



Universidad Veracruzana

Facultad de Negocios y Tecnologías



## **ReparaYa**

Karla Sofía Castro Pérez

Emma Hernández Mendoza

Facultad de Negocios y tecnologías

Universidad Veracruzana

Requisitos de Ingeniería de Software

María Dolores González Martínez

09/12/2025

## Contenido

<b>2. Introducción .....</b>	<b>1</b>
<b>3. Descripción general del sistema .....</b>	<b>1</b>
<b>4. Problema que resuelve .....</b>	<b>1</b>
<b>5. Usuarios principales.....</b>	<b>3</b>
<b>6. Objetivos del sistema .....</b>	<b>4</b>
<b>7. Alcance del sistema .....</b>	<b>4</b>
<b>8. Requisitos Funcionales .....</b>	<b>6</b>
<b>9. Requisitos No Funcionales .....</b>	<b>11</b>
<b>10. Stack Tecnológico Utilizado .....</b>	<b>13</b>
<b>11. Modelo del Sistema.....</b>	<b>14</b>
<b>12. Funcionalidad en tiempo real.....</b>	<b>40</b>
<b>13. Trabajo a Futuro .....</b>	<b>41</b>
<b>14. Conclusiones .....</b>	<b>42</b>
<b>15. Equipo de Trabajo .....</b>	<b>43</b>

## 2. Introducción

El proyecto **ReparaYA** consiste en el desarrollo de un **Sistema de Gestión de Soporte Técnico** con una arquitectura web robusta, diseñado para optimizar las operaciones de un taller de reparación de dispositivos electrónicos. El sistema abarca la gestión del ciclo de vida de las órdenes de servicio, el control de inventario de piezas y la administración de la información de clientes. El *backend* se sustenta en **Java con Spring Boot** y la persistencia de datos utiliza **PostgreSQL**.

## 3. Descripción general del sistema

- **¿Qué es el sistema?** Es un Sistema de Gestión de Soporte Técnico diseñado para optimizar las órdenes de reparación, inventario y gestión de clientes.
- **Componentes principales:** Gestión de usuarios y roles; Clientes y Servicios; Inventario y Gastos, y Reportes.
- **Tipo de arquitectura:** Aplicación **Web** robusta, sustentada en el patrón **MVC (Modelo-Vista-Controlador)**.

## 4. Problema que resuelve

El sistema **ReparaYA** se concibe como la solución centralizada para los problemas de **control, trazabilidad y gestión financiera** que limitan la rentabilidad y la calidad del servicio en el taller de reparación de dispositivos electrónicos.

### Descripción del problema actual

El taller opera bajo procesos de gestión insuficientes o manuales, generando un ambiente de **caos operativo y desconfianza** en las transacciones con los clientes. Los principales problemas son:

- **Falta de trazabilidad del dispositivo:** No existe un registro exacto y estandarizado del estado de un equipo al momento de la recepción, ni de la razón original de la falla. Esto dificulta la defensa contra reclamos posteriores del cliente.

- **Caos operativo e inventario desorganizado:** La falta de un sistema centralizado lleva a una gestión desordenada de los equipos y al desconocimiento del *stock* disponible, lo cual impacta directamente en los tiempos de reparación.
- **Desconocimiento financiero:** La ausencia de un registro formal y centralizado de los ingresos y gastos impide obtener una orientación clara sobre la rentabilidad real del negocio.

## Evidencia

La problemática fue identificada y validada a través de la siguiente evidencia cualitativa recopilada durante la fase de Análisis de Necesidades:

Instrumento	Hallazgos Recopilados (Evidencia)	Módulo que Resuelve
Entrevista Directa	"La gente me dice mentiras acerca de cómo se entregó su equipo [y] sacan un problema extra del que originalmente era."	<b>Órdenes de servicio:</b> Registro detallado del problema inicial.
Observación de Procesos	"Tengo un desastre respecto a los equipos" y "No tengo un registro exacto de por qué llegó un teléfono."	<b>Gestión de dispositivos:</b> Registro organizado y único por dispositivo.
Entrevista Directa	"No tengo una orientación de mis ganancias y gastos."	<b>Reportes:</b> Módulo de Ingresos y Gastos.
Entrevista Directa	"No anoto las contraseñas para hacer sus respectivas pruebas después."	<b>Gestión de clientes:</b> Campos opcionales para información crítica de prueba (manejada de forma segura).

## Impacto de No resolver la problemática

La persistencia de estas fallas operativas y de control tiene un impacto directo y negativo en la reputación y la rentabilidad del negocio:

- **Pérdida de credibilidad y tiempo:** La falta de registros precisos del estado inicial de un dispositivo abre la puerta a reclamos inválidos ("sacar un problema extra"), obligando al técnico a invertir tiempo extra en disputas o reparaciones no cobradas.
- **Decisiones financieras incorrectas:** Al no tener una orientación clara de **ganancias y gastos**, el administrador no puede establecer precios, evaluar la rentabilidad de los servicios, ni tomar decisiones informadas sobre la compra de inventario.
- **Retrasos operacionales:** El desorden en los equipos y la falta de información inmediata sobre las contraseñas o las fallas originales provocan fricciones y **retrasos** innecesarios en el tiempo de entrega prometido al cliente.

## 5. Usuarios principales

Esta sección se revisa para reflejar que, si bien el sistema está diseñado para múltiples roles, solo el rol de Administrador es funcional en el MVP.

Tipo de usuario	Perfil y necesidades clave (Rol)
-----------------	-------------------------------------

<b>Administrador</b> (Rol Operativo MVP)	Es el único rol implementado en esta versión. <b>El Administrador tiene acceso a la totalidad de las funcionalidades del sistema (gestión de clientes, órdenes de servicio, inventario, reportes) y, por lo tanto, asume las responsabilidades operacionales de los roles de Recepcionista y Técnico</b> en esta etapa inicial del proyecto.
<b>Recepcionista</b>	<i>(Rol Conceptual para Fases Futuras)</i> Diseñado para la captura inicial de servicios, gestión de la información de clientes, y la emisión de recibos.

<b>Técnico</b>	( <i>Rol Conceptual para Fases Futuras</i> ) Diseñado para la actualización del estado de reparación, el registro de procedimientos realizados, y la asignación de piezas utilizadas.
----------------	---

## 6. Objetivos del sistema

### Objetivos Generales y Específicos

#### *Objetivo General*

Implementar un sistema de gestión integral que centralice las operaciones de un taller de soporte técnico, mejorando la eficiencia operativa, el control de inventario y la trazabilidad de cada orden de reparación.

#### *Objetivos Específicos*

- **Garantizar la trazabilidad:** Crear un sistema que permita generar órdenes de servicio con número único y mantener un historial auditado de los cambios de estado y procedimientos realizados en cada reparación.
- **Optimizar el inventario:** Controlar los niveles de *stock* de piezas y accesorios en tiempo real, con la capacidad de generar alertas cuando el *stock* sea inferior al mínimo configurado.
- **Implementar control de acceso:** Desarrollar un sistema de autenticación seguro que permita el acceso con usuario y contraseña, e implementar **múltiples roles** (Administrador, Técnico, Recepcionista) para aplicar la seguridad basada en perfiles.

## 7. Alcance del sistema

El alcance de la versión inicial (MVP) del sistema **ReparaYA** se enfoca en las funcionalidades categorizadas como **MUST HAVE** y **SHOULD HAVE**.

El proyecto **cubre** los siguientes módulos principales en esta versión:

- **Gestión de usuarios y roles (Administrador):** El sistema permite la gestión de usuarios y roles. **Nota:** En esta versión, la funcionalidad completa y el flujo de trabajo están diseñados y probados únicamente bajo el rol de **Administrador**, posponiendo la

implementación de las interfaces y restricciones específicas para los roles de Recepcionista y Técnico para el trabajo a futuro.

- **Gestión de clientes:** Incluye el CRUD completo de clientes y el registro de especificaciones de los servicios asociados.
- **Gestión de órdenes de servicio y estados:** Permite manejar los estados clave del flujo de reparación (Pendiente, En proceso, Listo, Entregado).
- **Inventario básico:** Implementa el CRUD de piezas, el control de *stock* y las alertas de *stock* bajo.
- **Reportes básicos:** Genera reportes iniciales sobre el inventario y las reparaciones pendientes.

El proyecto **NO incluye** funcionalidades avanzadas como la integración con sistemas contables externos, una aplicación móvil para clientes, sistemas de pagos en línea o geolocalización de tiendas.

### **Justificación de la importancia**

La implementación de **ReparaYA** se justifica por la necesidad de migrar de procesos manuales o dispersos a un sistema centralizado y digital. Esto permite:

- **Reducir errores operacionales:** Al automatizar el control de *stock* y la asignación de piezas a las reparaciones, se previenen pérdidas por *stock* insuficiente o errores de cálculo de costos.
- **Mejorar la atención al cliente:** El registro detallado de clientes, dispositivos, y el historial de reparaciones, junto con la gestión de estados en tiempo real, permite ofrecer un servicio más transparente y eficiente.
- **Asegurar la escalabilidad:** El sistema está diseñado con una arquitectura modular para soportar la expansión del negocio a **tres tiendas simultáneas** en el futuro.

## 8. Requisitos Funcionales

### Requisitos Funcionales (RF)

Los RF definen las funciones que el sistema debe ejecutar. Los requisitos críticos (MUST HAVE) para el MVP incluyen:

#### MUST HAVE

- **RF01 - Autenticación y control de acceso:** Login con usuario/contraseña e implementación de roles (Administrador, Técnico, Recepcionista).
- **RF02 - Gestión de clientes – CRUD de clientes.**
  - Campos obligatorios: nombre, teléfono, email, dirección, INE
  - Debe incluir búsqueda de clientes por nombre, teléfono o email
- **RF03 - Gestión de servicios – CRUD de servicios.**
  - Debe permitir registrar dispositivos asociados a clientes
  - Campos obligatorios: tipo, marca, modelo, propietario
  - Campos opcionales: número de serie, color, especificaciones técnicas
  - Debe permitir consultar historial de reparaciones por dispositivo
  - Generación de órdenes únicas, captura del problema, y cálculo del tiempo de entrega estimado.
  - Debe asignar estado inicial "Pendiente" a nuevas órdenes
  - Debe calcular y mostrar tiempo de entrega estimado
- **RF04 - Gestión de estados de reparación**
  - Estados disponibles: Pendiente, En Proceso, Listo, Entregado
  - Solo técnicos pueden actualizar estados de reparación
  - Debe mantener historial auditado de cambios de estado
  - Debe registrar fecha/hora de cada transición de estado



- **RF05 - Gestión de Inventario básico:** CRUD de piezas, control de stock y alertas cuando el stock sea inferior al mínimo.
- **RF06 - Asignación de piezas a reparaciones**
  - Debe registrar piezas utilizadas en cada reparación
  - Debe descontar automáticamente del inventario
  - Debe calcular costo total de piezas por reparación
  - Debe prevenir uso de piezas con stock insuficiente
- **RF07 – Gestión de problemas comunes (Catálogo)**
  - Debe permitir agregar, editar y eliminar problemas comunes (ej: "Pantalla rota", "Batería no carga").
  - Debe mostrar lista de problemas disponibles para selección rápida en órdenes de servicio.
  - Debe permitir asociar problemas a categorías de servicio.
- **RF08 – Gestión de categorías de servicios (CRUD)**
  - Debe permitir crear, editar y eliminar categorías de servicio.
  - Campos obligatorios: Nombre de categoría, Precio base, Tiempo estimado.
  - Debe mostrarse en tabla con opciones de edición.
  - Debe usarse al generar órdenes de servicio para pre-cálculo de costos.
- **RF09 – Gestión de estados de servicio**
  - Debe permitir activar/desactivar estados predefinidos.
  - Estados base: Pendiente, En diagnóstico, En espera de piezas, En reparación, Listo, Entregado.
  - Debe permitir agregar estados personalizados (opcional).
- **RF10 – Gestión de marcas de productos**
  - Debe permitir agregar, editar y eliminar marcas (ej: Apple, Samsung, Dell).

- Debe usarse al registrar un dispositivo para selección desde lista desplegable.
- Debe evitar duplicados.
- **RF11 – Gestión de tipos de dispositivos**
  - Debe permitir agregar, editar y eliminar tipos (ej: Celular, Laptop, Tablet, Smartwatch).
  - Debe usarse en registro de dispositivos para clasificación.
- **RF12 – Gestión de productos en inventario:**
  - Debe permitir agregar un nuevo producto con: Nombre del producto, Categoría (Pantallas, Accesorios, Herramientas, etc.), Descripción (opcional), Precio costo\*, Precio venta\*, Stock actual\*, Stock mínimo\* (para alertas), Imagen del producto (opcional, hasta 5MB, formatos PNG/JPG)
  - Debe calcular automáticamente:
    - $\text{Margen de ganancia} = (\text{Precio venta} - \text{Precio costo})$
    - $\text{Valor del inventario} = \text{Sumatoria} (\text{Precio costo} \times \text{Stock actual})$
  - Debe permitir buscar productos por nombre o categoría.
  - Debe mostrar estado del producto (Normal, Bajo stock, etc.).

## **SHOULD HAVE**

- **RF13 - Sistema de búsqueda avanzada**
  - Debe permitir búsqueda unificada de clientes y dispositivos
  - Debe incluir filtros por múltiples criterios
  - Debe mostrar resultados en tiempo real durante la escritura
- **RF14 - Gestión de imágenes**
  - Debe permitir capturar y almacenar imágenes del dispositivo
  - Límite de 5 imágenes por dispositivo

- Debe permitir visualización de imágenes en registros
- Debe soportar formatos comunes (JPG, PNG)
- **RF15 – Seguimiento de procedimientos**
  - Debe permitir registrar procedimientos técnicos realizados
  - Debe incluir descripción detallada del trabajo ejecutado
  - Debe permitir adjuntar observaciones técnicas
- **RF16 - Sistema de notificaciones**
  - Debe generar alertas por reparaciones atrasadas
  - Debe notificar sobre piezas con stock bajo
  - Debe mostrar recordatorios de entregas pendientes
- **RF17 – Gestión de gastos:**
  - Debe permitir registrar un nuevo gasto con los siguientes campos obligatorios: Fecha. Categoría (Inventario, Herramientas, Servicios, etc.), Descripción, Monto, Comprobante (opcional, ej: FAC-001)
  - Debe mostrar un historial de gastos ordenado por fecha descendente.
  - Debe permitir editar un gasto existente.
  - Debe permitir filtrar gastos por categoría
  - Debe calcular y mostrar el total de gastos por período.

#### **COULD HAVE**

- **RF18 - Sistema de facturación**
  - Debe generar facturas detalladas con IVA
  - Debe imprimir recibos de entrega
  - Debe calcular totales (mano de obra + piezas)
- **RF19 - Reportes avanzados**

- Debe generar reportes de productividad por técnico
- Debe mostrar estadísticas de tiempos de reparación
- Debe reportar ganancias por período
- Debe identificar piezas más utilizadas
- **RF20 - Gestión de herramientas**
  - Debe registrar herramientas utilizadas en reparaciones
  - Debe controlar estado y disponibilidad de herramientas
  - Debe generar alertas de mantenimiento de herramientas
- **RF21 - Sistema de backup**
  - Debe permitir configuración de backups automáticos
  - Debe programar copias de seguridad regulares
  - Debe permitir restauración de datos desde backup

### **WON'T HAVE (Funcionalidades Excluidas)**

Estos requisitos **funcionales** están **excluidos** del alcance de esta versión del sistema:

- **RF22 - Aplicación móvil para clientes**
- **RF23 - Sistema de pagos en línea**
- **RF24 - Geolocalización de tiendas**
- **RF25 - Chat interno en tiempo real**
- **RF26 - Integración con redes sociales**
- **RF27 - Sistema de garantías automatizado**
- **RF28 - App para técnicos externos**

**Nota Operacional del MVP:** En la versión inicial de ReparaYA, la ejecución de todos los requisitos funcionales listados (RF01 a RF14), incluyendo aquellos conceptualmente asignados

a los roles de Recepcionista (ej. RF02) y Técnico (ej. RF04, RF06), es realizada exclusivamente por el **Administrador** debido a las limitaciones de alcance en la fase de implementación.

## 9. Requisitos No Funcionales

### ***Requisitos No Funcionales (RNF)***

Los RNF definen las cualidades del sistema. Los requisitos críticos (MUST HAVE) son26:

#### **MUST HAVE**

- **RNF01 - Rendimiento:** Tiempo de respuesta menor a 1 segundo para el 95% de las transacciones.
- **RNF02 - Seguridad:** Contraseñas almacenadas con hash bcrypt y sesiones con timeout automático de 8 horas.
- **RNF003 - Disponibilidad**
  - Sistema disponible durante horario comercial (8:00-20:00)
  - Tiempo de inactividad máximo del 1% mensual
  - Recuperación ante fallos en menos de 30 minutos
- **RNF04 - Usabilidad**
  - Interfaz intuitiva con curva de aprendizaje < 1 día
  - Navegación consistente entre módulos
  - Mensajes de error claros y orientativos

#### **SHOULD HAVE**

- **RNF05 - Escalabilidad**
  - Capacidad de soportar hasta 3 tiendas simultáneas
  - Arquitectura modular para expansión futura
  - Base de datos normalizada para múltiples ubicaciones
- **RNF06 - Mantenibilidad**

- Código documentado y comentado
- Estructura modular con separación de responsabilidades
- Logs detallados para diagnóstico de problemas
- **RNF07 - Compatibilidad**
  - Soporte para navegadores modernos (Chrome, Firefox, Edge)
  - Responsive design para tablets y pantallas de 10"+
  - Compatible con Windows 10/11 y servidores Windows/Linux

#### **COULD HAVE**

- **RNF08 - Integración**
  - API REST para integración con otros sistemas
  - Exportación de datos en formatos CSV y PDF
  - Posibilidad de integración con sistemas contables
- **RNF09 - Internacionalización**
  - Soporte para múltiples idiomas (Español, Inglés)
  - Formato de fechas y moneda configurable
  - Adaptación a regulaciones locales
- **RNF010 - Auditoría**
  - Logs completos de todas las operaciones del sistema
  - Trazabilidad de cambios en datos críticos
  - Reportes de actividad por usuario

#### **WON'T HAVE (Cualidades Excluidas)**

Este requisito **no funcional** está **excluido** del alcance de esta versión del sistema (ya que la integración con sistemas contables se considera un atributo de integración, es decir, no funcional):

- **RNF011 - Integración con sistemas contables externos** (Aunque existe en el **COULD HAVE** como potencial, se confirma su exclusión de esta versión).

## 10. Stack Tecnológico Utilizado

El sistema **ReparaYA** se desarrolló bajo un *stack* de tecnologías robusto, moderno y optimizado para el patrón MVC (Modelo-Vista-Controlador), tal como se describe en la arquitectura de software.

Componente	Tecnología	Propósito	Justificación en el proyecto
<b>Backend / Lógica</b>	<b>Java</b>	Lenguaje principal para la lógica de negocio.	Se eligió por ser robusto, seguro y manejar bien la lógica de negocio compleja <sup>2</sup> .
<b>Framework</b>	<b>Spring Boot</b>	Simplifica la configuración y permite crear un servidor web con facilidad.	Facilita la implementación de la arquitectura MVC <sup>3</sup> .
<b>Persistencia / BD</b>	<b>PostgreSQL</b>	Sistema de gestión de bases de datos relacional (SGBDR) de código abierto.	Elegido por ser robusto, confiable y para asegurar la integridad de datos críticos <sup>4</sup> .
<b>Frontend / Motor</b>	<b>Thymeleaf</b>	Motor de plantillas que inyecta datos de Java en archivos HTML.	Se integra nativamente con Spring Boot y facilita la creación de vistas dinámicas <sup>5</sup> .

<b>Frontend / Estructura</b>	<b>HTML5</b>	Estructura el contenido del sistema (el esqueleto) <sup>6</sup> .	Estándar web para definir la estructura de la interfaz.
<b>Frontend / Estilo</b>	<b>CSS3</b>	Da el estilo visual (la piel) <sup>7</sup> .	Utilizado para replicar el diseño exacto de la interfaz <sup>8</sup> .
<b>Frontend / Interactividad</b>	<b>JavaScript (Vanilla)</b>	Permite la interactividad simple (abrir modales, pestañas) <sup>9</sup> .	Se usó sin <i>frameworks</i> pesados (como React o Angular) para mantener la ligereza <sup>10</sup> .

## 11. Modelo del Sistema

### Casos de Uso (UC)

Los casos de uso definen las interacciones funcionales entre los actores y el sistema. Los actores principales son:

- **Recepcionista (UC001-UC006/UC014):** Registra nuevos clientes y dispositivos, genera órdenes de servicio, y busca registros.
- **Técnico (UC007-UC009):** Actualiza el estado de reparación, registra procedimientos y asigna piezas utilizadas, que a su vez incluye la acción de Actualizar stock.
- **Administrador (UC010-UC01):** Gestiona usuarios e inventario, genera reportes y configura el backup.



ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Login				UC001
ACTORES	Recepcionista, Técnico, Administrador				
TIPO	Primario				
REFERENCIAS	RF01				
PRECONDICIÓN	El usuario debe estar registrado en el sistema.				
POSTCONDICIÓN	El usuario accede al sistema con permisos según su rol.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Permitir a los usuarios autenticarse en el sistema.					
RESUMEN					
El usuario ingresa sus credenciales y el sistema valida su identidad y rol.					
CURSO NORMAL					
1	Usuario: ingresa usuario y contraseña.	2	Sistema valida credenciales y rol.		
		3	Sistema redirige según rol.		
CURSOS ALTERNOS					
1a	Credenciales incorrectas → mensaje de error.				
OTROS DATOS					
FRECUENCIA ESPERADA	Alta (diaria)	RENDIMIENTO	< 1 segundo		
IMPORTANCIA	Vital	URGENCIA	Alta		
ESTADO	Terminado	ESTABILIDAD	Alta		

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Registrar nuevo cliente				UC002
ACTORES	Recepcionista				
TIPO	Primario				
REFERENCIAS	RF02		CU001		
PRECONDICIÓN	El usuario debe haber iniciado sesión como Recepcionista.				
POSTCONDICIÓN	Se crea un nuevo registro de cliente en el sistema.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Capturar información de un cliente nuevo.					
RESUMEN					
El recepcionista ingresa los datos obligatorios del cliente y guarda el registro.					
CURSO NORMAL					
1	Recepcionista selecciona “Nuevo cliente”.		3	Sistema guarda y asigna ID único.	
2	Ingresa información del nuevo cliente (Nombre completo, teléfono, email, dirección, INE)				
CURSOS ALTERNOS					
2a	Datos incompletos → sistema solicita completar campos obligatorios.				
OTROS DATOS					
FRECUENCIA ESPERADA		Moderada (varios por día)	RENDIMIENTO		< 2 segundos
IMPORTANCIA		Alta	URGENCIA		Alta
ESTADO		Terminado	ESTABILIDAD		Alta

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Generar orden de servicio				UC003
ACTORES	Recepcionista				
TIPO	Primario				
REFERENCIAS	RF03		CU002		
PRECONDICIÓN	El cliente debe estar registrado.				
POSTCONDICIÓN	Se crea una orden de servicio con estado “Pendiente”.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Crear una orden de reparación con descripción del problema y tiempo estimado.					
RESUMEN					
Se captura el problema, se asigna un técnico y se calcula tiempo de entrega.					
CURSO NORMAL					
1	Recepcionista selecciona cliente y dispositivo3		Sistema crea nuevo servicio asociado a cliente.		
2	Ingresa descripción del problema y la fecha de ingreso del dispositivo.				
CURSOS ALTERNOS					
1a	Cliente no existe → se redirige a UC02.				
OTROS DATOS					
FRECUENCIA ESPERADA	Alta (varios por día)		RENDIMIENTO		< 2 segundos
IMPORTANCIA	Alta		URGENCIA		Alta
ESTADO	Terminado		ESTABILIDAD		Moderada

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Capturar imágenes del dispositivo				UC004
ACTORES	Recepcionista				
TIPO	Secundario				
REFERENCIAS	RF14	UC003			
PRECONDICIÓN	Servicio activo en el sistema.				
POSTCONDICIÓN	Imágenes almacenadas y asociadas al servicio.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Registrar visualmente el estado del dispositivo al ingreso.					
RESUMEN					
Recepcionista toma fotos del dispositivo y las adjunta al registro.					
CURSO NORMAL					
1	Recepcionista selecciona dispositivo.	3	Sistema valida formato y límite (5 imágenes)		
2	Activa cámara o carga imágenes.				
4	Imágenes se guardan asociadas.				
CURSOS ALTERNOS					
1a	Cliente no existe → se redirige a UC002.				
OTROS DATOS					
FRECUENCIA ESPERADA	Alta (varios por día)	RENDIMIENTO		< 2 segundos	
IMPORTANCIA	Alta	URGENCIA		Alta	
ESTADO	Terminado	ESTABILIDAD		Moderada	

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Buscar cliente/dispositivo				UC005
ACTORES	Recepcionista, Técnico, Administrador				
TIPO	Primario				
REFERENCIAS	RF13		UC001		
PRECONDICIÓN	Usuario autenticado.				
POSTCONDICIÓN	<u>Se muestran resultados de búsqueda.</u>				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Localizar rápidamente registros de clientes o dispositivos.					
RESUMEN					
Usuario ingresa criterios de búsqueda y sistema devuelve coincidencias.					
CURSO NORMAL					
1	Usuario ingresa término (nombre, teléfono, modelo).		2	Sistema busca en tiempo real	
			3	Muestra lista de resultados.	
CURSOS ALTERNOS					
2a	Sin resultados → mensaje informativo.				
OTROS DATOS					
FRECUENCIA ESPERADA	Alta		RENDIMIENTO	< 1 segundo	
IMPORTANCIA	Alta		URGENCIA	Alta	
ESTADO	Terminado		ESTABILIDAD	Alta	

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Actualizar datos de contacto				UC006
ACTORES	Recepcionista				
TIPO	Primario				
REFERENCIAS	RF02				
PRECONDICIÓN	Cliente debe existir en el sistema.				
POSTCONDICIÓN	Datos de contacto actualizados.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Mantener actualizada la información de contacto del cliente.					
RESUMEN					
Recepcionista modifica teléfono, email o dirección del cliente.					
CURSO NORMAL					
1	Recepcionista busca cliente.				
2	Selecciona "Editar datos".				
3	Modifica campos y guarda.				
CURSOS ALTERNOS					
3a	Datos inválidos (email mal escrito) → error.				
OTROS DATOS					
FRECUENCIA ESPERADA	Baja	RENDIMIENTO		< 2 segundos	
IMPORTANCIA	Moderada	URGENCIA		Moderada	
ESTADO	Terminado	ESTABILIDAD		Alta	

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Generar facturas/recibos				UC007
ACTORES	Recepcionista				
TIPO	Secundario				
REFERENCIAS	RF18				
PRECONDICIÓN	Reparación en estado "Listo" y con costos calculados.				
POSTCONDICIÓN	Factura/recibo generado e impreso.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Emitir comprobante de pago al cliente.					
RESUMEN					
Sistema genera factura con IVA y totales, permitiendo impresión.					
CURSO NORMAL					
1	Recepcionista selecciona servicio "Listo para entregar".		2	Sistema calcula total (mano de obra + piezas).	
			3	Sistema muestra el apartado "Cotización".	
			3	Genera factura/recibo y opción de impresión.	
CURSOS ALTERNOS					
OTROS DATOS					
FRECUENCIA ESPERADA	Alta		RENDIMIENTO	< 4 segundos	
IMPORTANCIA	Alta		URGENCIA	Moderada	
ESTADO	Terminado		ESTABILIDAD	Alta	

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Actualizar estado de reparación				UC008
ACTORES	Técnico				
TIPO	Primario				
REFERENCIAS	RF03, RF04		UC001		
PRECONDICIÓN	Orden de servicio creada y asignada al técnico.				
POSTCONDICIÓN	El estado de la reparación se actualiza y se registra en el historial.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Permitir al técnico cambiar el estado de una reparación.					
RESUMEN					
El técnico selecciona una orden y actualiza su estado (En Proceso, Listo, etc.).					
CURSO NORMAL					
1	Técnico selecciona orden asignada.		3	Sistema guarda cambio con fecha/hora.	
2	Cambia estado.				
CURSOS ALTERNOS					
OTROS DATOS					
FRECUENCIA ESPERADA	Alta		RENDIMIENTO		< 1 segundo
IMPORTANCIA	Alta		URGENCIA		Moderada
ESTADO	Terminado		ESTABILIDAD		Alta





ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Registrar procedimiento realizado				UC009
ACTORES	Técnico				
TIPO	Primario				
REFERENCIAS	RF15		UC008		
PRECONDICIÓN	Servicio en estado "En proceso".				
POSTCONDICIÓN	Procedimiento registrado en historial de reparación.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Documentar los pasos técnicos realizados en una reparación.					
RESUMEN					
Técnico describe trabajo realizado, tiempo invertido y observaciones.					
CURSO NORMAL					
1	Técnico selecciona orden.		3	Sistema guarda en historial.	
2	Ingresa descripción del procedimiento y tiempo.				
CURSOS ALTERNOS					
OTROS DATOS					
FRECUENCIA ESPERADA	Alta		RENDIMIENTO	< 3 segundos	
IMPORTANCIA	Moderada		URGENCIA	Moderada	
ESTADO	Terminado		ESTABILIDAD	Alta	

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Consultar historial de dispositivo				UC010
ACTORES	Técnico, Recepcionista				
TIPO	Secundario				
REFERENCIAS	RF03				
PRECONDICIÓN	Dispositivo registrado con reparaciones previas.				
POSTCONDICIÓN	Historial de reparaciones mostrado.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Revisar reparaciones anteriores de un dispositivo.					
RESUMEN					
Usuario selecciona dispositivo y sistema muestra historial ordenado.					
CURSO NORMAL					
1	Usuario busca dispositivo.	3	Sistema lista reparaciones anteriores con detalles.		
2	Selecciona "Ver historial".				
CURSOS ALTERNOS					
2a	Sin historial → mensaje informativo.				
OTROS DATOS					
FRECUENCIA ESPERADA	Moderada	RENDIMIENTO	< 2 segundos		
IMPORTANCIA	Moderada	URGENCIA	Baja		
ESTADO	Terminado	ESTABILIDAD	Alta		

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Gestionar usuarios del sistema				UC011
ACTORES	Administrador				
TIPO	Primario				
REFERENCIAS	RF01	UC001			
PRECONDICIÓN	Usuario autenticado como Administrador.				
POSTCONDICIÓN	Usuarios creados, modificados o eliminados.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Administrar cuentas de usuario y asignar roles.					
RESUMEN					
Administrador realiza CRUD de usuarios y asigna roles (Recepcionista, Técnico, Administrador).					
CURSO NORMAL					
1	Administrador accede a "Gestión de usuarios".		3	Sistema valida y guarda.	
2	Crea/edita usuario con datos y rol.				
CURSOS ALTERNOS					
2a	Datos duplicados (usuario existente) → error.				
OTROS DATOS					
FRECUENCIA ESPERADA	Baja (ocasional)		RENDIMIENTO	< 3 segundos	
IMPORTANCIA	Alta		URGENCIA	Moderada	
ESTADO	Terminado		ESTABILIDAD	Alta	

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Generar reportes de productividad				UC012
ACTORES	Administrador				
TIPO	Secundario				
REFERENCIAS	RF19				
PRECONDICIÓN	Debe existir historial de reparaciones.				
POSTCONDICIÓN	Se genera un reporte con métricas de productividad.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Proporcionar métricas de rendimiento por técnico y tiempos de reparación.					
RESUMEN					
El administrador selecciona período y tipo de reporte, y el sistema lo genera.					
CURSO NORMAL					
1	Administrador selecciona "Reportes".		3	Sistema calcula métricas y muestra reporte.	
2	Administrador define las métricas para generar el reporte				
CURSOS ALTERNOS					
2a	Sin datos en el período → mensaje informativo.				
OTROS DATOS					
FRECUENCIA ESPERADA	Baja (mensual)		RENDIMIENTO	< 5 segundos	
IMPORTANCIA	Moderada		URGENCIA	Baja	
ESTADO	Terminado		ESTABILIDAD	Alta	

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Configurar sistema de backup				UC013
ACTORES	Administrador				
TIPO	Secundario				
REFERENCIAS	RF21	UC001			
PRECONDICIÓN	Acceso de administrador.				
POSTCONDICIÓN	Configuración de backup guardada.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Establecer parámetros de copias de seguridad automáticas.					
RESUMEN					
Administrador programa frecuencia, destino y retención de backups.					
CURSO NORMAL					
1	Administrador accede a "Configuración" y "Backup".		3	Sistema guarda configuración.	
2	Define frecuencia y por cuanto tiempo se conservarán las backups en el sistema.				
CURSOS ALTERNOS					
OTROS DATOS					
FRECUENCIA ESPERADA	Muy baja (inicial)		RENDIMIENTO	< 3 segundos	
IMPORTANCIA	Baja		URGENCIA	Baja	
ESTADO	Terminado		ESTABILIDAD	Baja	

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Ver estadísticas de negocio				UC014
ACTORES	Administrador				
TIPO	Secundario				
REFERENCIAS	RF19	UC001			
PRECONDICIÓN	Datos históricos disponibles.				
POSTCONDICIÓN	Panel de estadísticas mostrado.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Visualizar métricas de negocio (ganancias, reparaciones, etc.)					
RESUMEN					
Sistema muestra gráficos y KPIs de negocio.					
CURSO NORMAL					
1	Administrador accede a "Reportes".		2	Sistema carga panel con gráficos y resúmenes	
CURSOS ALTERNOS					
OTROS DATOS					
FRECUENCIA ESPERADA	Baja (semanal)		RENDIMIENTO	< 2 segundos	
IMPORTANCIA	Moderada		URGENCIA	Baja	
ESTADO	Terminado		ESTABILIDAD	Alta	

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Gestionar gastos				UC015
ACTORES	Administrador				
TIPO	Secundario				
REFERENCIAS	RF17		UC001		
PRECONDICIÓN	Usuario autenticado con permisos.				
POSTCONDICIÓN	Gasto registrado en historial.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Registrar un gasto operativo del taller.					
RESUMEN					
El usuario registra un gasto operativo del taller completando fecha, categoría, descripción, monto y comprobante; el sistema lo guarda en el historial y recalcula totales.					
CURSO NORMAL					
1	Administrador selecciona "Gastos" → "Nuevo gasto" o "Editar gasto"		3	Sistema guarda y actualiza historial.	
2	Ingresa fecha, categoría, descripción, monto, comprobante.				
CURSOS ALTERNOS					
2a	Monto inválido (≤0) → error.				
OTROS DATOS					
FRECUENCIA ESPERADA	Moderada (varios por semana)		RENDIMIENTO	< 2 segundos	
IMPORTANCIA	Moderada		URGENCIA	Moderada	
ESTADO	Terminado		ESTABILIDAD	Alta	



ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Gestionar productos en inventario				UC016
ACTORES					
TIPO	Primario				
REFERENCIAS	RF12	UC001			
PRECONDICIÓN	Usuario autenticado como Administrador.				
POSTCONDICIÓN	Producto agregado al inventario, stock y valor actualizados.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Incorporar un nuevo producto al catálogo de inventario.					
RESUMEN					
El administrador agrega un producto al inventario con datos básicos, precios, stock y proveedor; el sistema calcula automáticamente margen y valor del inventario.					
CURSO NORMAL					
1	Administrador selecciona "Inventario" → "Nuevo producto" o "Editar producto"	3	Sistema guarda y actualiza historial.		
2	Completa nombre, categoría, proveedor, precios, stock.				
CURSOS ALTERNOS					
2a	Stock mínimo mayor que stock actual → alerta.				
	Imagen >5MB → error.				
OTROS DATOS					
FRECUENCIA ESPERADA	Moderada (semanal)	RENDIMIENTO	< 3 segundos		
IMPORTANCIA	Alta	URGENCIA	Alta		
ESTADO	Terminado	ESTABILIDAD	Alta		

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Gestionar problemas comunes				UC017
ACTORES	Administrador				
TIPO	Secundario				
REFERENCIAS	RF07		UC001		
PRECONDICIÓN	Usuario autenticado como Administrador.				
POSTCONDICIÓN	Catálogo de problemas actualizado.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Mantener un listado de problemas frecuentes para selección rápida					
RESUMEN					
El administrador agrega, edita o elimina problemas comunes desde la configuración, y estos quedan disponibles para su selección al crear órdenes de servicio.					
CURSO NORMAL					
1	Administrador accede a Configuración → Problemas.		3	Sistema guarda y muestra en lista.	
2	Agrega/edita problema.				
CURSOS ALTERNOS					
2a	Problema duplicado → advertencia.				
OTROS DATOS					
FRECUENCIA ESPERADA	Baja (ocasional)		RENDIMIENTO	< 2 segundos	
IMPORTANCIA	Moderada		URGENCIA	Baja	
ESTADO	Terminado		ESTABILIDAD	Alta	

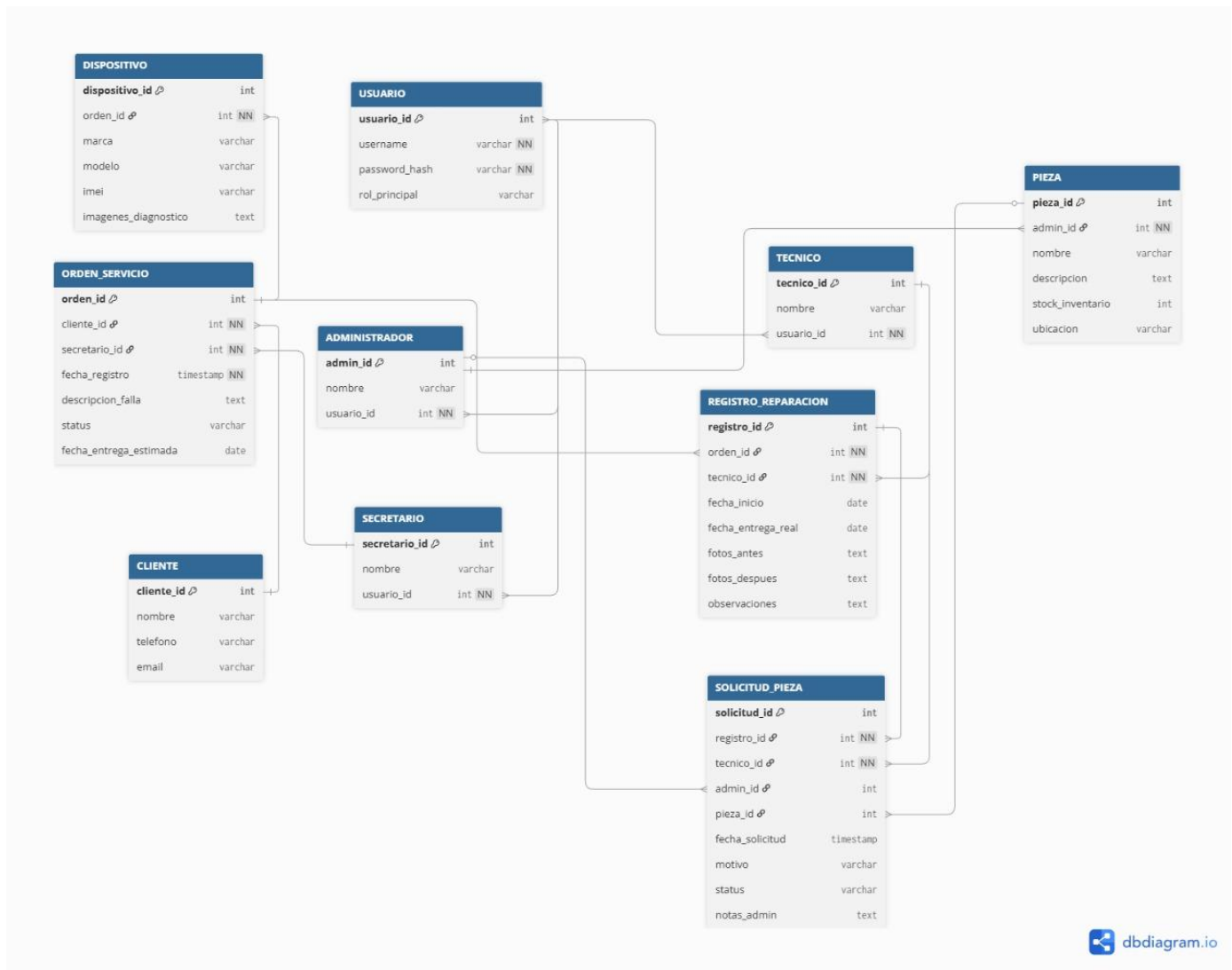
ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Gestionar categorías de servicios				UC018
ACTORES	Administrador				
TIPO	Secundario				
REFERENCIAS	RF08		UC001		
PRECONDICIÓN	Acceso de administrador.				
POSTCONDICIÓN	Categorías actualizadas, disponibles en órdenes de servicio.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Configurar los tipos de reparaciones ofrecidas con precios y tiempos base.					
RESUMEN					
El administrador define categorías de servicio con nombre, precio base y tiempo estimado; estas categorías se utilizan para preconfigurar órdenes de servicio y cálculos de costos.					
CURSO NORMAL					
1	Administrador accede a Configuración → Categorías de servicios.		3	Sistema valida y guarda.	
2	CRUD de categorías con nombre, precio, tiempo. (el administrador elige la opción que requiera)				
CURSOS ALTERNOS					
2a	Precio o tiempo inválido → error.				
OTROS DATOS					
FRECUENCIA ESPERADA	Baja (inicial y ajustes ocasionales)		RENDIMIENTO	< 2 segundos	
IMPORTANCIA	Alta		URGENCIA	Alta	
ESTADO	Terminado		ESTABILIDAD	Alta	

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Configurar estados de servicio				UC019
ACTORES	Administrador				
TIPO	Secundario				
REFERENCIAS	RF09		UC001		
PRECONDICIÓN	Acceso de administrador.				
POSTCONDICIÓN	Estados disponibles para uso en flujo de reparación.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Personalizar los estados por los que pasa una orden de servicio.					
RESUMEN					
El administrador crea, modifica o elimina estados personalizados para el flujo de reparación, asegurando que solo los estados configurados estén disponibles para los técnicos.					
CURSO NORMAL					
1	Administrador accede a Configuración → Estados de servicio.		3	Sistema valida y guarda.	
2	CRUD de los estados de servicio. (el administrador elige la opción que requiera)				
CURSOS ALTERNOS					
2a	Estado personalizado con nombre duplicado → error.				
OTROS DATOS					
FRECUENCIA ESPERADA		Muy baja (configuración inicial)	RENDIMIENTO		< 2 segundos
IMPORTANCIA		Alta	URGENCIA		Moderada
ESTADO		Terminado	ESTABILIDAD		Alta

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Gestionar marcas de productos				UC020
ACTORES	Administrador				
TIPO	Secundario				
REFERENCIAS	RF10		UC001		
PRECONDICIÓN	Acceso de administrador.				
POSTCONDICIÓN	Marcas disponibles en registro de productos o dispositivos.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Mantener catálogo de marcas para dispositivos reparables o productos en inventario.					
RESUMEN					
El administrador administra el catálogo de marcas (ej: Apple, Samsung), que luego se muestran como opciones en los formularios de registro de dispositivos, evitando entradas manuales inconsistentes.					
CURSO NORMAL					
1	Administrador accede a Configuración → Marcas.		3	Sistema actualiza lista desplegable en dispositivos.	
2	Agrega/edita/elimina marca.				
CURSOS ALTERNOS					
2a	Tipo duplicado → advertencia.				
OTROS DATOS					
FRECUENCIA ESPERADA		Baja (ocasional)	RENDIMIENTO		< 2 segundos
IMPORTANCIA		Alta	URGENCIA		Alta
ESTADO		Terminado	ESTABILIDAD		Alta

ESPECIFICACIÓN DE CASOS DE USO					
CASO DE USO	Gestionar tipos de dispositivos				UC021
ACTORES	Administrador				
TIPO	Secundario				
REFERENCIAS	RF11		UC001		
PRECONDICIÓN	Acceso de administrador.				
POSTCONDICIÓN	Tipos disponibles en registro de dispositivos.				
AUTOR	Analista del sistema	FECHA	08-12-25	VERSIÓN	1.0
PROPOSITO					
Clasificar dispositivos por tipo (Celular, Laptop, etc.).					
RESUMEN					
El administrador define los tipos de dispositivos que el taller repara; estos tipos se usan para clasificar y filtrar dispositivos en el sistema.					
CURSO NORMAL					
1	Administrador accede a Configuración → Tipos de dispositivos.		3	Sistema actualiza opciones en formulario de dispositivo.	
2	CRUD de tipos				
CURSOS ALTERNOS					
2a	Estado personalizado con nombre duplicado → error.				
OTROS DATOS					
FRECUENCIA ESPERADA	Muy baja (configuración inicial)		RENDIMIENTO	< 2 segundos	
IMPORTANCIA	Alta		URGENCIA	Moderada	
ESTADO	Terminado		ESTABILIDAD	Alta	

## Diagrama Relacional de BD:



## Arquitectura propuesta:

La arquitectura de **ReparaYA** se basa en tres capas (Niveles Lógicos) principales, siguiendo el patrón **Modelo-Vista-Controlador (MVC)**, implementado con **Spring Boot** para el *backend* y **PostgreSQL** para la persistencia.

## Flujo y componentes de la arquitectura

Nivel Lógico	Componente	Descripción	Tecnologías Clave
Capa de Presentación (Vista)	Vista (View)	Responsable de la interfaz de usuario. Muestra los datos y captura las interacciones del usuario. Generada dinámicamente por el <i>backend</i> .	HTML5, CSS3, JavaScript, Thymeleaf <sup>1</sup>
Capa de Lógica (Controlador/Servicio)	Controlador (Controller)	Actúa como el <i>router</i> o puerta de entrada. Recibe las peticiones HTTP del navegador, valida la sesión, y decide qué lógica de negocio (Servicio) llamar.	Java (Spring Boot) <sup>2</sup>
	Servicio (Service)	Contiene toda la Lógica de Negocio y orquestación. Realiza cálculos, validaciones, y gestiona las transacciones de datos.	Java (Spring Boot) <sup>3</sup>
Capa de Datos (Modelo/Persistencia)	Modelo (Model)	Define la estructura de los datos (POJOs). Representa las entidades de negocio (Cliente, Servicio, Gasto, etc.).	Java POJOs
	Persistencia/Base de Datos	Responsable de almacenar y recuperar los datos de forma segura.	PostgreSQL <sup>4</sup>



## Representación del flujo de información (Diagrama de bloques)

El flujo de información en el sistema sigue un ciclo predecible:

1. **USUARIO** interactúa con el **NAVEGADOR**.
2. **NAVEGADOR** envía una **Petición HTTP** (ej. /clientes).
3. La petición es interceptada por el **CONTROLADOR** (ej. ClienteController).
4. El **CONTROLADOR** llama al **SERVICIO** (ej. ClienteService) para ejecutar la lógica de negocio (ej. obtenerTodos()).
5. El **SERVICIO** interactúa con la **PERSISTENCIA (PostgreSQL)** para leer o escribir los datos, usando el **MODELO** como estructura.
6. Los datos del **MODELO** son devueltos al **SERVICIO**, y luego al **CONTROLADOR**.
7. El **CONTROLADOR** selecciona la **VISTA (Thymeleaf)** apropiada y le inyecta los datos.
8. El **NAVEGADOR** renderiza la **VISTA** final para el **USUARIO**.

A[Usuario/Navegador] -->|1. Petición HTTP| B(Controlador)

B -->|2. Llama a la lógica| C{Servicio}

C -->|3. Leer/Escribir datos| D[Base de Datos: PostgreSQL]

D -->|4. Devuelve Datos (Model)| C

C -->|5. Devuelve Resultado| B

B -->|6. Renderiza Vista| E[Vista: HTML/Thymeleaf]

E -->|7. Respuesta al Usuario| A

## 12. Funcionalidad en tiempo real

El sistema **ReparaYA** está diseñado para proporcionar una respuesta inmediata en operaciones críticas, adhiriéndose al requisito de rendimiento que exige un tiempo de respuesta inferior a 1 segundo para el 95% de las transacciones.

Las principales funcionalidades con respuesta en tiempo real o recuperación de información inmediata son:

- **Historial completo por dispositivo:** El sistema permite consultar el historial completo de reparaciones de un dispositivo específico de forma instantánea. Esto incluye el registro de procedimientos realizados y el seguimiento de cada cambio de estado.
- **Registro organizado de clientes:** El Administrador puede acceder y gestionar de forma inmediata el listado completo de clientes , permitiendo actualizar datos de contacto y asociar rápidamente un nuevo dispositivo a un cliente existente.
- **Búsqueda rápida:** La búsqueda unificada de clientes y dispositivos muestra resultados de forma inmediata al escribir, facilitando la localización de registros (RF007).
- **Actualización de KPIs:** Los indicadores clave de desempeño (KPIs) en el *Dashboard* se actualizan inmediatamente después de cualquier cambio en el estado de un servicio, manteniendo una visión precisa del estado del taller.

### 13. Trabajo a Futuro

El plan de trabajo a futuro se centrará en dos ámbitos de corrección y expansión que son críticos para la versión final del sistema **ReparaYA**.

#### **Ámbito 1: Corrección crítica y Persistencia de datos (Prioridad Máxima)**

La primera fase del trabajo a futuro se enfocará en resolver los problemas de conectividad encontrados durante la implementación, lo cual es vital para migrar el sistema de la operación en memoria a la persistencia.

- **Implementación de persistencia:** Resolver el problema de conexión con la base de datos **PostgreSQL** para que el sistema pueda almacenar, recuperar y actualizar los datos de manera permanente.
- **Activación de roles:** Desarrollar las vistas específicas, restricciones de seguridad y lógica de negocio para habilitar por completo los roles de **Recepcionista y Técnico**, que actualmente solo operan bajo el perfil de Administrador.

#### **Ámbito 2: Expansión funcional y requisitos (COULD HAVE)**

Una vez resueltos los problemas de persistencia y roles, el plan de expansión se centrará en el desarrollo de funcionalidades que agregan valor y completan los requisitos definidos como **COULD HAVE**:

- **Gestión completa:** Desarrollar **Sistema de Backup (RF014)** para copias de seguridad automáticas y restauración de datos.
- **Reportes avanzados (RF012):** Implementar la generación de métricas de productividad por técnico y estadísticas de tiempos promedio de reparación y ganancias por período.
- **Módulo de herramientas (RF013):** Desarrollar la gestión de herramientas utilizadas en reparaciones, incluyendo su estado y mantenimiento.
- **Expansión del negocio:** Configurar la arquitectura para soportar operativamente hasta **tres tiendas simultáneas** (RNF005).

## 14. Conclusiones

El desarrollo de **ReparaYA** culminó con éxito la implementación de un **Producto Mínimo Viable (MVP)** funcional, logrando los principales objetivos de la fase de análisis. La adopción de la metodología ágil, combinada con la arquitectura **MVC** sobre **Spring Boot**, garantizó un desarrollo ordenado y una estructura de código fácil de mantener y expandir.

Los objetivos de negocio clave, como la **Trazabilidad** de los servicios y el **Control de inventario** con alertas, se cumplieron plenamente en la lógica de la aplicación (RF03, RF04, RF05).

No obstante, es crucial documentar dos limitaciones técnicas que definen el alcance final de esta versión:

1. **Conexión a la base de datos:** La configuración de la conexión con **PostgreSQL** **no se completó con éxito**. Por esta razón, el sistema en su estado actual opera con toda la lógica de negocio y los datos gestionados **en memoria**, lo que requiere la corrección inmediata de la persistencia para futuras versiones.
2. **Control de acceso limitado:** Aunque el diseño contempla los roles de **Administrador, Recepcionista y Técnico**, esta versión solo permite la funcionalidad completa bajo el rol de **Administrador**. El desarrollo de las interfaces y restricciones específicas para los roles de Recepcionista y Técnico se pospone para el trabajo a futuro.

A pesar de estas limitaciones, el proyecto establece una base sólida y una estructura lógica bien definida. La corrección de la persistencia y la activación de los roles pendientes son la **máxima prioridad** para el trabajo a futuro, permitiendo así liberar el potencial completo del sistema ReparaYA.

## 15. Equipo de Trabajo

El equipo de trabajo estuvo compuesto por dos integrantes, cuyas responsabilidades fueron divididas de manera complementaria a lo largo del ciclo de vida del software, cubriendo desde la conceptualización hasta la implementación.

Integrante	Rol principal	Responsabilidades y Fases cubiertas
<b>Karla Sofía Castro Pérez</b>	<b>Analista de Requisitos y Documentación</b>	Lideró las fases iniciales del ciclo de vida del software: <b>Planeación, Análisis</b> (necesidades de usuario, RF y RNF, casos de uso). También se encargó del <b>Diseño del Prototipo</b> y de la <b>Documentación completa</b> del proyecto.
<b>Emma Hernández Mendoza</b>	<b>Desarrollador Full-Stack</b>	Lideró la fase de <b>Implementación</b> del proyecto, incluyendo el desarrollo completo del <b>Backend</b> (lógica de negocio y controladores en Java/Spring Boot) y el desarrollo del <b>Frontend</b> (vistas y estilos con Thymeleaf/CSS/JS).
<b>Jose Evaristo Rodríguez López</b>	<b>Especialista en Bases de Datos</b>	Se encargó del <b>Diseño de la Base de Datos Relacional (PostgreSQL)</b> , la creación de la base de datos física y la configuración inicial de la <b>conexión con el Backend</b> . <i>Nota: Se documenta que la conexión con el backend no fue exitosa en esta versión, por lo que el sistema opera actualmente con datos en memoria.</i>