1. **VISIÓN DEL PROYECTO**
   1. **DESCRIPCIÓN GENERAL**

El Sistema de Gestión de Alquiler de Inmuebles es una solución tecnológica diseñada para optimizar y automatizar la administración de propiedades en alquiler para la empresa Lagartos S.A.C.. Este sistema permite a propietarios, administradores y usuarios gestionar inmuebles, contratos de alquiler, pagos, reportes y otros procesos relacionados de forma eficiente y centralizada. El objetivo principal es mejorar la experiencia del usuario y la gestión operativa mediante herramientas modernas y accesibles.

* 1. **OBJETIVOS PRINCIPALES**
* **Centralizar la gestión de inmuebles:** Facilitar el registro y alquiler de propiedades.
* **Optimizar procesos administrativos:** Automatizar tareas como generación de contratos, boletas de pago.
* **Proporcionar análisis detallados:** Ofrecer reportes sobre ocupación y tendencias de alquiler.
* Este objetivo busca proporcionar acceso multiplataforma, pero también asegurar que cada tipo de usuario (administradores, propietarios, inquilinos) acceda solo a las funcionalidades correspondientes a su rol.  
  permitir que solo los clientes registrados alquilen inmuebles asegura que el sistema sea funcional y organizado para los diferentes usuarios.
* **Fortalecer la seguridad:** Implementar estándares de seguridad para proteger datos sensibles.  
  Restringir el alquiler a clientes registrados es una medida de control que protege los datos y asegura que los inmuebles sean alquilados por usuarios identificados, lo que reduce el riesgo de fraude o mal uso del sistema.
  1. **STAKEHOLDERS**
* Propietarios
* Administradores de inmuebles
* Inquilinos
* Contadores/Financieros
* Equipo de TI

1. **ALCANCE Y OBJETIVOS**
   1. **MÓDULOS DEL SISTEMA**
      1. **Gestión de Inmuebles**

* Registro y actualización de propiedades.
* Clasificación por tipo: oficinas, locales comerciales, residenciales.
* Asignación de imágenes y descripciones detalladas.
* Estado de disponibilidad y ubicación geográfica.
  + 1. **Gestión de Contratos**
* Generación y almacenamiento de contratos de alquiler.
* Definición de términos y condiciones.
* Automatización de vencimientos y renovaciones
  + 1. **Gestiónde de Pagos**
* Registro de pagos realizados.
* Generación de recibos digitales.
  + 1. **Gestión de Usuarios**
* Registro de inquilinos.
* Acceso diferenciado por roles.
* Portal del usuario con información personalizada.

**2.1.5. Gestión de Alquiler**

* Validación automática de la disponibilidad del inmueble.
* Alquiler de inmuebles en estado *Disponible* si el usuario tiene saldo suficiente.
* Actualización del estado del inmueble tras el alquiler (*Disponible* → *Ocupado*).
* Historial de inmuebles alquilados por cada usuario.

1. **ARQUITECTURA DEL SISTEMA**
   1. **STACK TECNOLÓGICO**

**Frontend:**

* **HTML, CSS y JavaScript**: Usados para el diseño y la estructura básica de las vistas.
* **Bootstrap**: Para el diseño responsivo y la maquetación.
* **Spring Boot con Thymeleaf** : Gestiona las plantillas y facilita la integración entre las vistas del frontend y el backend.

**Backend:**

* **Spring Boot**: Framework principal para el desarrollo del backend. Permite implementar servicios REST y gestionar la lógica del negocio con rapidez.
* **Java**: Lenguaje base del backend para la implementación de la lógica.Jest para pruebas
* **Eclipse IDE**: Entorno de desarrollo integrado donde se escribe y gestiona el código Java y Spring Boot.

**Base de Datos**:

* **MySQL**: Sistema de gestión de bases de datos relacional que almacena la información sobre los inmuebles, contratos de alquiler, pagos, usuarios y reportes.
* Redis para caché
* Backups automatizados

**Infraestructura:**

* **Spring Boot Embedded Server**: Para ejecutar la aplicación web directamente sin necesidad de servidores externos adicionales como Apache Tomcat.
  1. **REQUISITOS NO FUNCIONALES**

**Disponibilidad:**

* Uptime: 99.9%
* Tiempo máximo de inactividad: 8.76 horas/año

**Rendimiento:**

* Tiempo de respuesta < 2 segundos
* Throughput: 1000 requests/segundo
* Latencia máxima: 200ms

**Seguridad:**

* Autenticación JWT
* Encriptación de datos sensibles
* Auditoría de acciones
* Backups cifrados

**Escalabilidad:**

* Horizontal para componentes stateless
* Vertical para base de datos
* Auto-scaling basado en carga

**4. HISTORIAS DE USUARIO**

**4.1. EPIC 1: Gestión de Usuarios**

**HU-001: Registro de Usuarios**  
**Como** usuario nuevo  
**Quiero** poder registrarme en el sistema  
**Para** acceder a las funcionalidades según mi rol (administrador, propietario, o inquilino)

**Criterios de Aceptación**:

1. El formulario debe incluir:
   * Datos personales (nombre, apellidos, DNI, correo electrónico, teléfono).
   * Contraseña con validación de seguridad.
   * Selección de rol de usuario (administrador, propietario o inquilino).
2. Validar que el DNI y el correo electrónico no estén duplicados.

**Estimación**: 5 puntos  
**Prioridad**: Alta

**HU-002: Gestión de Perfiles**  
**Como** usuario registrado  
**Quiero** editar los datos de los perfiles  
**Para** mantener mis datos siempre actualizados

**Criterios de Aceptación**:

1. Permitir la edición de:
   * Datos de contacto
   * Nombres
   * Apellidos
   * Correo
2. Gestionar los datos a cualquier hora

**Estimación**: 3 puntos  
**Prioridad**: Media

**4.2. EPIC 2: Gestión de Inmuebles**

**HU-003: Registro de Inmuebles**  
**Como** administrador  
**Quiero** registrar nuevos inmuebles en el sistema  
**Para** hacerlos visibles y disponibles para alquiler

**Criterios de Aceptación**:

1. Permitir el ingreso de:
   * Nombre del inmueble.
   * Tipo (oficina, local comercial, residencial).
   * Dirección, precio y descripción.
   * Estado inicial (disponible, ocupado, suspendido).
2. Mostrar la ubicación geográfica.

**Estimación**: 8 puntos  
**Prioridad**: Alta

**HU-004: Clasificación de Inmuebles**  
**Como** usuario  
**Quiero** filtrar inmuebles según su tipo y estado  
**Para** encontrar más rápido el inmueble que me interesa

**Criterios de Aceptación**:

1. Filtrar por:
   * Tipo de inmueble (oficina, local comercial, residencial).
   * Estado (disponible, ocupado, suspendido).
   * Rango de precios.
2. Mostrar resultados en una lista con imágenes y descripción básica.

**Estimación**: 5 puntos  
**Prioridad**: Media

**4.3. EPIC 3: Gestión de Contratos**

**HU-005: Generación de Contratos**  
**Como** administrador  
**Quiero** generar automáticamente un contrato al alquilar un inmueble  
**Para** formalizar el acuerdo entre las partes

**Criterios de Aceptación**:

1. Permitir incluir:
   * Fechas de inicio y fin del contrato.
2. Validar que el inmueble esté en estado disponible.
3. Guardar automáticamente el contrato en la base de datos.

**Estimación**: 8 puntos  
**Prioridad**: Alta

**4.5. EPIC 5: Reportes y Análisis**

**HU-007: Reporte de Inmuebles**  
**Como** administrador  
**Quiero** generar reportes sobre la ocupación de inmuebles  
**Para** analizar la rentabilidad de las propiedades

**Criterios de Aceptación**:

1. Generar reportes con:
   * Inmuebles ocupados vs. disponibles.
   * Tendencias de alquiler por tipo de inmueble.
   * Ingresos totales por períodos específicos.
2. Mostrar estadísticas visuales mediante gráficos.

**Estimación**: 8 puntos  
**Prioridad**: Media

**Notas Técnicas**

**Definition of Ready (DoR):**

* Criterios de aceptación definidos
* Estimación realizada
* Dependencias identificadas
* Recursos necesarios disponibles

**Definition of Done (DoD):**

* Código implementado y probado
* Pruebas automatizadas escritas
* Documentación actualizada
* Code review completado
* Desplegado en ambiente de pruebas
* Aprobado por Product Owner

**Priorización:**

* Alta: Crítico para el MVP
* Media: Importante pero no bloqueante
* Baja: Deseable, pero puede esperar

**Estimación:**

* 1-2 puntos: Tarea simple
* 3-5 puntos: Complejidad media
* 8 puntos: Tarea compleja
* 13+ puntos: Requiere división

1. **ANEXOS**
   1. **DIAGRAMAS**

* Arquitectura del sistema  
  Modelo Caso de Uso del Negocio:  
  Diagrama, Esquemático

  Descripción generada automáticamente

Gestion usuario:  
Caso de uso rol admin  
Diagrama

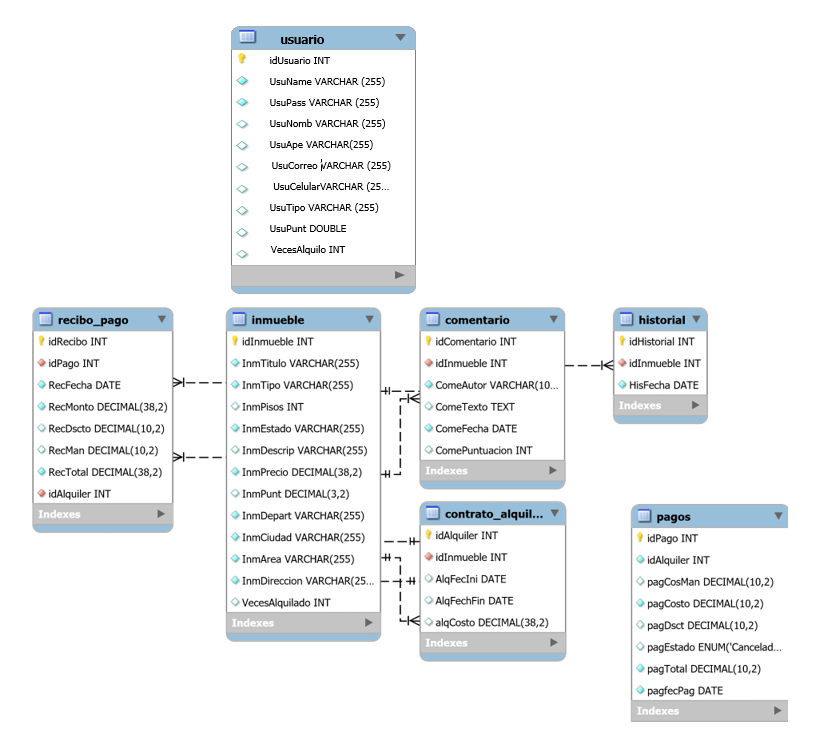
Descripción generada automáticamente  
  
Gestion Inmueble:  
Caso de uso rol admin   
Diagrama

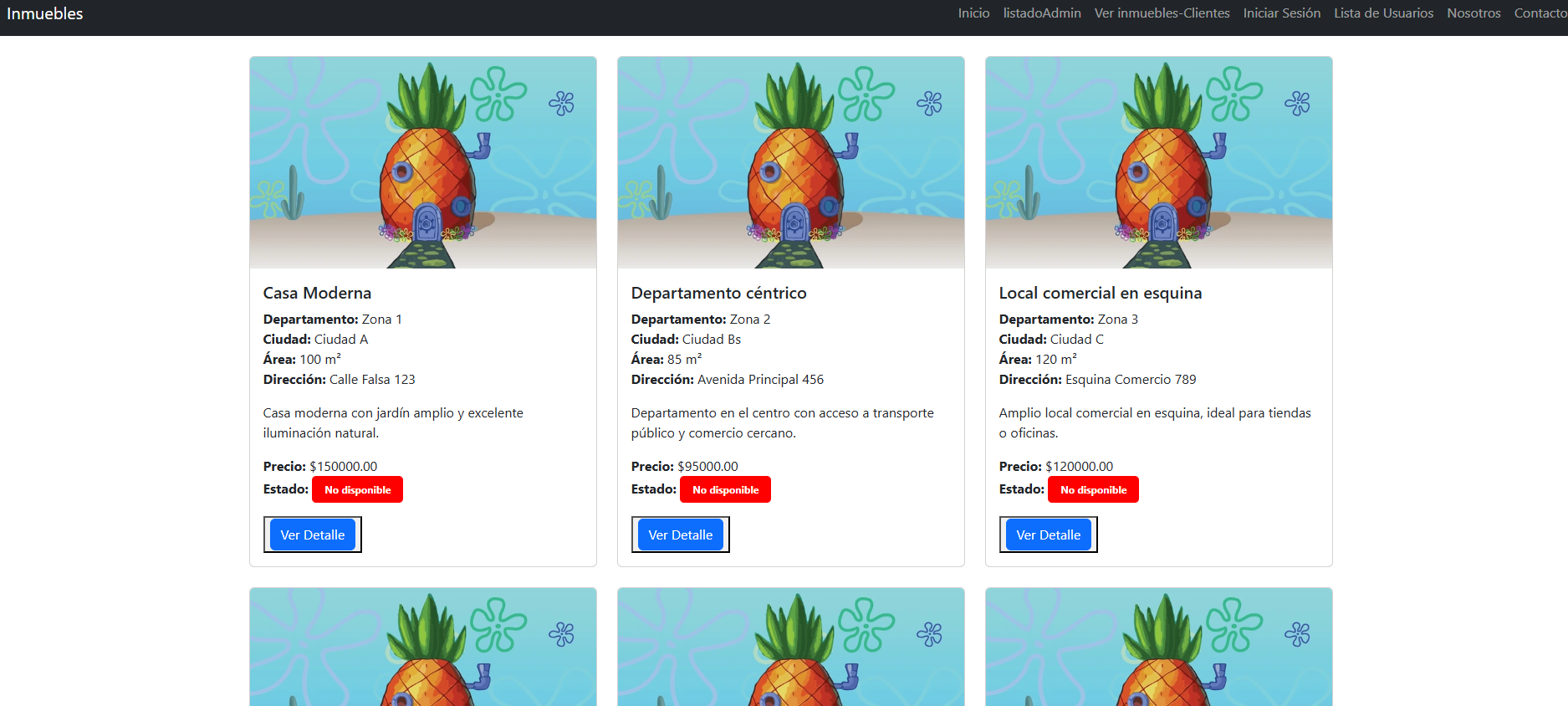
Descripción generada automáticamente

Gestión Inmueble:  
Caso de uso rol admin:  
Diagrama

Descripción generada automáticamente

* Modelo de datos



* 1. **MOCKUPS**
* Interfaces principales  
  interfaz principal  
    
    
  Interfaz de Inmuebles:  
    
    
  Interfaz de Administrador de Inmuebles:  
  

**GITHUB**

* CÓDIGO FUENTE
* DOCUMENTACION
* INFORME
* BD (SCRIPT)
* MANUAL DE INSTALACIÓN Y CONFIGURACIÓN