Seminario de Práctica de Informática

Trabajo Práctico N°4

Araoz, Francisco Javier

2° Cuatrimestre

Año 2025

Índice

Título del proyecto

Sistema de Gestión de Stock y Reparaciones para el Centro de Informática del PJN

Introducción

El Centro de Informática del PJN se ocupa de la administración y seguimiento de los equipos tecnológicos que utilizan las distintas dependencias. Actualmente, gran parte de esta tarea se realiza mediante planillas de Excel elaboradas de forma manual y con criterios poco estandarizados. Esto genera dispersión de datos, dificultades para localizar información y una alta dependencia del conocimiento personal de los técnicos.

El presente proyecto propone el desarrollo de un sistema informático que permita centralizar la gestión de los equipos, facilitando la consulta, el registro de movimientos y la generación de reportes de manera más eficiente.

Justificación

La gestión de equipos de cómputo es un proceso esencial para garantizar la continuidad de los servicios de la organización. El manejo actual con planillas distribuidas implica problemas de actualización, duplicación de información y falta de trazabilidad en las reparaciones. Estas limitaciones producen retrasos en la atención de incidentes, reducen la confiabilidad de los datos y dificultan la planificación de recursos.

La implementación de un sistema propio permitirá resolver estas deficiencias al concentrar la información en una única herramienta, mejorando la calidad y la disponibilidad de los datos. Esto impactará positivamente en la eficiencia del área de soporte, en la transparencia de los procesos y en la capacidad de respuesta ante las necesidades de las dependencias.

Este problema se agravó en los últimos años, sobre todo después de la pandemia. En ese momento la institución aceleró su proceso de informatización, pero de manera bastante desordenada. Algunas áreas modernizaron sus registros, mientras que otras siguieron con métodos viejos. Eso dejó un sistema poco integrado y con cuellos de botella que hacen más difícil tener información confiable y rápida.

Si no se hace nada, la situación va a seguir igual: registros incompletos, tiempos muertos y dificultades para responder rápido a las dependencias que nos piden datos. En cambio, si se implementa el sistema, se lograría mejorar la calidad de la información, agilizar los procesos diarios y facilitar la toma de decisiones.

Objetivo general

Desarrollar un sistema de gestión integral que centralice el inventario de equipos informáticos y el historial de reparaciones, con el fin de optimizar la administración de recursos tecnológicos y garantizar la disponibilidad de información confiable para la toma de decisiones.

Objetivos específicos

- Implementar un registro único y estandarizado de equipos informáticos vinculados a cada dependencia.
- Mantener actualizado el historial de altas, bajas y modificaciones de equipos.
- Registrar y dar seguimiento a las solicitudes y estados de reparación.
- Generar reportes de stock y reparaciones con exportación en formatos estándar.
- Garantizar seguridad de acceso mediante perfiles de usuario y registro de auditoría.
- Facilitar la consulta ágil de información por dependencia, tipo de equipo o número de inventario.

Objetivos del sistema

El sistema permitirá registrar, administrar y consultar equipos informáticos; gestionar solicitudes de reparación; llevar historial de intervenciones; generar reportes exportables y administrar perfiles de usuarios con diferentes permisos.

Límites del proyecto

- **Desde:** relevamiento de los procesos actuales de gestión de inventario y reparaciones en el área de soporte, diseño e implementación de la aplicación en entorno de intranet, carga inicial de datos y capacitación de los usuarios directos.
- **Hasta:** puesta en marcha del sistema, uso en producción para administración de equipos informáticos y reparaciones, incluyendo la generación de reportes operativos y de gestión.

Definición del proyecto y sistema

Proyecto

El presente proyecto consiste en el desarrollo de un sistema para la gestión integral de stock y reparaciones de los equipos informáticos que se encuentran en el centro de informática. En esta primera etapa se enfocó en computadoras e impresoras, aunque el diseño previó la posibilidad de ampliar su alcance en el futuro e incorporar otros dispositivos sin mayores cambios en la estructura.

Sistema

El sistema tendrá una doble función. Por un lado, permitirá registrar y mantener actualizado el stock de equipos informáticos, incluyendo datos básicos como número de serie, sector al que pertenecen y su estado actual. Por otro lado, incorporará un módulo específico para la gestión de reparaciones, donde se podrá llevar un control detallado de las intervenciones realizadas, los equipos en proceso de reparación y los ya finalizados.

Ambos módulos estarán integrados, lo que constituye la principal innovación frente al método previo de trabajo. De esta manera, será posible consultar la información completa de cada equipo en un solo lugar, evitando duplicaciones de datos y reduciendo la probabilidad de errores. Además, el sistema permitirá generar reportes de stock y de reparaciones que facilitarán el análisis y la toma de decisiones en el área.
En definitiva, el objetivo central del sistema es reemplazar procedimientos manuales y dispersos por una solución más organizada, eficiente y escalable, que sirva no solo para mejorar la gestión actual, sino también como base para futuras mejoras y expansiones.

Elicitación

El proceso de elicitación tuvo como objetivo identificar las necesidades y limitaciones en el Centro de Informática. La intención fue relevar los principales problemas que enfrentan los técnicos del sistema y las expectativas respecto a una solución que aporte eficiencia y reduzca la recurrencia de incidencias.

El alcance del relevamiento incluyó los procesos operativos más frecuentes del área, principalmente la atención de consultas, resolución de incidentes básicos y gestión de solicitudes. Quedaron fuera del análisis los procesos menos habituales o los que dependen de otros sectores ajenos a la operatoria central.

Para llevar adelante el relevamiento se utilizaron distintas fuentes: entrevistas directas con usuarios y responsables, observación de las tareas en el entorno de trabajo y documentación existente (procedimientos internos y reportes de incidencias). Esto permitió obtener una visión equilibrada entre la experiencia práctica de los técnicos y la información formal disponible.

Los principales actores considerados fueron los técnicos del área y sus responsables directos. Ellos aportaron tanto la perspectiva operativa como la de gestión, lo que facilitó una visión integral de la problemática.

Entre las técnicas aplicadas se utilizaron entrevistas semiestructuradas para profundizar en problemas concretos, encuestas breves para medir la frecuencia de ciertas dificultades, y observación directa de la operatoria diaria. La combinación de estas técnicas permitió confirmar la información y detectar coincidencias en los planteos de los distintos actores.

Finalmente, los hallazgos fueron validados mediante una revisión general con el área involucrada. Esto permitió corregir detalles menores y dar por confirmados los resultados de la elicitación, que sirvieron de base para la definición de los requerimientos posteriores.

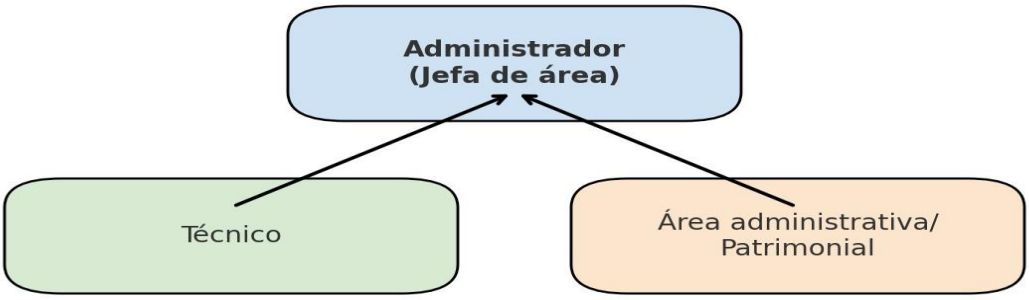
Requerimientos Funcionales (RF)

ID	Requerimiento Funcional
RF1	El sistema debe permitir registrar equipos informáticos (PC, impresoras, escáneres y monitores).
RF2	El sistema debe asociar cada equipo a una dependencia/área específica.
RF3	El sistema debe permitir consultar el stock de equipos filtrado por tipo y dependencia.
RF4	El sistema debe registrar altas, bajas y modificaciones de equipos.
RF5	El sistema debe permitir llevar un historial de reparaciones realizadas sobre cada equipo.
RF6	El sistema debe registrar quién realizó cada operación sobre los equipos.
RF7	El sistema debe generar reportes de stock y reparaciones.
RF8	El sistema debe permitir búsquedas rápidas de equipos por número de inventario o dependencia.
RF9	El sistema debe permitir exportar datos a Excel o PDF.
RF10	El sistema debe gestionar distintos perfiles de usuario: Lectura, Edición, Administración

Requisitos No Funcionales (RNF)

ID	Categoría	Detalle
RNF1	Rendimiento	- Las consultas y reportes deben generarse en menos de 2 segundos.
		- El sistema debe soportar al menos 10.000 registros de equipos sin degradar el rendimiento.
RNF2	Seguridad	- Acceso mediante usuario y contraseña.
		- Gestión de diferentes niveles de permisos.
		- Registro de auditoría con fecha y hora de cada operación.
RNF3	Disponibilidad	- El sistema debe estar disponible en horario de oficina todos los días hábiles.
		- Debe funcionar en red local (intranet), sin requerir acceso a internet.
RNF4	Usabilidad	- La interfaz debe ser simple e intuitiva, pensada para usuarios con conocimientos técnicos básicos.
		- Debe contar con diseño adaptable a pantallas de distintos tamaños, incluyendo acceso básico desde celulares.
RNF5	Mantenibilidad y escalabilidad	- El sistema debe permitir su expansión futura para incluir otros dispositivos o funcionalidades (como mantenimientos preventivos).
		- La arquitectura debe permitir una futura migración hacia una versión 100% web.
RNF6	Restricciones técnicas	- La implementación debe realizarse en Java con base de datos MySQL.
		- Debe funcionar en sistemas Windows 7, 10 y 11.
		- Posibilidad de exportar datos en formatos estándar (Excel, PDF).

Mapa de actores



Actores principales

Técnicos del área de informática

Rol actual: Mantienen la planilla de Excel, actualizan datos de stock y registran reparaciones.

Relación con el sistema: Serán los principales usuarios de la aplicación, con permisos de carga, consulta y actualización de información.

Jefa del área de informática

Rol actual: Supervisa los procesos de control de equipos y toma decisiones a partir de los reportes que generan los técnicos.

Relación con el sistema: Tendrá permisos de administración (gestión de usuarios, control de permisos, validación de datos). También dependerá del sistema para obtener reportes inmediatos y confiables.

Usuarios de dependencias

Rol actual: Reportan fallas de equipos a través de correos electrónicos o formularios. No llevan un control sistemático de su stock.

Relación con el sistema: Su interacción será indirecta. No ingresarán datos, pero sus solicitudes (mails o formularios) generarán los registros que luego procesará el área de informática.

Actores secundarios

Áreas de gestión administrativa y patrimonial

Rol actual: Requieren información sobre la cantidad y el estado de los equipos informáticos para fines de auditoría y control.

Relación con el sistema: Consumidores de reportes emitidos por el área de informática, aunque no serán usuarios directos.

Dirección general

Rol actual: Toma decisiones sobre presupuesto, compra de nuevos equipos y asignación de recursos.

Relación con el sistema: No utilizará el sistema de forma directa, pero se beneficia de la información consolidada que provee la jefatura de informática.

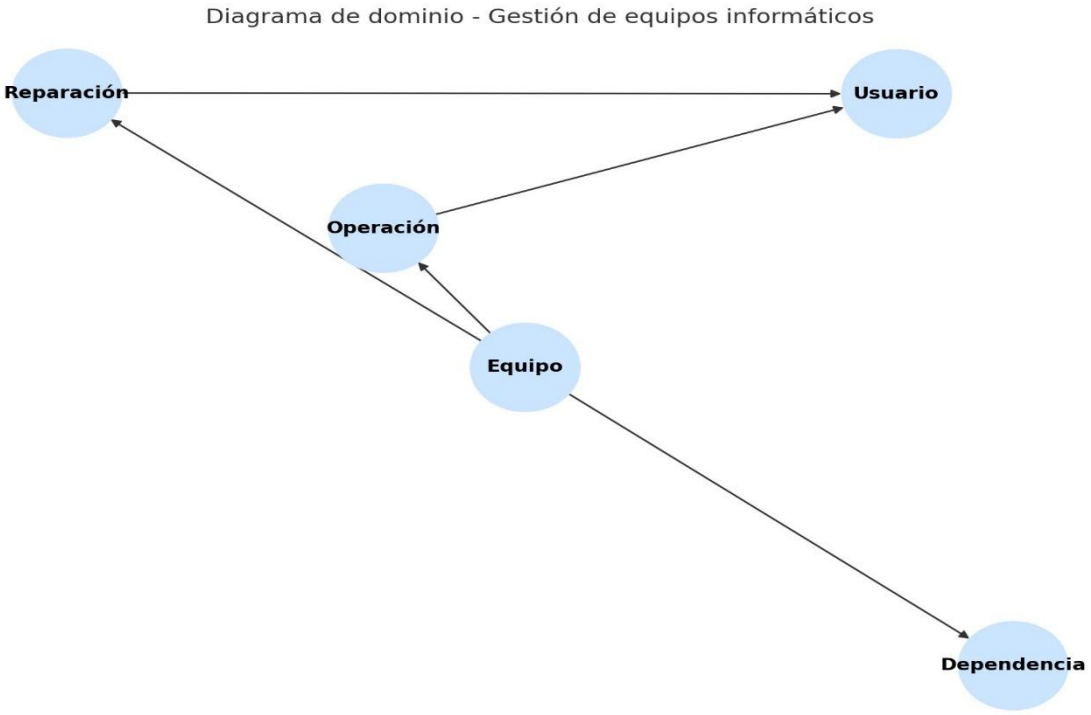
Conocimiento del negocio

El sistema que se va a desarrollar debe permitir la gestión y monitoreo de equipos informáticos de la organización y realizar, de esta manera, el registro y procesamiento de la información necesaria para optimizar su utilización y mantenimiento.
En un entorno institucional, los dispositivos informáticos —tales como computadoras, impresoras, escáneres y monitores— constituyen recursos críticos para el desarrollo de las actividades diarias. Cada equipo debe estar asignado a una dependencia o área específica, lo que permite garantizar un uso eficiente y facilitar su control.
Para asegurar la trazabilidad de los activos, resulta necesario registrar las altas, bajas y modificaciones de equipos, así como conservar un historial de reparaciones realizadas. De esta manera, se logra un seguimiento integral de cada dispositivo durante su ciclo de vida. Asimismo, la gestión centralizada de estos datos permite obtener reportes de stock, detectar necesidades de renovación y planificar adecuadamente las tareas de soporte.
El sistema propuesto, además de registrar y administrar los equipos, debe garantizar la seguridad de la información mediante el uso de perfiles de usuario con diferentes niveles de acceso. Con esto se busca proteger la integridad de los datos y asegurar que solo personal autorizado pueda realizar operaciones críticas.

Diagrama de dominio

Para representar el modelo de negocio se consideran las principales entidades, atributos y asociaciones que forman parte del contexto:

- **Equipo** (código inventario, tipo, marca, modelo, estado).
- **Dependencia** (nombre, responsable).
- **Operación** (alta, baja, modificación).
- **Reparación** (fecha, descripción, técnico responsable).
- **Usuario** (nombre, rol, perfil de acceso).



Propuesta de solución

Se plantea el desarrollo de un sistema de gestión de inventario y reparaciones de equipos informáticos, implementado en Java con base de datos MySQL, diseñado para funcionar en la intranet de la organización.

El sistema permitirá:
Registrar y administrar equipos informáticos (PC, impresoras, escáneres y monitores).
Asociar cada equipo a una dependencia específica.
Gestionar altas, bajas y modificaciones de equipos.
Registrar y actualizar solicitudes de reparación, manteniendo un historial con detalles de fallas, fechas y técnicos responsables.
Generar reportes de stock y reparaciones, con posibilidad de exportación en formatos estándar (Excel y PDF).
Incorporar un esquema de perfiles de usuario (lectura, edición, administración) que garantice la seguridad y el control de los datos.

A futuro, la arquitectura propuesta permitirá ampliar el alcance del sistema, incorporando nuevas funcionalidades como el seguimiento de mantenimientos preventivos o la migración hacia una versión 100% web.

Objetivos principales

Centralizar y normalizar la información sobre equipos informáticos (PC, impresoras, escáneres y monitores).
Agilizar la gestión de reparaciones, incorporando registros claros y trazables desde el momento en que un usuario informa la falla hasta la resolución final.
Brindar reportes confiables y actualizados que permitan conocer rápidamente la distribución de equipos por dependencia, el estado de stock y los equipos en reparación.
Optimizar la toma de decisiones, asegurando a la jefatura información inmediata y precisa.

Alcance de la solución

El sistema permitirá:

Registrar altas, bajas y modificaciones de equipos.
Asociar cada equipo a una dependencia específica.
Gestionar solicitudes de reparación, indicando fecha, estado y acciones realizadas.
Generar reportes filtrados por dependencia, tipo de equipo o estado (operativo, en reparación, dado de baja).
Definir distintos roles de usuario (administrador, técnico, consulta).

Beneficios esperados

Confiabilidad: Eliminación de errores frecuentes en planillas manuales y aseguramiento de la integridad de los datos.
Rapidez: Acceso inmediato a información que hoy requiere búsquedas manuales y relevamientos en oficina.
Escalabilidad: Posibilidad de incorporar nuevas funciones a futuro, como control de insumos, historial de mantenimiento o acceso web ampliado.
Soporte a la gestión: Información clara para auditorías, controles patrimoniales y planificación de recursos.

Tecnologías a utilizar

Lenguaje de programación: Java.
Base de datos: MySQL.
Entorno de ejecución: Aplicación de escritorio en entorno Windows (versiones 7, 10 y 11).
Infraestructura: Instalación local en la intranet del área de informática.

Inicio del análisis: Casos de uso

Requerimiento	Caso de uso	Actor principal	Comentario
RF1, RF2, RF4, RF6	CU1 Registrar equipo	Técnico	Registra nuevo equipo y lo asocia a una dependencia.
RF2, RF4, RF6	CU2 Modificar datos de equipo	Técnico	Actualiza atributos o dependencia de un equipo.
RF4, RF6	CU3 Dar de baja equipo	Administrador	Marca un equipo como dado de baja.
RF3, RF8	CU4 Consultar stock por dependencia	Administrador / Área administrativa	Consulta equipos filtrados por dependencia y/o tipo.
RF5, RF6	CU5 Registrar solicitud de reparación	Técnico	Crea y vincula solicitud de reparación a un equipo.
RF5, RF6	CU6 Actualizar estado de reparación	Técnico	Actualiza estado y detalles de una reparación.
RF7, RF3, RF5	CU7 Generar reportes	Administrador / Área administrativa	Genera reportes de stock y de reparaciones.
RF10	CU8 Asignar roles y permisos	Administrador	Otorga o revoca permisos según perfil.
RF10	CU9 Gestionar usuarios	Administrador	Alta, modificación y baja de usuarios.

Tabla de Casos de uso - trazabilidad

Descripcion de casos de Uso

CU1 – Registrar equipo	
Actor principal:	Técnico
Requerimientos asociados:	RF1, RF2, RF4, RF6
Descripción:	Permite dar de alta un equipo informático y asociarlo a una dependencia.
Precondiciones:	Usuario autenticado con perfil Técnico; la dependencia existe; el número de inventario
Flujo principal:	1. Se selecciona «Registrar equipo». 2. El sistema muestra el formulario de alta. 3. Se ingresan tipo, marca, modelo, nro. de inventario/serie, dependencia y estado ini 4. (Opcional) Se adjuntan observaciones/foto/etiqueta patrimonial. 5. Se confirma. 6. El sistema valida unicidad de nro. de inventario y campos obligatorios. 7. El sistema crea el registro, asocia la dependencia y genera auditoría. 8. El sistema muestra confirmación y datos clave.
Postcondiciones:	Equipo incorporado al inventario con historial inicial y auditoría.
Flujos alternativos:	A1: nro. inventario duplicado → el sistema informa y solicita corrección. A2: dependencia inexistente → el sistema informa y solicita una válida.
Excepciones:	Error de base de datos → se registra evento y se informa indisponibilidad.

CU2 – Modificar datos de equipo	
Actor principal:	Técnico
Requerimientos asociados:	RF2, RF4, RF6
Descripción:	Permite actualizar atributos del equipo, como dependencia, estado o datos técnicos.
Precondiciones:	Usuario autenticado con perfil Técnico; equipo existente.
Flujo principal:	1. Se accede a «Administrar equipo». 2. Se localiza el equipo (nro. inventario, tipo o dependencia). 3. El sistema muestra la ficha actual. 4. Se editan campos permitidos. 5. Se confirma. 6. El sistema valida cambios y registra auditoría. 7. El sistema guarda y confirma.
Postcondiciones:	Ficha del equipo actualizada con auditoría.
Flujos alternativos:	A1: equipo no encontrado → el sistema informa «sin coincidencias». A2: cambio no permitido → el sistema informa restricción de edición.
Excepciones:	Error de concurrencia → se solicita recargar.

CU3 – Dar de baja equipo	
Actor principal:	Administrador
Requerimientos asociados:	RF4, RF6
Descripción:	Permite dar de baja un equipo, dejándolo fuera del stock operativo.
Precondiciones:	Usuario autenticado con perfil Administrador; equipo existente; sin bloqueos críticos
Flujo principal:	1. Se localiza el equipo. 2. Se selecciona «Dar de baja». 3. Se ingresa motivo y, si corresponde, destino (resguardo/reciclaje). 4. Se confirma. 5. El sistema valida condiciones (ej. reparaciones abiertas). 6. El sistema cambia estado a «Baja», registra auditoría y retira de listados. 7. El sistema muestra confirmación.
Postcondiciones:	Equipo marcado como «Baja» con motivo y fecha.
Flujos alternativos:	A1: reparación abierta → el sistema solicita confirmación adicional o bloquea.
Excepciones:	Falta de permisos → se informa y se deniega la acción.

CU4 – Consultar stock por dependencia	
Actor principal:	Administrador / Área administrativa
Requerimientos asociados:	RF3, RF8
Descripción:	Permite listar equipos existentes, filtrando por dependencia, tipo o estado.
Precondiciones:	Usuario autenticado; existencia de datos.
Flujo principal:	1. Se accede a «Consultar stock». 2. Se aplican filtros de búsqueda (dependencia, tipo de equipo, estado). 3. El sistema genera el listado. 4. El sistema muestra los resultados en pantalla. 5. (Opcional) Se exporta el resultado.
Postcondiciones:	Información disponible para visualización o exportación.
Flujos alternativos:	Búsqueda sin resultados → el sistema informa «sin coincidencias».
Excepciones:	Tiempo de respuesta excedido → se sugiere acotar filtros.

CU5 – Registrar solicitud de reparación	
Actor principal:	Técnico
Requerimientos asociados:	RF5, RF6
Descripción:	Permite registrar una solicitud de reparación para un equipo.
Precondiciones:	Usuario con perfil Técnico; equipo existente; equipo no en baja.
Flujo principal:	1. Se busca el equipo en el inventario. 2. Se crea nueva solicitud de reparación. 3. Se ingresan datos de falla, fecha y prioridad. 4. (Opcional) Se asigna técnico responsable. 5. Se confirma la operación. 6. El sistema valida y registra la solicitud. 7. El sistema registra auditoría y muestra confirmación.
Postcondiciones:	Solicitud vinculada al equipo.
Flujos alternativos:	Sin técnico asignado → solicitud queda «Pendiente de asignación».
Excepciones:	Error de validación → se informa y se mantiene en edición.

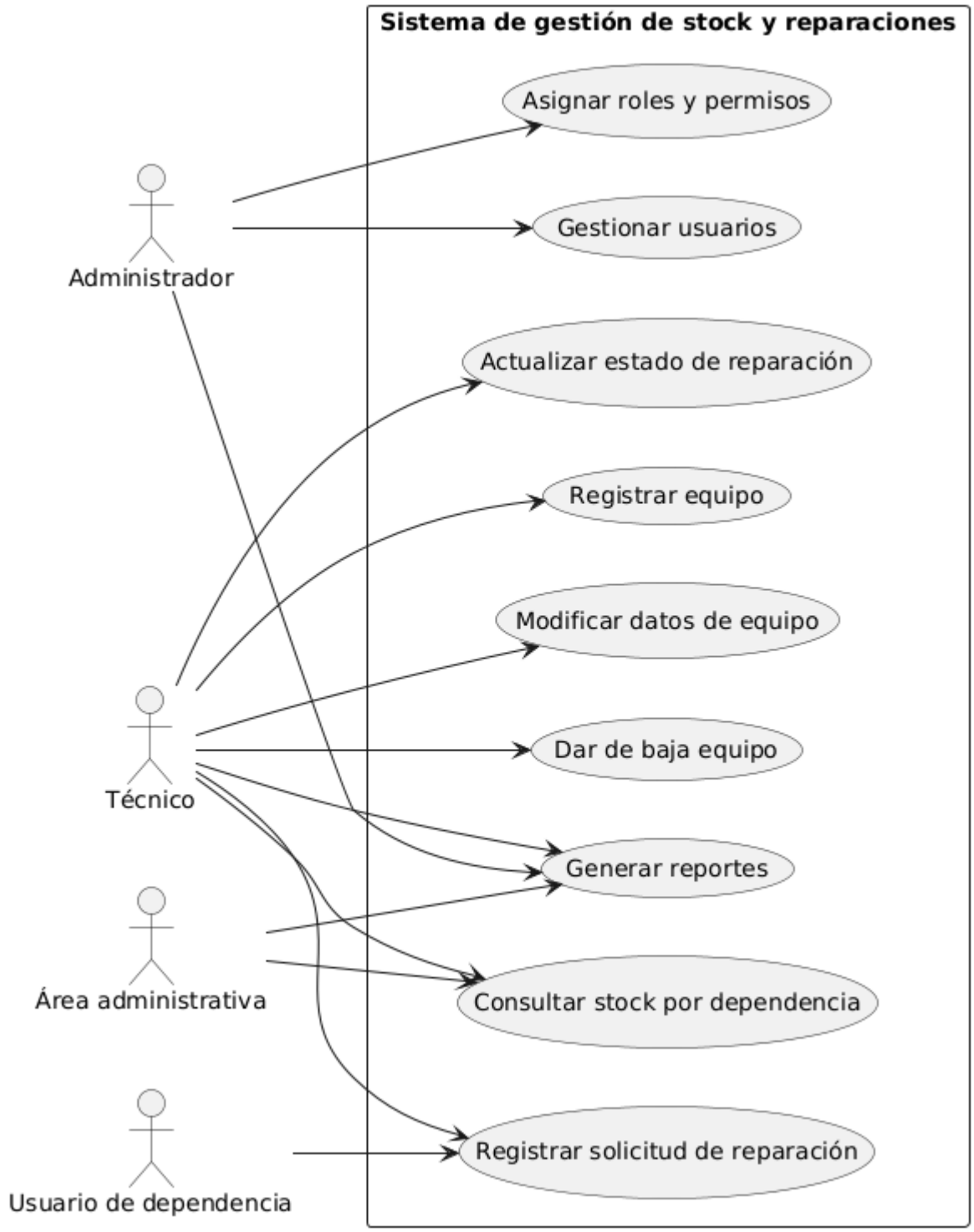
CU6 – Actualizar estado de reparación	
Actor principal:	Técnico
Requerimientos asociados:	RF5, RF6
Descripción:	Permite actualizar estado y detalles de una reparación existente.
Precondiciones:	Reparación existente; usuario con perfil Técnico.
Flujo principal:	1. Se accede a la reparación. 2. Se cambia el estado (En curso / En espera / Finalizada / No reparable). 3. Se ingresan observaciones o insumos. 4. Se guarda la actualización. 5. El sistema valida, guarda y registra auditoría.
Postcondiciones:	Reparación actualizada en el historial del equipo.
Flujos alternativos:	Si se marca «No reparable» → el sistema sugiere baja del equipo.
Excepciones:	Reparación cerrada previamente → el sistema impide modificaciones.

CU7 – Generar reportes	
Actor principal:	Administrador / Área administrativa
Requerimientos asociados:	RF7, RF3, RF5
Descripción:	Permite generar reportes de stock y reparaciones.
Precondiciones:	Usuario autenticado; datos existentes; permisos para exportar.
Flujo principal:	1. Se selecciona «Generar reporte». 2. Se define tipo de reporte y filtros. 3. El sistema procesa y muestra vista previa. 4. Se elige formato (Excel/PDF). 5. El sistema genera el archivo y ofrece descarga.
Postcondiciones:	Reporte generado y disponible para descarga.
Flujos alternativos:	Sin datos → el sistema genera reporte vacío con mensaje.
Excepciones:	Error de exportación → se informa y se sugiere reintentar.

CU8 – Asignar roles y permisos	
Actor principal:	Administrador
Requerimientos asociados:	RF10
Descripción:	Permite modificar roles y permisos de usuarios.
Precondiciones:	Sesión iniciada con perfil Administrador; usuario objetivo existente.
Flujo principal:	1. Se localiza el usuario. 2. Se asignan o revocan roles/permisos. 3. Se confirma la operación. 4. El sistema valida y guarda cambios, registrando auditoría. 5. El sistema notifica que aplican al próximo inicio.
Postcondiciones:	Permisos del usuario actualizados.
Flujos alternativos:	Usuario con sesión activa → cambios aplican al próximo inicio.
Excepciones:	Usuario inexistente → se informa y finaliza.

CU9 – Gestionar usuarios	
Actor principal:	Administrador
Requerimientos asociados:	RF10
Descripción:	Permite crear, modificar o deshabilitar usuarios.
Precondiciones:	Sesión iniciada con perfil Administrador.
Flujo principal:	Alta: 1. Completar datos y rol inicial. 2. Confirmar. 3. El sistema valida y crea. Modificación/Deshabilitar: 1. Buscar usuario. 2. Editar datos o marcar «Deshabilitar». 3
Postcondiciones:	Usuario creado, modificado o deshabilitado.
Flujos alternativos:	Alta duplicada → se informa error. Deshabilitar con sesión activa → cambios aplican al cerrar sesión.
Excepciones:	Falta de permisos → se deniega la operación.

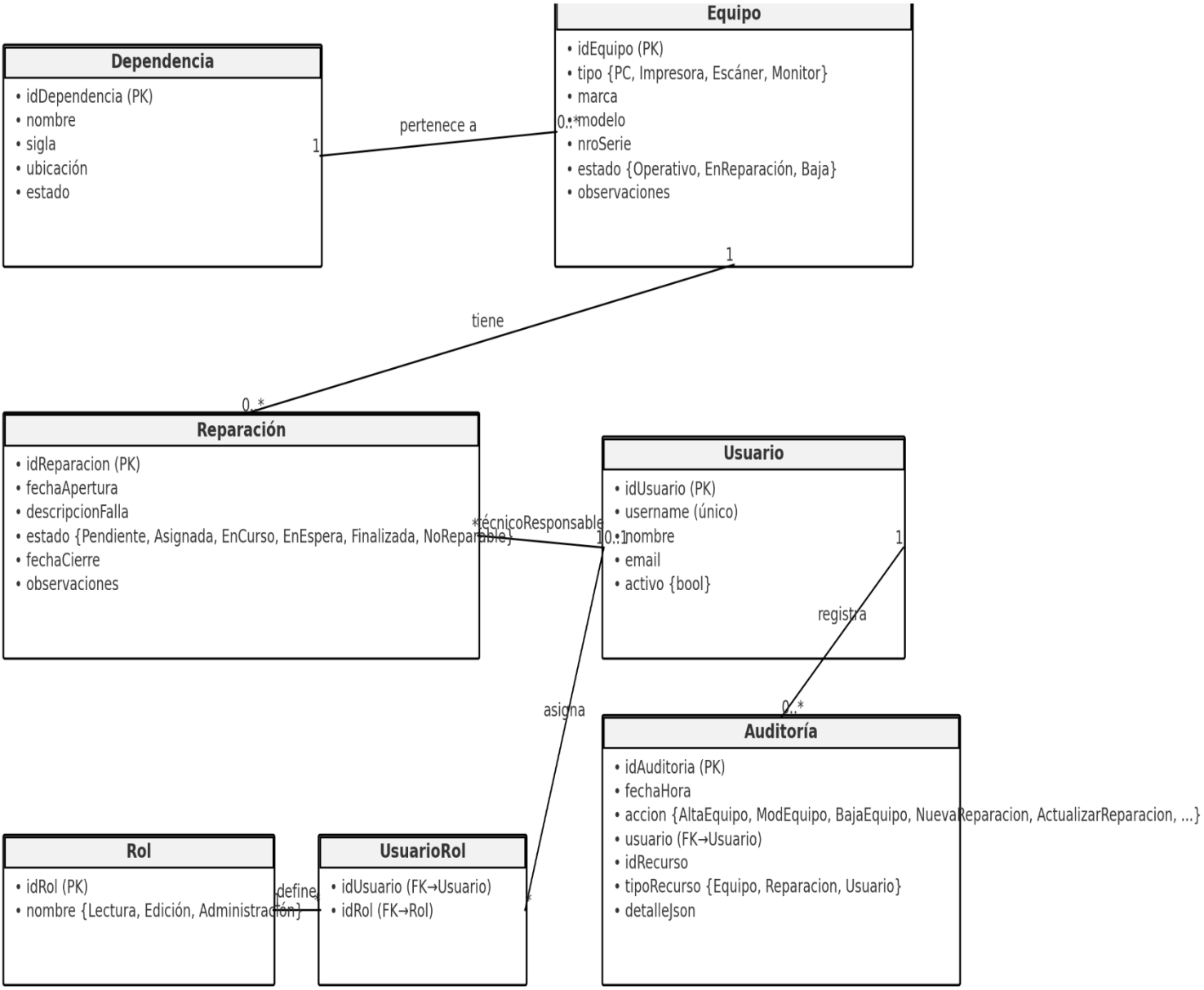
Diagrama de casos de uso



Etapa de diseño

2.1 Diagrama de clases

El diagrama representa las entidades principales y sus relaciones dentro del sistema de gestión de inventario y reparaciones. Se definen las clases Dependencia, Equipo, Reparación, Usuario, Rol, UsuarioRol y Auditoría, con sus atributos clave. Se establecen multiplicidades que reflejan el dominio: cada dependencia posee múltiples equipos; cada equipo puede tener varias reparaciones; las reparaciones pueden asignarse a un único técnico; y cada usuario puede tener uno o más roles, además de registrar acciones en la auditoría. El modelo permite gestionar inventario, registrar reparaciones y mantener trazabilidad mediante el historial de auditoría.

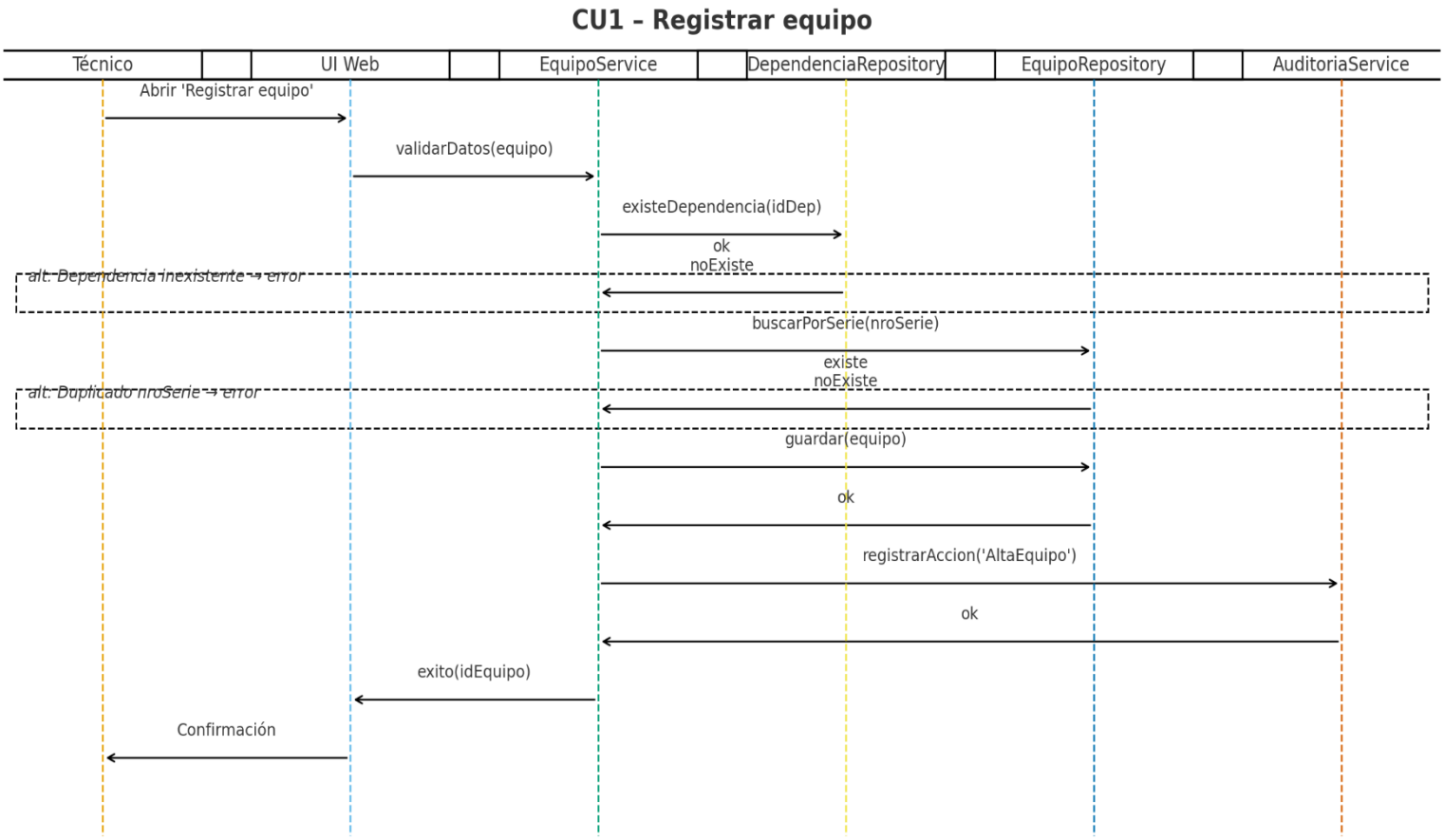


Selección de casos de uso para diagramas de secuencia:

Dado que el sistema abarca múltiples funcionalidades, no se elaboraron diagramas de secuencia para todos los casos de uso, sino que se decidió focalizar en aquellos más representativos y complejos. En este caso, se seleccionaron tres casos de uso clave para detallar mediante diagramas de secuencia. La elección de estos casos de uso – Registrar equipo, Registrar reparación y Generar reporte – responde a que cubren los módulos principales del sistema (gestión de inventario, gestión de reparaciones y generación de informes, respectivamente) y ejemplifican las interacciones críticas entre los componentes. Mediante esta selección acotada se abarcan los procesos esenciales del negocio sin incurrir en redundancias, mostrando cómo participan las distintas capas de la aplicación (interfaz, lógica de negocio y persistencia) y cómo se manejan escenarios alternativos en cada flujo. A continuación se presentan los diagramas de secuencia correspondientes a cada caso de uso elegido, junto con una breve descripción de cada uno.

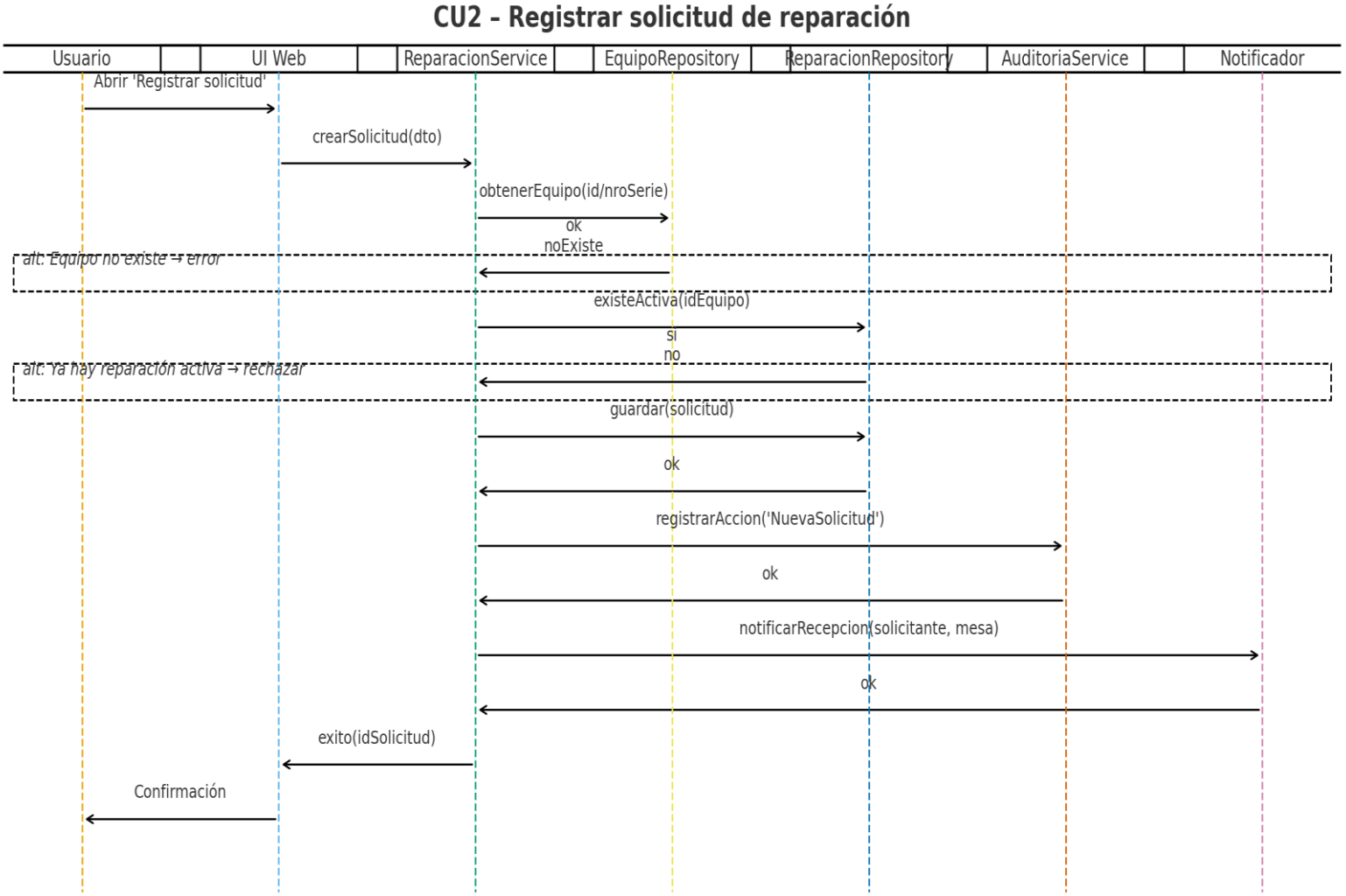
2.2 Diagrama de secuencia – Caso de uso "Registrar equipo"

El diagrama ilustra la interacción entre el Técnico, la interfaz de usuario, los servicios de negocio y los repositorios durante el proceso de alta de un nuevo equipo. El flujo principal incluye la apertura del formulario, la validación de la dependencia y la verificación de duplicados en el número de serie, finalizando con la persistencia del equipo y el registro en la auditoría. Se representan además dos escenarios alternativos mediante fragmentos alt: uno para el caso de duplicación del número de serie y otro para la inexistencia de la dependencia. Este esquema evidencia la coordinación entre capas y la gestión de errores prevista.



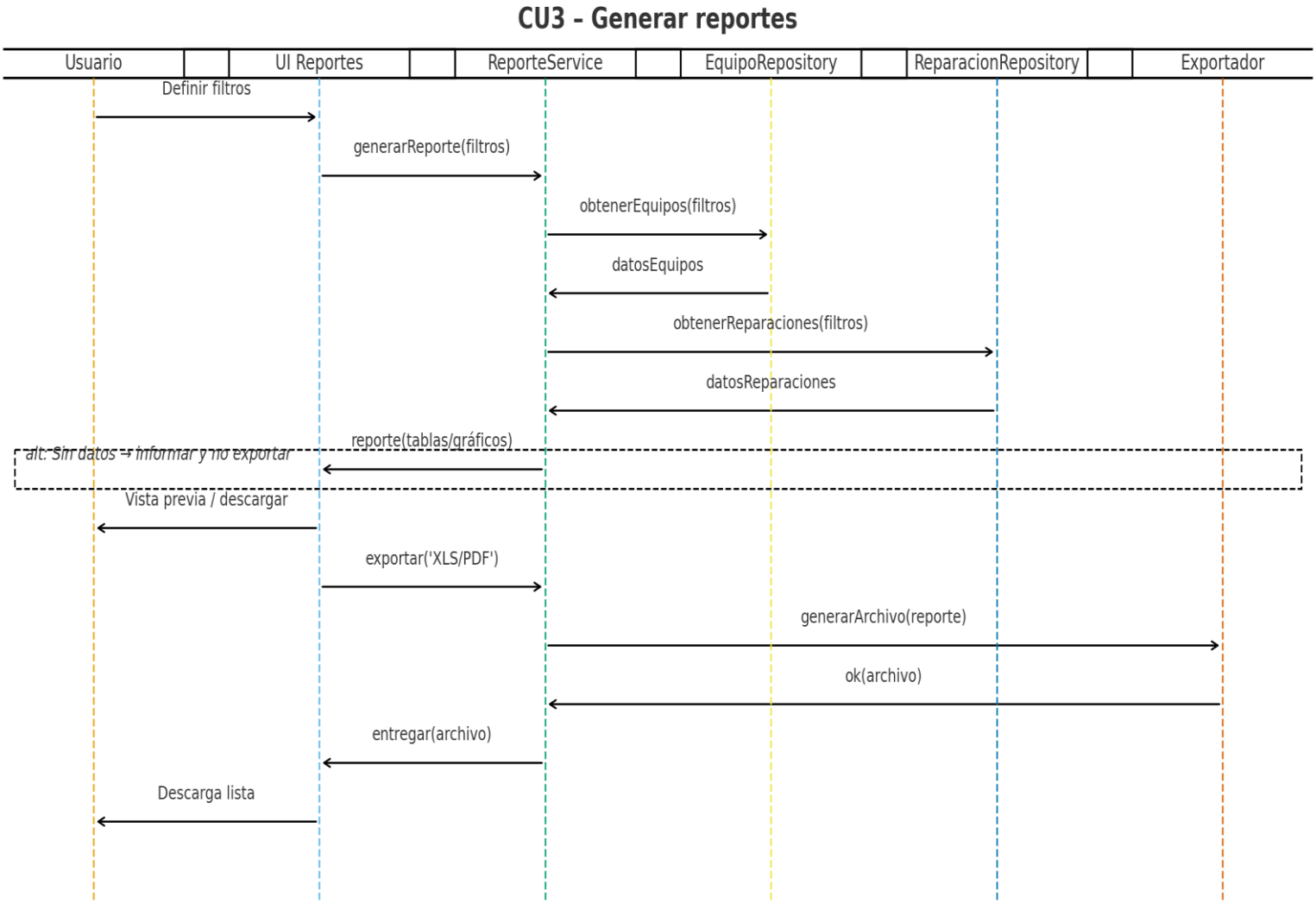
2.3 Diagrama de secuencia – Caso de uso "Registrar solicitud de reparación"

El diagrama de secuencia del caso de uso Registrar solicitud de reparación muestra la interacción entre el Técnico, la interfaz de usuario, los servicios de negocio y los repositorios durante el proceso de registro de una nueva solicitud de reparación de un equipo. El flujo principal abarca la carga de los datos de la falla, la verificación de que el equipo existe en el inventario (y no tiene ya una reparación en curso) y la creación del registro de reparación asociada, actualizando el estado del equipo a “en reparación”. Finalmente, se persiste la nueva reparación en la base de datos correspondiente y se registra la acción en la auditoría. Se contemplan también dos escenarios alternativos mediante fragmentos alt: uno en caso de que el equipo ingresado no exista en el sistema, y otro si el equipo ya tiene una reparación activa. En cualquiera de esos casos el sistema aborta el flujo principal y notifica la situación al Técnico. Este esquema destaca la interacción coordinada entre los módulos de inventario y reparaciones, aplicando las reglas de negocio necesarias para mantener la consistencia de la información.



2.4 Diagrama de secuencia – Caso de uso "Generar reportes"

El diagrama de secuencia del caso de uso Generar reportes describe la interacción entre un Usuario autorizado (por ejemplo, un técnico o administrador), la interfaz de generación de informes, los servicios de negocio y los repositorios durante el proceso de obtención de un reporte del sistema. En el flujo principal, el Usuario especifica los criterios de filtrado deseados (por dependencia, tipo de equipo, estado, etc.), tras lo cual el sistema recopila los datos relevantes de las bases de datos de equipos y/o reparaciones y genera el informe solicitado, que luego es mostrado en pantalla con opción de exportación (por ejemplo, a Excel o PDF). Se incluye un flujo alternativo mediante un fragmento alt en caso de que no existan datos que coincidan con los filtros proporcionados; en tal situación, el sistema informa al Usuario la ausencia de resultados en lugar de generar un reporte vacío. De este modo, el diagrama evidencia la integración de los módulos de inventario y reparaciones para brindar información consolidada, así como el manejo adecuado de situaciones en las que no hay datos disponibles para reportar.



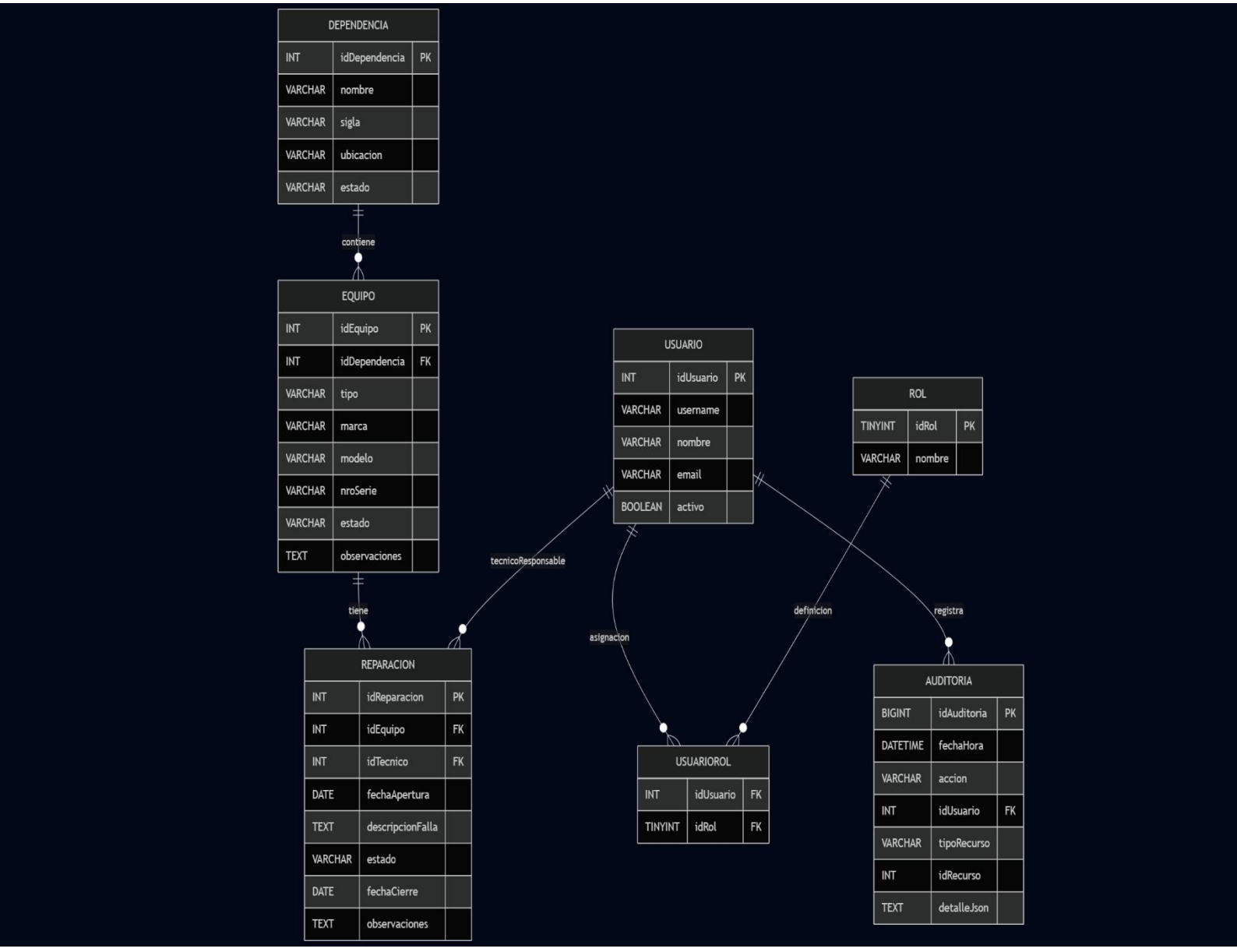
Etapa de implementación

3.1. Introducción

En esta etapa se presenta una aproximación inicial a la implementación del sistema con el objetivo de demostrar la factibilidad técnica de lo diseñado. La evidencia se limita a la definición de estructuras de datos en MySQL y a la ejecución de operaciones básicas sobre la base de datos, sin constituir aún un desarrollo completo de la aplicación. Las actividades incluyeron: creación de tablas, carga mínima de datos de prueba y ejecución de consultas alineadas a los requerimientos y casos de uso priorizados.

DER de la Base de Datos a utilizar

Para una mejor visualización del mismo, también se adjunta por fuera del informe en formato PNG

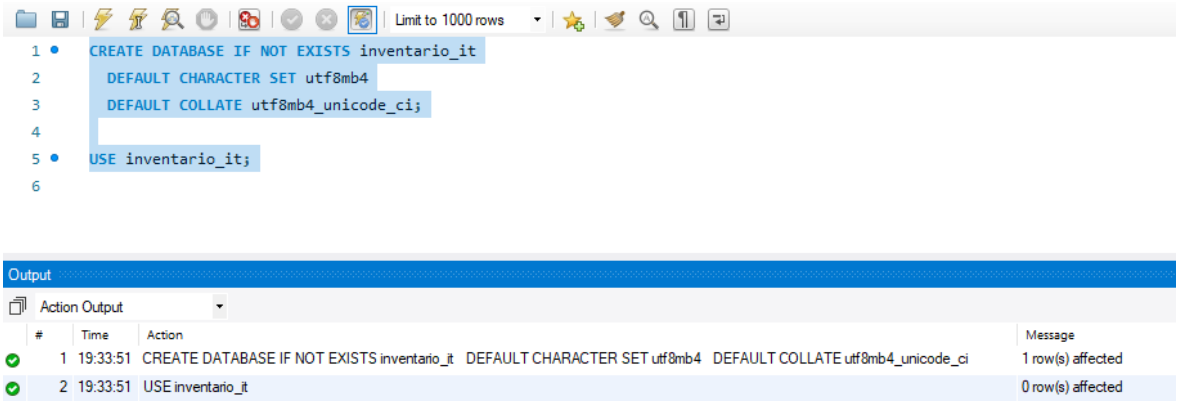


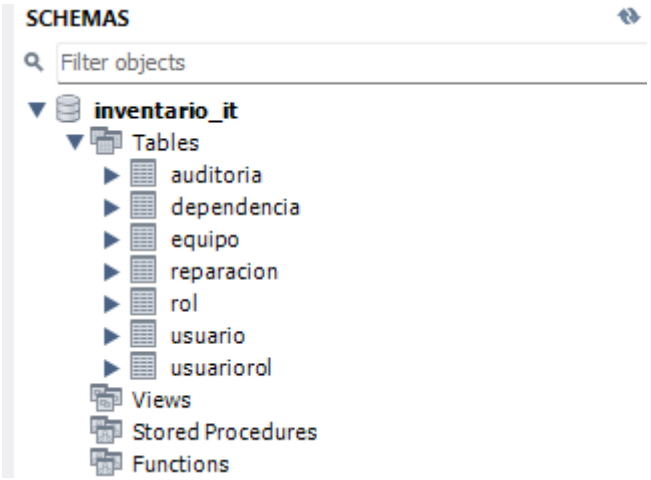
3.2. Alcance y supuestos de la implementación

La implementación se acota al modelo de datos que da soporte a la gestión de equipos, dependencias, reparaciones, usuarios/roles y auditoría. Se asume un entorno de **intranet** con servidor MySQL accesible desde los puestos del área de soporte. La carga de datos se realizó con un conjunto reducido de ejemplos representativos para validar reglas de unicidad, integridad referencial y trazabilidad básica (altas y modificaciones).

3.3. Entorno y criterios de configuración

Las pruebas se realizaron en un motor **MySQL** local, con esquema dedicado al proyecto. Se aplicaron convenciones de nombres coherentes con el diseño (claves primarias con sufijo “id*”, claves foráneas explicitas y dominios de valores acordes a las entidades definidas). Se verificó la restricción de unicidad en atributos críticos (p. ej., número de serie) y la presencia de claves foráneas para garantizar consistencia entre equipos, dependencias y reparaciones.

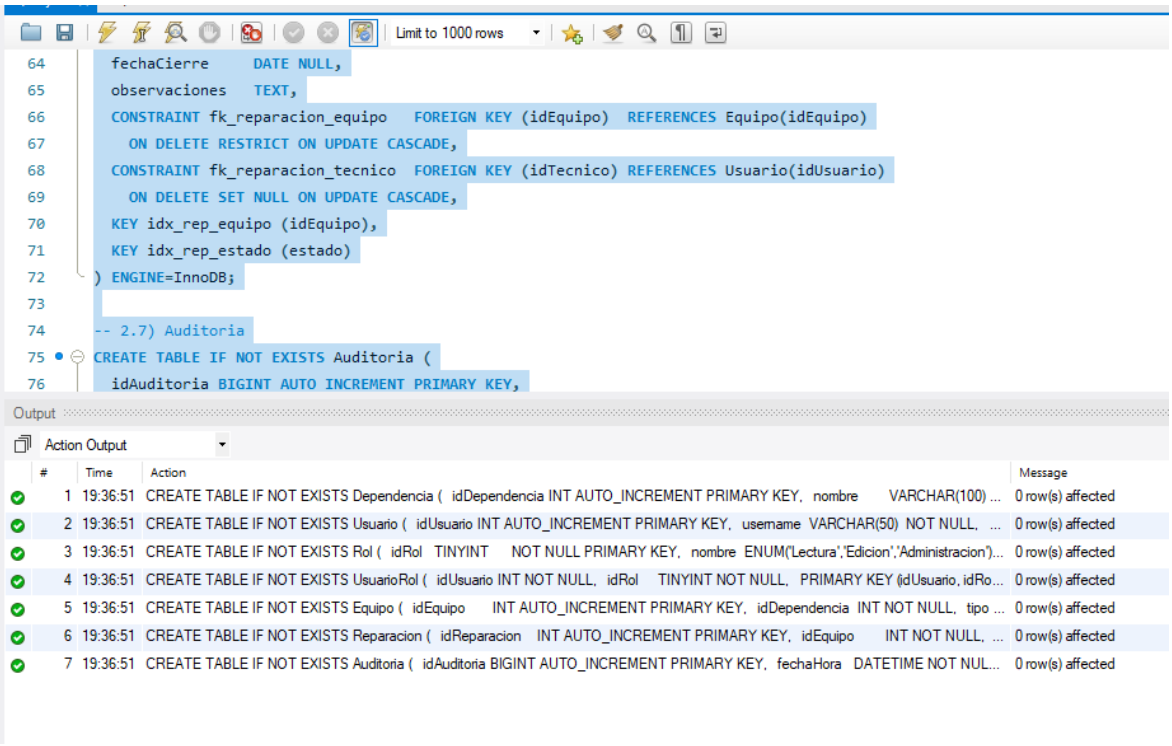




3.4. Creación de tablas

Se evidencian las tablas principales del sistema: **Dependencia**, **Equipo**, **Reparacion**, **Usuario**, **Rol**, **UsuarioRol** y **Auditoria**. La estructura implementada respeta las cardinalidades del diagrama de clases y del modelo ER, asegurando:

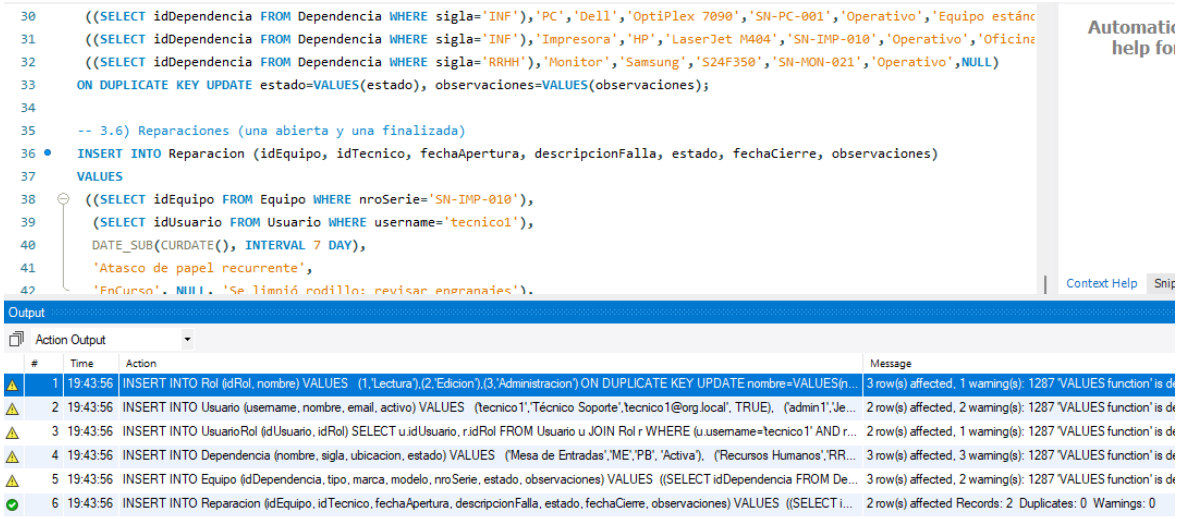
- Integridad referencial entre **Equipo** y **Dependencia**; y entre **Reparacion** y **Equipo**.
- Relación **muchos-a-muchos** entre **Usuario** y **Rol** mediante **UsuarioRol**.
- Registro de eventos en **Auditoria** ante operaciones relevantes.



3.5. Operaciones básicas

Se efectuó una carga mínima de datos de ejemplo para validar altas, consultas y bajas controladas. La evidencia se presenta como capturas de pantalla del cliente SQL utilizado.

- **Alta de datos**: inserción de dependencias y equipos con número de serie único.
- **Consultas**: verificación de equipos por dependencia y estado; consulta del historial de reparaciones por equipo.
- **Baja controlada**: marcado de equipos en estado “Baja” y verificación de su exclusión de listados operativos.



```
30      ((SELECT idDependencia FROM Dependencia WHERE sigla='INF'),'PC','Dell','OptiPlex 7090','SN-PC-001','Operativo','Equipo estánc
31      ((SELECT idDependencia FROM Dependencia WHERE sigla='INF'),'Impresora','HP','LaserJet M404','SN-IMP-010','Operativo','Oficina
32      ((SELECT idDependencia FROM Dependencia WHERE sigla='RRHH'),'Monitor','Samsung','S24F350','SN-MON-021','Operativo',NULL)
33    ON DUPLICATE KEY UPDATE estado=VALUES(estado), observaciones=VALUES(observaciones);
34
35    -- 3.6) Reparaciones (una abierta y una finalizada)
36    INSERT INTO Reparacion (idEquipo, idTecnico, fechaApertura, descripcionFalla, estado, fechaCierre, observaciones)
37    VALUES
38      ((SELECT idEquipo FROM Equipo WHERE nroSerie='SN-IMP-010'),
39       (SELECT idUsuario FROM Usuario WHERE username='tecnico1'),
40       DATE_SUB(CURDATE(), INTERVAL 7 DAY),
41       'Atasco de papel recurrente',
42       'EnCurso', NULL, 'Se limpió rodillo; revisar engranajes'),
```

Automatic help fo

Context Help | Snip

Output

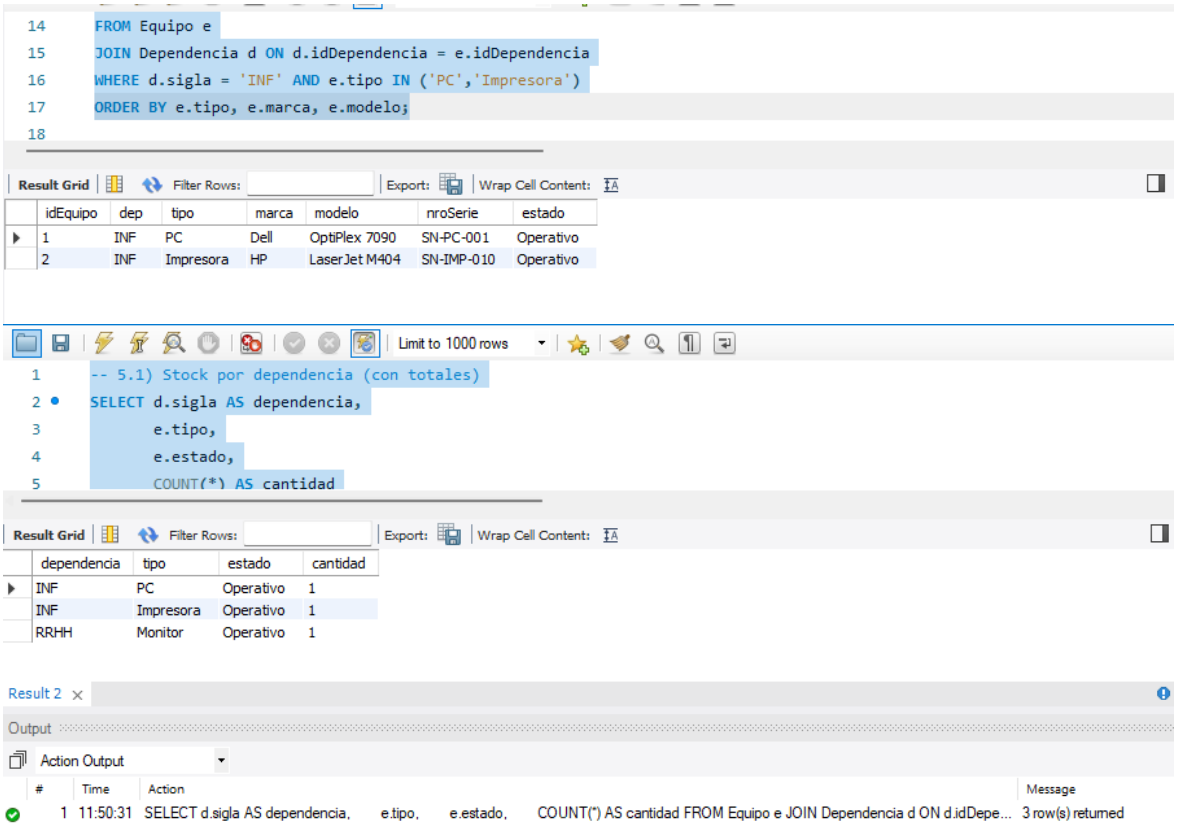
Action Output

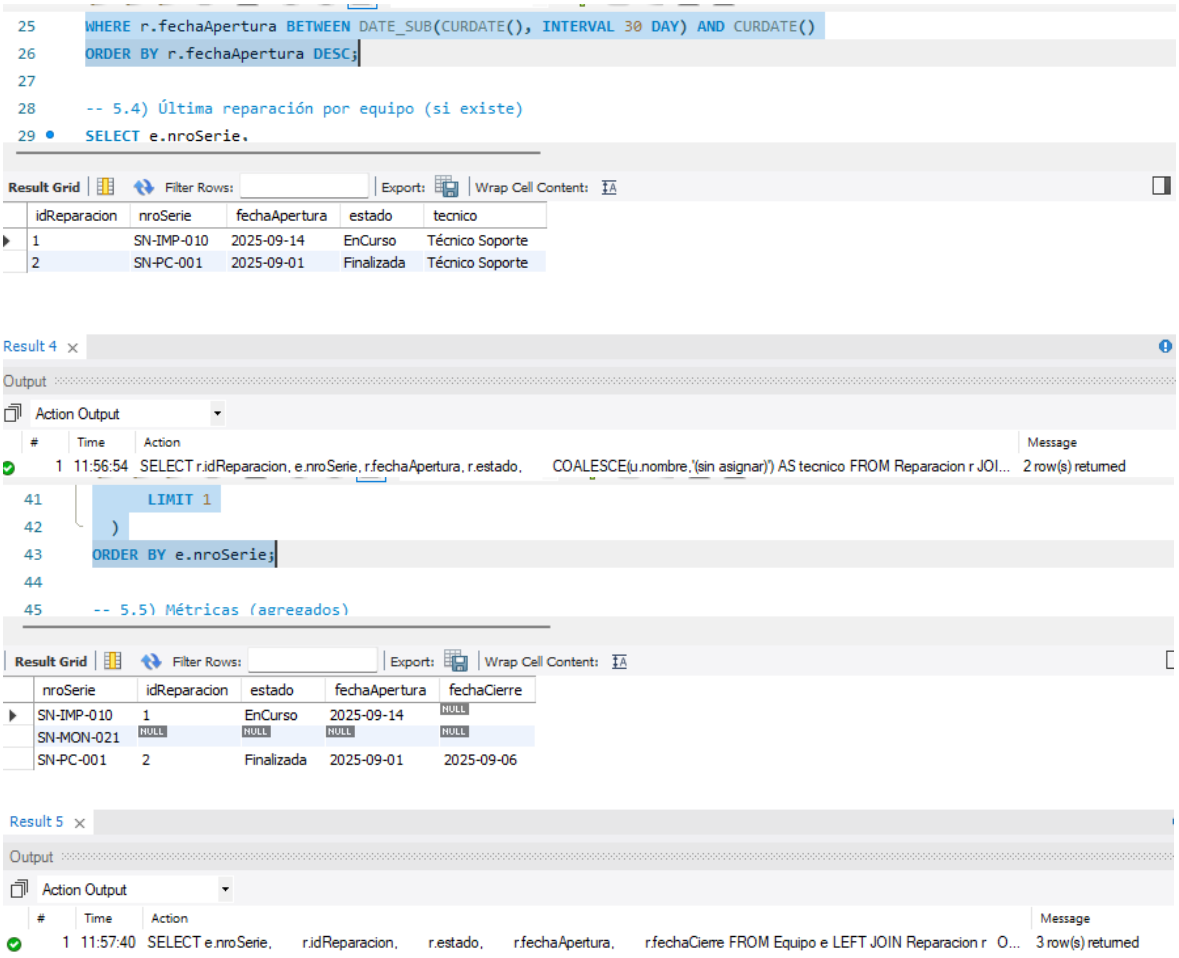
#	Time	Action	Message
1	19:43:56	INSERT INTO Rol (idRol, nombre) VALUES (1,'Lectura'),(2,'Edicion'),(3,'Administracion') ON DUPLICATE KEY UPDATE nombre=VALUES(n...	3 row(s) affected, 1 warning(s): 1287 'VALUES function' is d
2	19:43:56	INSERT INTO Usuario (username, nombre, email, activo) VALUES ('tecnico1','Técnico Soporte','tecnico1@org.local', TRUE), ('admin1','Je...	2 row(s) affected, 2 warning(s): 1287 'VALUES function' is d
3	19:43:56	INSERT INTO UsuarioRol (idUsuario, idRol) SELECT u.idUsuario, r.idRol FROM Usuario u JOIN Rol r WHERE (u.username='tecnico1' AND r...	2 row(s) affected, 1 warning(s): 1287 'VALUES function' is d
4	19:43:56	INSERT INTO Dependencia (nombre, sigla, ubicacion, estado) VALUES ('Mesa de Entradas','ME','PB', 'Activa'), ('Recursos Humanos','RR...	3 row(s) affected, 3 warning(s): 1287 'VALUES function' is d
5	19:43:56	INSERT INTO Equipo (idDependencia, tipo, marca, modelo, nroSerie, estado, observaciones) VALUES ((SELECT idDependencia FROM De...	3 row(s) affected, 2 warning(s): 1287 'VALUES function' is d
6	19:43:56	INSERT INTO Reparacion (idEquipo, idTecnico, fechaApertura, descripcionFalla, estado, fechaCierre, observaciones) VALUES ((SELECT i...	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0

3.6. Consultas SQL para reportes (muestras)

Se ejecutaron consultas orientadas a los reportes del sistema, coherentes con los requerimientos funcionales:

- Listado de **stock por dependencia** (filtros por tipo y estado).
- **Reparaciones** dentro de un período, con estado actual y técnico responsable.
- **Totales** de equipos por tipo y de reparaciones por estado (agregaciones).
  Las consultas se muestran como capturas y se interpretan con breves descripciones de resultado esperado (totales, conteos y ejemplos de filas).

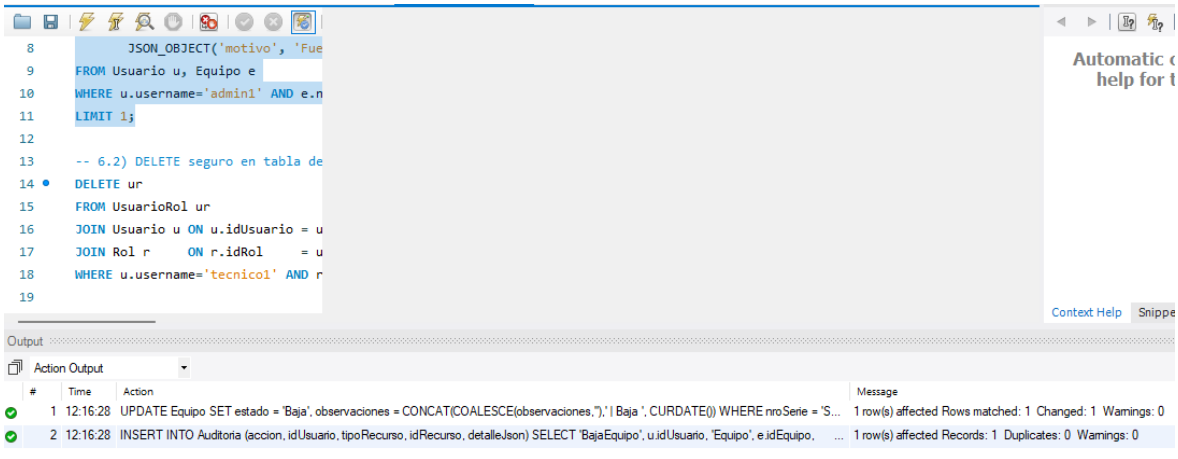


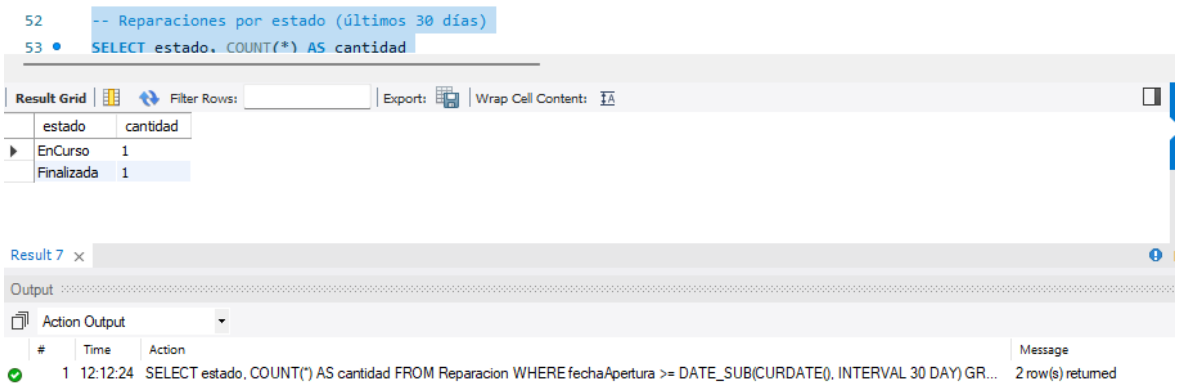


3.7. Resultados y verificación mínima

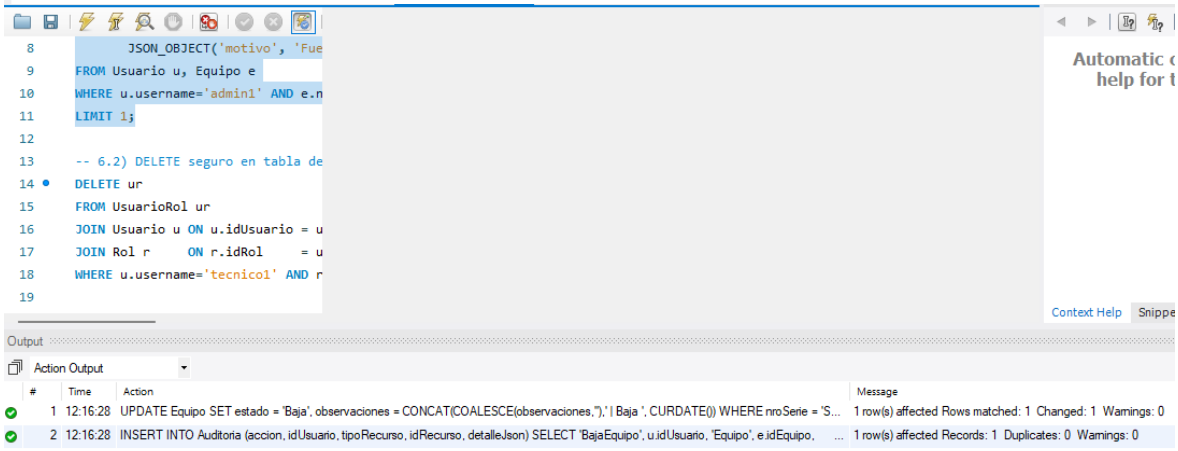
Las pruebas ejecutadas permitieron confirmar:

- La creación correcta de las tablas y relaciones definidas en el diseño.
- La consistencia de claves primarias y foráneas en operaciones de alta y actualización.
- La obtención de listados y métricas básicas para **stock** y **reparaciones**, acordes a los casos de uso prioritarios.
- La baja y eliminación de registros
  Se considera que la base de datos se encuentra lista para su integración con la capa de aplicación en etapas siguientes.


JSON_OBJECT('motivo', 'Fue
FROM Usuario u, Equipo e
WHERE u.username='admin1' AND e.n
LIMIT 1;
-- 6.2) DELETE seguro en tabla de
DELETE ur
FROM UsuarioRol ur
JOIN Usuario u ON u.idUsuario = u
JOIN Rol r ON r.idRol = u
WHERE u.username='tecnico1' AND r
Automatic c
help for t
Context Help
Snippe
Output
Action Output
Time Action Message
1 12:16:28 UPDATE Equipo SET estado = 'Baja', observaciones = CONCAT(COALESCE(observaciones,''),' | Baja ', CURDATE()) WHERE nroSerie = 'S... 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
2 12:16:28 INSERT INTO Auditoria (accion, idUsuario, tipoRecurso, idRecurso, detalleJson) SELECT 'BajaEquipo', u.idUsuario, 'Equipo', e.idEquipo, ... 1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0


-- Reparaciones por estado (últimos 30 días)
SELECT estado, COUNT(*) AS cantidad
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
estado cantidad
EnCurso 1
Finalizada 1
Result 7
Output
Action Output
Time Action Message
1 12:12:24 SELECT estado, COUNT(*) AS cantidad FROM Reparacion WHERE fechaApertura >= DATE_SUB(CURDATE(), INTERVAL 30 DAY) GR... 2 row(s) returned


-- 6.2) DELETE seguro en tabla de
DELETE ur
FROM UsuarioRol ur
JOIN Usuario u ON u.idUsuario = u
JOIN Rol r ON r.idRol = u
WHERE u.username='tecnico1' AND r
Output
Action Output
Time Action Message
1 12:16:57 DELETE ur FROM UsuarioRol ur JOIN Usuario u ON u.idUsuario = ur.idUsuario JOIN Rol r ON r.idRol = ur.idRol WHERE u.username='t... 1 row(s) affected


JSON_OBJECT('motivo', 'Fue
FROM Usuario u, Equipo e
WHERE u.username='admin1' AND e.n
LIMIT 1;
-- 6.2) DELETE seguro en tabla de
DELETE ur
FROM UsuarioRol ur
JOIN Usuario u ON u.idUsuario = u
JOIN Rol r ON r.idRol = u
WHERE u.username='tecnico1' AND r
Automatic c
help for t
Context Help
Snippe
Output
Action Output
Time Action Message
1 12:16:28 UPDATE Equipo SET estado = 'Baja', observaciones = CONCAT(COALESCE(observaciones,''),' | Baja ', CURDATE()) WHERE nroSerie = 'S... 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
2 12:16:28 INSERT INTO Auditoria (accion, idUsuario, tipoRecurso, idRecurso, detalleJson) SELECT 'BajaEquipo', u.idUsuario, 'Equipo', e.idEquipo, ... 1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0

Etapa de desarrollo – Prototipo operacional en Java

4.1 Introducción

En esta etapa se presenta el desarrollo del **prototipo funcional del sistema**, implementado en lenguaje **Java**, como continuación de la etapa de implementación del TP2. El objetivo fue demostrar la viabilidad técnica del modelo propuesto y aplicar los principios de la **Programación Orientada a Objetos (POO)**, garantizando la coherencia con el diseño lógico y físico del sistema.

4.2 Arquitectura general

El sistema se organiza en una arquitectura **por capas**, que facilita la separación de responsabilidades y la evolución futura del proyecto:

- **Capa de presentación:** interfaz gráfica desarrollada con Java Swing, compuesta por formularios (como *EquipoForm*, *ReparacionForm* y *AuditPanel*) y controladores que gestionan la interacción con el usuario.
- **Capa de negocio:** servicios (*EquipoService*, *ReparacionService*, *DependenciaService*, *AuditService*) encargados de aplicar las reglas funcionales, validar datos y coordinar operaciones entre la interfaz y los repositorios.
- **Capa de acceso a datos:** interfaces de repositorio (*EquipoRepository*, *ReparacionRepository*, *DependenciaRepository*) e implementaciones *InMemory*, que simulan la persistencia de datos y abstraen el origen del almacenamiento.

Esta estructura promueve **modularidad**, **bajo acoplamiento** y **facilidad de mantenimiento**, posibilitando la futura integración con MySQL sin modificar la lógica del negocio.

4.3 Aplicación de los principios de POO

Encapsulamiento

El **encapsulamiento** se aplica en todas las clases del dominio mediante el uso de atributos *privados* y métodos *públicos* de acceso (*getters/setters*), garantizando la integridad del estado interno de los objetos.

Por ejemplo, las clases *Equipo*, *Reparacion* y *Dependencia* definen sus atributos principales como privados y sólo permiten su modificación a través de métodos controlados. Esto impide alteraciones directas desde otras capas del sistema y centraliza la validación de datos.

Además, la capa de servicios refuerza el encapsulamiento al gestionar todas las operaciones críticas (altas, bajas, modificaciones) mediante métodos que validan las reglas de negocio antes de afectar los datos del dominio.

Abstracción

La **abstracción** se refleja en la definición de interfaces que ocultan los detalles de implementación y exponen únicamente las operaciones esenciales. Las interfaces *EquipoRepository*, *ReparacionRepository* y *DependenciaRepository* especifican qué acciones se pueden realizar (guardar, buscar, eliminar, listar) sin definir cómo se llevan a cabo.

Esto permite reemplazar una implementación *InMemory* por otra basada en MySQL sin alterar las demás capas, cumpliendo el principio de *programar contra una interfaz, no contra una implementación*.

Herencia

La **herencia** se utiliza principalmente en la capa de presentación, donde las clases de interfaz heredan componentes de *Swing* para extender su funcionalidad. Por ejemplo, *EquipoForm* y

ReparacionForm heredan de *JPanel* y agregan sus propios controles, campos y validaciones específicas.
Asimismo, las enumeraciones (*EstadoEquipo*, *EstadoReparacion*, *TipoEquipo*) establecen jerarquías conceptuales reutilizables que simplifican el mantenimiento y amplían la expresividad del código.

Polimorfismo

El **polimorfismo** permite que distintas implementaciones de una misma interfaz sean utilizadas de manera intercambiable. Por ejemplo, *EquipoService* puede funcionar indistintamente con un *InMemoryEquipoRepository* o un *JdbcEquipoRepository* sin que cambie su comportamiento externo. Esto garantiza flexibilidad y escalabilidad en la arquitectura.

4.4 Estructuras de control y manejo de excepciones

El sistema emplea estructuras condicionales y repetitivas (*if*, *for*, *while*) para validar entradas, recorrer colecciones y filtrar resultados. Estas estructuras se aplican, por ejemplo, al verificar la unicidad del número de serie de un equipo o al generar listados ordenados mediante *Comparator*.
El **manejo de excepciones** se implementa en la capa de servicios mediante *IllegalArgumentException* y se propaga a la interfaz gráfica, donde se capturan los errores con *try/catch* y se informan al usuario mediante cuadros de diálogo (*JOptionPane*). Este enfoque asegura robustez y mejora la experiencia del usuario final.

4.5 Interfaz y menú de navegación

La interfaz gráfica se organiza en pestañas (*JTabbedPane*) que agrupan los principales módulos: gestión de equipos, gestión de reparaciones y auditoría. Esta estructura cumple la función de menú de selección, proporcionando una navegación clara e intuitiva.
Adicionalmente, el diseño contempla la futura incorporación de una barra de menús (*JMenuBar*) para optimizar la usabilidad y facilitar el acceso a funciones como exportar datos o generar reportes.

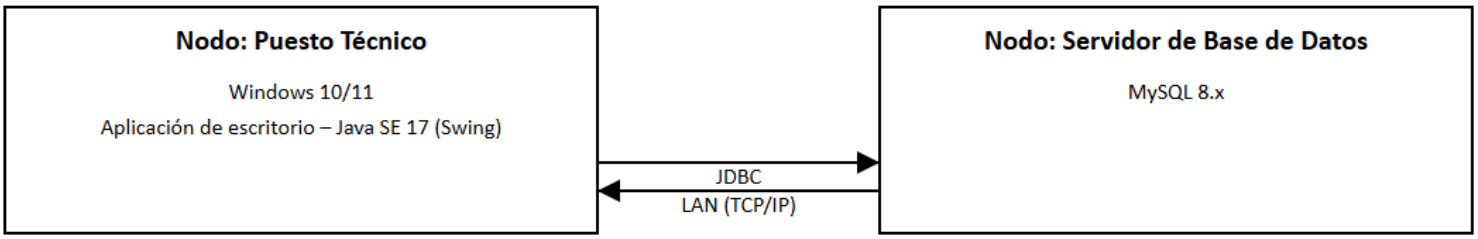
4.6 Validación funcional del prototipo

Las pruebas realizadas sobre el prototipo demostraron el funcionamiento correcto de las operaciones clave:

- Registro de equipos nuevos y validación de duplicados.
- Asociación de equipos a dependencias existentes.
- Registro y cierre de reparaciones, con actualización del estado del equipo.
- Exportación de listados a formato CSV.
- Registro de eventos en el módulo de auditoría.

En todos los casos, los resultados fueron coherentes con los requerimientos definidos en el TP2 y confirmaron la correcta interacción entre las capas del sistema.

Diagrama de despliegue



Arquitectura y patrones de diseño

Patrones aplicados.

- **MVC (Model–View–Controller):** la UI Swing (formularios y tablas) muestra/captura datos; los **controladores** gestionan eventos y coordinan con **servicios**; el **modelo** son las entidades de dominio. Beneficio: separación clara de responsabilidades y facilidad de mantenimiento.
- **Repository con interfaces:** contratos en domain.repo.* e implementaciones intercambiables en infra.repo.mem.* (memoria) y infra.repo.mysql.* (MySQL). Beneficio: bajo acoplamiento con la tecnología de persistencia; la lógica de negocio permanece estable al alternar el almacenamiento.

Justificación de los patrones de diseño

Por qué MVC (Model–View–Controller).

- **Separación de responsabilidades:** la UI Swing solo presenta y captura datos (Vista), los controladores orquestan eventos y validaciones livianas (Controlador) y el modelo concentra el estado del negocio (Modelo). Esto reduce acoplamiento entre interfaz y lógica.
- **Mantenibilidad y evolución:** permite cambiar formularios/tablas o agregar nuevas pantallas sin tocar la lógica de negocio; al mismo tiempo, las reglas pueden evolucionar sin reescribir la UI.
- **Testabilidad:** la lógica queda en servicios/modelo, facilitando pruebas sin necesidad de levantar la UI.
- **Alineación con el alcance:** al tratarse de un prototipo operacional con varias pantallas (equipos, reparaciones, auditoría), MVC aporta claridad y evita “lógica en la vista”.

Por qué Repository con interfaces.

- **Bajo acoplamiento con la persistencia:** los servicios dependen de **interfaces** (domain.repo.*), no de una tecnología concreta. Así, la aplicación funciona igual con repos en memoria o con MySQL.
- **Intercambiabilidad y pruebas:** se puede alternar fácilmente entre infra.repo.mem.* (para pruebas/demos) y infra.repo.mysql.* (persistencia real) sin cambiar la capa de negocio ni la UI.
- **Encapsulamiento del mapeo y SQL:** consultas, PreparedStatement y decisiones de mapeo (p. ej., Equipo.codigo ↔ observaciones) quedan aisladas en los repositorios MySQL, manteniendo limpia la lógica de dominio.

- **Escalabilidad del diseño:** si mañana se migra a ORM/JPA o a otra fuente (archivos/servicios), basta con otra implementación de las mismas interfaces.

Implementación de persistencia en MySQL

- **Conexión JDBC** centralizada en infra.repo.mysql.MySqlConnectionProvider a la base **inventario_it**.
- **Repositorios implementados:** MySqlDependenciaRepository, MySqlEquipoRepository, MySqlReparacionRepository, con **CRUD** mediante PreparedStatement.
- **Mapeos clave** (compatibles con el DDL previo):
  - Equipo.codigo ↔ columna Equipo.observaciones.
  - Enums de dominio ↔ enums SQL (TipoEquipo, EstadoEquipo, EstadoReparacion).
- **Datos de prueba ampliados:** cada dependencia tiene **2–6 PCs** y **1–3 impresoras** para validar filtros y búsquedas.

Manejo de excepciones (BD y UI)

- **Capa de datos:** captura de SQLException y de integridad referencial (SQLIntegrityConstraintViolationException), traducidas a **mensajes de negocio** (p. ej., al intentar eliminar equipos con reparaciones asociadas).
- **UI/Controladores:** try/catch y presentación de errores con JOptionPane; la aplicación no se interrumpe ante errores esperables.

Estructuras de datos utilizadas

- **Arreglos** (String[]): encabezados fijos en modelos de tabla.
- **ArrayList/List:** datasets dinámicos (filas) y listas de entidades para búsqueda/filtrado.
  Uso complementario: estructura fija + colección mutable para recarga y filtrado eficiente.

Empleo de archivos

- **Exportación CSV** (util.CsvExporter) en **UTF-8 con BOM** para compatibilidad con Excel (tildes/ñ correctos).
- **Auditoría persistente** (service.AuditService) en logs/audit-log.csv (se conserva entre ejecuciones).
- **Preferencia de auditoría** (util.AuditConfig) en config/audit-mode.txt (modo básico/detallado).