todos los parámetros, resultados y no conformidades.

\* Historia H2.2 (Consulta): Acceso al historial. Los criterios incluyen filtros por fecha, tipo de grano y un tiempo de respuesta medible (RNF: p95 ≤ 2s).

\* Épica E3: Plataforma y Seguridad: Aborda los requerimientos no funcionales (RNF).

\* Historia H3.1 (Autenticación): Control de acceso por roles (Analista/Operador) y registro de accesos en una bitácora.

\* Historia H3.2 (Escalabilidad y Rendimiento): El sistema debe ser capaz de manejar grandes volúmenes de datos. Los criterios incluyen pruebas de carga (JMeter ≥ 200 req/min) y el uso de índices y particiones en la base de datos para garantizar la eficiencia.

### **4.2.2. Plan de Trabajo: 3 Sprints de una Semana**

El desarrollo se ha planificado en tres sprints de una semana, cada uno con objetivos claros y entregables definidos.

\* Sprint 1 (Arquitectura y Diseño): El objetivo es sentar las bases. Se dedicarán los primeros días a investigar y tomar la decisión sobre el enfoque híbrido. Al final del sprint, se tendrán definidos los diagramas de contexto, contenedores y componentes, listos para la implementación.

\* Sprint 2 (Implementación del Núcleo): La meta es construir el esqueleto funcional del sistema. Se crearán los proyectos .NET para la API y los servicios en segundo plano. Se desarrollarán los módulos de ingesta y validación, y se modelarán las tablas en SQL Server. Los entregables incluirán el código base, scripts SQL y una colección de Postman para las pruebas iniciales.

\* Sprint 3 (Informes y Calidad): Se completarán las funcionalidades clave. Se creará el servicio de reportes en PDF, se implementarán los filtros de consulta, la autenticación por roles y, lo más importante, se realizarán pruebas de rendimiento con JMeter. Los entregables finales serán los reportes, los resultados de las pruebas y la documentación para el usuario y el administrador.

## **4.3. Riesgos y Estrategias de Mitigación**

Se ha realizado un análisis de riesgos para identificar posibles obstáculos y establecer planes de mitigación.

\* Riesgo: Integración con equipos de laboratorio propietarios: Estos equipos pueden tener protocolos de comunicación complejos.

\* Mitigación: Se ha definido un plan B que permite la importación de datos a través de archivos estandarizados (CSV), asegurando la continuidad del flujo de trabajo incluso si la conexión directa falla.

\* Riesgo: Rendimiento con grandes lotes: La base de datos podría ralentizarse con un gran volumen de registros.

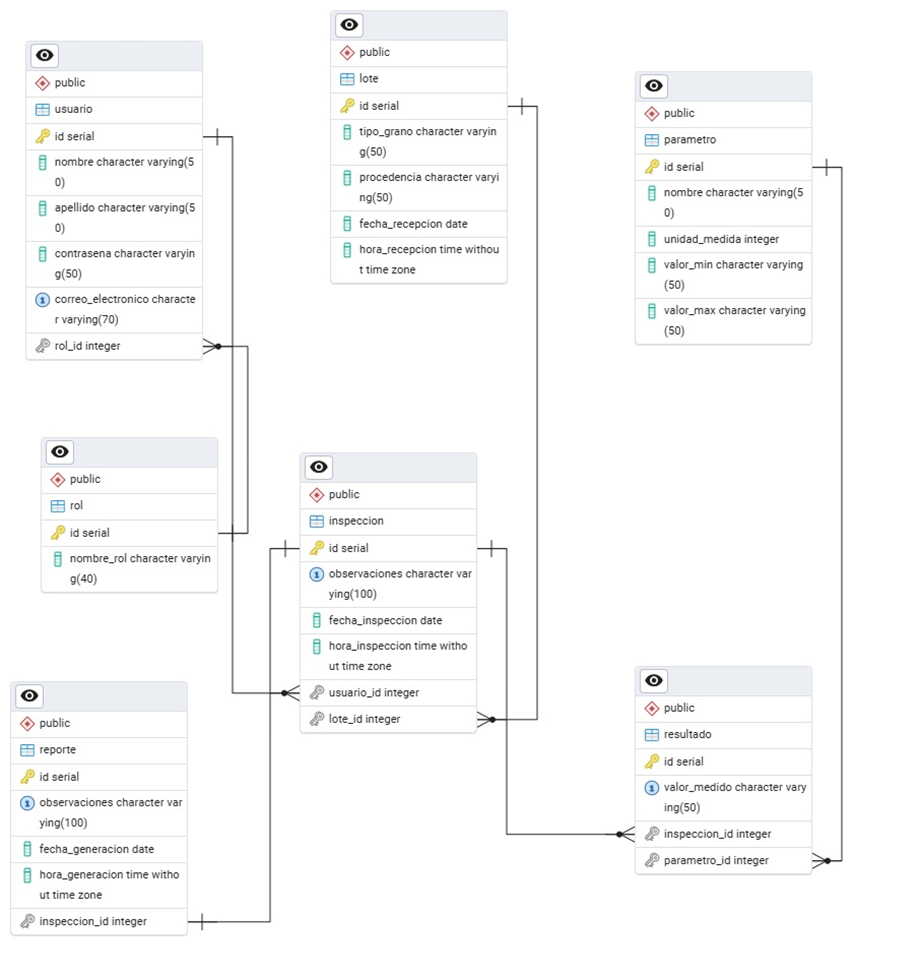
\* Mitigación: Desde el diseño, se ha planificado el uso de índices compuestos y particiones por fecha/lote en SQL Server. Además, las pruebas de rendimiento en el Sprint 3 permitirán ajustar las consultas y la infraestructura de manera proactiva.

\* Riesgo: Cambios en las reglas de calidad: Los estándares de la industria pueden evolucionar.

\* Mitigación: La arquitectura se ha diseñado con la capacidad de versionar las reglas de validación en la base de datos y controlar las migraciones, lo que permite adaptaciones sin afectar el historial de resultados.

Este plan de desarrollo, meticulosamente estructurado, no solo define qué construir, sino también cómo y por qué, garantizando un proyecto robusto, escalable y alineado con los objetivos del negocio.

# **Control de Acceso y Casos de Uso del Sistema**



La seguridad y la eficiencia de un sistema son tan sólidas como su diseño de control de acceso. En nuestro caso, hemos implementado un modelo de control de acceso basado en roles (RBAC), una estrategia de seguridad que asigna permisos de usuario según su función dentro de la organización. Esto no solo simplifica la administración, sino que también garantiza que cada usuario solo pueda realizar las acciones necesarias para su trabajo, minimizando los riesgos de acceso no autorizado y errores.

## **5.1 Roles y Responsabilidades Fundamentales**

El diseño del sistema de clasificación de granos se basa en una jerarquía clara de roles, cada uno con responsabilidades específicas que reflejan el flujo de trabajo real en una planta de procesamiento.

\* Operador: Este rol es la columna vertebral de la operación diaria. El Operador tiene la tarea principal de interactuar directamente con el material físico y los equipos para la captura de datos. Sus responsabilidades están estrictamente definidas:

\* Ingresar información de lotes: Registran los detalles iniciales de cada lote de granos que llega, lo que es el punto de partida para la trazabilidad.

\* Realizar inspecciones: Llevan a cabo las inspecciones físicas y registran los datos de calidad (como humedad e impurezas) que el sistema necesita para su posterior análisis.

\* Generar reportes básicos: Pueden producir informes preliminares para su propio uso o para compartir con el Analista, lo que agiliza la comunicación en la línea de producción.

Este rol es crucial para alimentar el sistema con información precisa desde la primera etapa del proceso.

\* Analista: A diferencia del Operador, el Analista tiene una función más estratégica y técnica. Su trabajo se centra en la gestión de la calidad y el análisis de los datos. Sus responsabilidades incluyen:

\* Configurar estándares de calidad: Son los únicos autorizados para definir los parámetros y umbrales que el sistema utilizará para evaluar la calidad de los granos. Esto asegura que el análisis se base en las normativas correctas.

\* Verificar y validar los datos: Analizan los resultados de las inspecciones realizadas por el Operador, asegurando la coherencia y la precisión.

\* Registrar conclusiones finales: Con base en su análisis, solo ellos pueden registrar las conclusiones definitivas, que son esenciales para la toma de decisiones de negocio.

Su rol es vital para convertir los datos brutos en inteligencia de negocio utilizable.

\* Súper Usuario: Este rol es el administrador supremo del sistema. Su función no es operativa, sino de gestión y control. Sus responsabilidades son las de mayor nivel de privilegio:

\* Gestión de roles y usuarios: Tienen la capacidad de crear, modificar y eliminar los roles de Operador y Analista, lo que garantiza que el sistema siempre esté alineado con la estructura organizacional.

\* Inserción de parámetros críticos: Pueden ingresar y modificar parámetros vitales del sistema, como las bases de datos externas o las configuraciones de seguridad.

\* Supervisión del sistema: Tienen acceso completo para auditar el uso del sistema y asegurar que la base de datos se mantenga íntegra y sin errores.

Este rol es la garantía de la estabilidad y escalabilidad del sistema a largo plazo.

## **5.2 Casos de Uso del Sistema**

Los casos de uso son acciones concretas que un usuario realiza para alcanzar un objetivo. Para nuestro sistema, se han definido a través del formato conciso: “Como (rol) quiero (función) para que (resultado)”. Estos casos de uso no son solo una lista de tareas, sino los pilares que impulsan el flujo de trabajo de la base de datos y aseguran la fluidez del proceso de control de calidad.

\* Casos de Uso del Operador: Estos son fundamentales para la primera etapa del proceso. Sin las acciones del Operador, el sistema no podría obtener los datos necesarios.

\* Añadir granos del lote: Esta acción es el punto de entrada que alimenta el sistema. Es vital para mantener un inventario actualizado y para que cada lote tenga un registro inicial que pueda ser rastreado.

\* Realizar inspecciones: Al ejecutar esta función, el Operador activa el proceso de evaluación de la calidad. Sus datos son la materia prima que el Analista utilizará para determinar si los granos cumplen con los estándares y son viables para su uso posterior.

\* Casos de Uso del Analista: Las acciones del Analista son cruciales para la excelencia del proceso.

\* Añadir estándares: Esta es una de las funcionalidades más importantes del sistema. La capacidad de insertar, modificar o eliminar parámetros y estándares es lo que permite que la evaluación de calidad sea precisa y relevante.

\* Recibir resultados: Esta función le permite no solo verificar la calidad de los granos, sino también supervisar que el Operador haya usado el sistema correctamente. Esto asegura que los datos sean confiables y que el proceso se mantenga optimizado.

\* Caso de Uso del Súper Usuario: Este rol es crítico para la estabilidad y escalabilidad del sistema.

\* Insertar roles: Esta función es la base de todo el control de acceso. Si el Súper Usuario no crea los roles de Operador y Analista, el sistema no puede funcionar. Esta capacidad también es vital para adaptarse a las necesidades futuras, como la adición de nuevos roles. Su supervisión general garantiza la integridad de la base de datos y el correcto uso del sistema.

En conjunto, estos roles y casos de uso forman un ecosistema donde cada usuario tiene una función clara y complementaria. Esta estructura asegura que la información fluya de manera lógica, desde la recolección inicial hasta el análisis final, lo que es esencial para el éxito del proyecto a largo plazo.