

Analisis Funcional y requisitos

1. Contexto y Objetivos del Sistema

1.1 Contexto

EcoRent Norte S.L. es una pyme dedicada al alquiler de maquinaria ligera desde una única sede. Actualmente opera mediante:

- Hojas de cálculo compartidas
- Registros manuales en papel
- Plantillas básicas para contratos
- Registro de pagos en documentos separados

Problemas identificados:

- Duplicidad de reservas
- Errores en fechas de devolución
- Desconocimiento del estado real de los equipos (disponible/mantenimiento/alquilado)
- Pérdida de histórico
- Falta de reporting consolidado
- Retrasos en atención en horas punta

Restricciones:

- Presupuesto limitado
 - Bajo nivel técnico del personal
 - No detener la operación diaria
 - Preservar información histórica
-

1.2 Objetivo General

Diseñar e implementar un **Sistema de Gestión de Alquileres** que:

- Centralice la información
 - Controle disponibilidad en tiempo real
 - Reduzca errores administrativos
 - Mejore la atención al cliente
 - Proporcione métricas clave de negocio
-

1.3 Objetivos Específicos

1. Eliminar duplicidades de reservas.
2. Automatizar el control de disponibilidad.
3. Gestionar clientes y contratos digitalmente.
4. Registrar pagos asociados a cada alquiler.
5. Gestionar el estado de mantenimiento de equipos.
6. Proporcionar información consolidada para la dirección.

2. Stakeholders y Necesidades

Stakeholder	Rol	Necesidades
Dirección	Toma de decisiones	Métricas de rentabilidad, equipos más alquilados, ingresos por periodo
Atención al cliente (4 empleados)	Gestión operativa	Crear reservas rápidamente, consultar disponibilidad en segundos
Técnicos de mantenimiento	Control de equipos	Marcar equipos en mantenimiento y bloquear alquiler
Clientes (indirectos)	Usuario final del servicio	Proceso ágil, menos errores, claridad en fechas
Administración	Gestión financiera	Control de pagos, historial por cliente
Responsable IT (si aplica)	Soporte técnico	Sistema simple, fácil despliegue

3. Alcance del Sistema

3.1 Incluye

- ☒ Gestión de equipos
- ☒ Gestión de clientes
- ☒ Gestión de reservas/alquileres
- ☒ Control de disponibilidad
- ☒ Gestión de estados (Disponible, Alquilado, Mantenimiento)
- ☒ Registro de pagos
- ☒ Consulta histórica
- ☒ Reportes básicos
- ☒ Interfaz web interna
- ☒ Persistencia en H2 (modo file)
- ☒ Despliegue local en servidor interno

3.2 Excluye (fase actual)

- ✗ Pasarela de pago online
 - ✗ Portal de clientes externos
 - ✗ Multisede
 - ✗ Integración contable externa
 - ✗ App móvil
 - ✗ Facturación electrónica oficial
-

4. Requisitos Funcionales (RF)

Redactados en formato verificable.

4.1 Gestión de Equipos

RF-01 El sistema debe permitir registrar un equipo con: nombre, categoría, código interno, precio por día y estado.

RF-02 El sistema debe permitir modificar los datos de un equipo existente.

RF-03 El sistema debe permitir cambiar el estado de un equipo a:

- Disponible
- Alquilado
- En mantenimiento

RF-04 El sistema debe impedir reservar equipos en estado "En mantenimiento".

4.2 Gestión de Clientes

RF-05 El sistema debe permitir registrar clientes con: nombre, DNI/NIF, teléfono y email.

RF-06 El sistema debe permitir consultar el historial de alquileres por cliente.

4.3 Gestión de Alquileres

RF-07 El sistema debe permitir crear un alquiler indicando:

- Cliente

- Equipo
- Fecha inicio
- Fecha fin

RF-08 El sistema debe validar que no exista solapamiento de fechas para el mismo equipo.

RF-09 El sistema debe calcular automáticamente el importe total (precio por día × número de días).

RF-10 El sistema debe actualizar automáticamente el estado del equipo a “Alquilado” cuando exista un alquiler activo.

RF-11 El sistema debe permitir registrar la devolución del equipo y cambiar su estado a “Disponible”.

4.4 Pagos

RF-12 El sistema debe permitir registrar pagos asociados a un alquiler.

RF-13 El sistema debe mostrar el estado del alquiler: Pendiente / Pagado parcial / Pagado completo.

4.5 Reportes

RF-14 El sistema debe permitir consultar:

- Equipos más alquilados
 - Ingresos por periodo
 - Clientes recurrentes
-

5. Requisitos No Funcionales (RNF)

5.1 Rendimiento

RNF-01 El sistema debe responder a cualquier operación de consulta en menos de 2 segundos bajo carga normal (hasta 5 usuarios concurrentes).

5.2 Disponibilidad

RNF-02 El sistema debe estar disponible durante horario laboral (8:00–20:00).

5.3 Seguridad

RNF-03 El sistema debe requerir autenticación mediante usuario y contraseña.

RNF-04 Las contraseñas deben almacenarse cifradas (BCrypt).

RNF-05 El acceso a endpoints REST debe estar protegido mediante Spring Security.

5.4 Usabilidad

RNF-06 La interfaz debe permitir crear una reserva en menos de 2 minutos por un usuario entrenado.

RNF-07 Debe utilizar formularios simples con validaciones visibles.

5.5 Persistencia

RNF-08 Los datos deben almacenarse en H2 modo file para evitar pérdida tras reinicio.

5.6 Mantenibilidad

RNF-09 Arquitectura en capas:

- Controller
- Service
- Repository
- Entity

6. Reglas de Negocio (RN)

RN-01 Un equipo no puede estar alquilado por más de un cliente en el mismo rango de fechas.

RN-02 Un equipo en mantenimiento no puede generar reservas.

RN-03 El precio total se calcula por días naturales completos.

RN-04 Un alquiler se considera activo si la fecha actual está entre fecha inicio y fecha fin y no se ha registrado devolución.

RN-05 No se puede registrar pago superior al importe total del alquiler.

7. Criterios de Aceptación Generales

El sistema será aceptado si:

1. No permite duplicidad de reservas.
 2. Permite consultar disponibilidad en menos de 2 segundos.
 3. Permite registrar alquiler completo en menos de 2 minutos.
 4. Genera reporte correcto de ingresos por periodo.
 5. Conserva datos tras reinicio del servidor.
 6. La dirección puede obtener listado de equipos más rentables.
-

8. Matriz de Trazabilidad Básica

Requisito	Módulo Backend	Componente Frontend	Base de Datos
RF-01 a RF-04	EquipmentController / Service	EquipmentView	Tabla EQUIPMENT
RF-05 a RF-06	ClientController	ClientView	Tabla CLIENT
RF-07 a RF-11	RentalController	RentalForm	Tabla RENTAL
RF-12 a RF-13	PaymentController	PaymentSection	Tabla PAYMENT
RF-14	ReportService	Dashboard	Consultas agregadas
RNF-03	SecurityConfig	LoginView	Tabla USER

9. Modelo Conceptual Simplificado

Entidades principales:

- **Equipment**
- **Client**
- **Rental**
- **Payment**
- **User**

Relaciones clave:

- 1 Client → N Rentals
 - 1 Equipment → N Rentals
 - 1 Rental → N Payments
-

10. Supuestos Adoptados

1. Todos los alquileres son por días completos.
 2. No existen descuentos complejos en fase inicial.
 3. Un equipo es una unidad física individual (no stock múltiple).
 4. El sistema será usado solo internamente.
-

11. Próximo Paso Recomendado

Dado que el proyecto se realizará en 2 días, recomiendo:

Día 1:

- Modelado de entidades JPA
- Implementación CRUD básico
- Validación de solapamiento

Día 2:

- Frontend React básico
 - Integración Axios
 - Reportes simples
 - Pruebas funcionales
-

Si lo deseas, el siguiente paso puede ser:

- ☒ Definir el Modelo de Datos detallado (entidades JPA completas)
- ☒ Definir los endpoints REST contract-first
- ☒ Diseñar los diagramas UML (clases y secuencia)
- ☒ Crear backlog priorizado para ejecución en 2 días
- ☒ Definir casos de prueba funcionales

Indícame el siguiente entregable que deseas desarrollar.