

RENT.IT

CFGS DAW

CURSO 2023/2024

SERGIO COCA PÉREZ

RAÚL CALVO HERNÁNDEZ.

DAVID SÁNCHEZ SÁNCHEZ

COLEGIO CALASANZ SALAMANCA

Índice

1. Introducción	3
2. Descripción de la aplicación.	3
3. Plan de empresa	3
4. Tecnologías escogidas y justificación	5
5. Diseño de la aplicación	6
Diagramas y definición de casos de uso	6
Diagramas de clase	7
Modelo entidad relación	8
6. Arquitectura de la aplicación.....	8
Estructura del proyecto	8
Librerías externas utilizadas	8
7. Manual de despliegue	9

1. Introducción

En la era digital actual, la gestión eficiente de recursos y materiales es crucial para la industria de la construcción. Rent.it surge como una solución innovadora destinada a facilitar el alquiler y la consulta de maquinaria de construcción. Esta memoria describe el desarrollo y la implementación de Rent.it, una aplicación diseñada para optimizar la disponibilidad y el uso de equipos de construcción mediante un sistema fácil de usar y accesible. El objetivo principal de este Trabajo de Fin de Grado es demostrar la capacidad de aplicar conocimientos teóricos a un proyecto práctico utilizando tecnologías modernas.

2. Descripción de la aplicación.

Rent.it es una plataforma web destinada a la gestión de alquileres y consultas de maquinaria de construcción. La aplicación permite a los empleados realizar alquileres de equipos específicos y realizar consultas sobre su disponibilidad. Entre las principales características de Rent.it se encuentran:

- **Catálogo de Maquinaria:** Visualización de una amplia gama de equipos disponibles para el alquiler.
- **Consulta de las Categorías:** Visualización de las diferentes categorías.
- **Eliminación Materiales:** Funcionalidad para la eliminación de materiales obsoletos.
- **Sistema de Alquiler:** Funcionalidad para reservar y alquilar maquinaria.
- **Consulta Disponibilidad:** Capacidad para verificar la disponibilidad de equipos.
- **Generación de Documentos:** Emisión de recibos en formato PDF para las transacciones realizadas.

3. Plan de empresa

El plan de empresa de Rent.it se centrará en varios aspectos clave para asegurar su éxito y sostenibilidad a largo plazo. Algunos de los puntos a considerar incluyen:

- **Mercado Objetivo**

Rent.it está dirigida a:

- **Empresas Constructoras:** Necesitan alquilar maquinaria especializada para proyectos específicos y a corto plazo.

- **Proveedores de Maquinaria:** Desean ofrecer sus equipos en alquiler a un público más amplio.
- **PYMES del sector de la construcción:** Buscan opciones rentables para acceder a maquinaria sin incurrir en costos de compra.

Análisis de la Competencia

En el mercado existen diversas plataformas de alquiler de maquinaria, sin embargo, Rent.it se diferencia por:

- **Interfaz Intuitiva:** Fácil de usar, diseñada para usuarios de diferentes niveles técnicos.
- **Integración de Funcionalidades:** Desde la búsqueda y alquiler hasta la generación automática de contratos y recibos en PDF.
- **Actualización en Tiempo Real:** Información precisa y actualizada sobre la disponibilidad de equipos.

Oportunidades de Mercado

- **Crecimiento de la Construcción:** Aumento continuo en proyectos de construcción, incrementando la demanda de alquiler de maquinaria.
- **Preferencia por el Alquiler:** Las empresas prefieren alquilar en lugar de comprar equipos, lo que favorece la demanda de plataformas como Rent.it.

Estrategia de Marketing

Propuesta de Valor

Rent.it ofrece una solución completa para la gestión de alquiler de maquinaria, mejorando la eficiencia operativa y reduciendo costos para las empresas de construcción.

Canales de Marketing

- **Marketing Digital:** Uso de campañas en redes sociales (LinkedIn, Facebook, Instagram) y publicidad en buscadores (Google Ads).
- **Colaboraciones Estratégicas:** Alianzas con proveedores de maquinaria y empresas constructoras para la promoción cruzada.
- **Participación en Eventos:** Asistencia a ferias y exposiciones de la industria para demostrar la plataforma y captar clientes.
- **Tarifas por Transacción:** Cargos por cada alquiler realizado a través de la plataforma.
- **Publicidad y Listados Premium:** Espacios publicitarios y opciones de listados destacados para proveedores de maquinaria.

4. Tecnologías escogidas y justificación

Para el desarrollo de Rent.it, se ha optado por una combinación de tecnologías modernas y robustas que aseguran un rendimiento óptimo, escalabilidad, y una experiencia para el empleado fluida. A continuación, se justifica el uso de cada una de las tecnologías seleccionadas

1. Java:

- **Robustez y Escalabilidad:** Java es un lenguaje de programación orientado a objetos conocido por su estabilidad y capacidad para manejar aplicaciones a gran escala. Su uso asegura que la lógica del servidor sea sólida y capaz de soportar un gran número de transacciones simultáneas.
- **Seguridad:** Java ofrece características de seguridad integradas que son cruciales para proteger los datos de los usuarios y las transacciones financieras.

2. Hibernate:

- **Gestión de la Base de Datos:** Hibérnate es un framework de mapeo objeto-relacional (ORM) que facilita la interacción con la base de datos. Simplifica la escritura de consultas SQL y reduce el tiempo de desarrollo al manejar automáticamente las operaciones CRUD (Create, Read, Delete).
- **Compatibilidad y Eficiencia:** Hibernate proporciona una abstracción sobre las particularidades de diferentes bases de datos, mejorando la eficiencia y la compatibilidad del código.

3. REST:

- **Comunicación Cliente-Servidor:** La arquitectura REST es ideal para desarrollar servicios web ligeros y escalables. Permite operaciones CRUD a través de métodos HTTP estándar (GET, POST, PUT, DELETE).
- **Interoperabilidad:** REST APIs permiten que diferentes sistemas interactúen entre sí, facilitando la integración con otros servicios y plataformas en el futuro.

4. Vue.js:

- **Interfaces Reactivas y Dinámicas:** Vue.js es un framework progresivo de JavaScript que facilita la creación de interfaces de usuario reactivas y dinámicas. Su diseño modular permite agregar funcionalidades de manera incremental.
- **Facilidad de Uso:** Vue.js es conocido por su curva de aprendizaje baja, lo que permite a los desarrolladores ser productivos rápidamente y construir componentes reutilizables de manera eficiente.

5. HTML, CSS, Java Script:

- **Desarrollo Front-end:** HTML, CSS y JavaScript son las tecnologías esenciales para construir la interfaz de usuario. HTML proporciona la estructura del contenido, CSS se encarga del diseño y el estilo, y JavaScript añade interactividad.

6. MySQL

- **Almacenamiento Persistente de Datos:** MySQL es una base de datos relacional ampliamente utilizada, conocida por su rendimiento, confiabilidad y facilidad de uso. Es ideal para gestionar grandes volúmenes de datos y realizar consultas complejas.
- **Compatibilidad y Soporte:** MySQL es compatible con Hibernate y otras tecnologías utilizadas en el proyecto, y cuenta con una amplia comunidad de soporte que facilita la resolución de problemas y la optimización del rendimiento.

7. Conclusión

La elección de estas tecnologías para el desarrollo de Rent.it se basa en sus capacidades para proporcionar una base sólida, escalable y segura, garantizando que el empleado pueda trabajar de la manera más cómoda posible. Cada tecnología se ha seleccionado por sus características específicas y su capacidad para integrarse eficientemente en la arquitectura del proyecto.

5. Diseño de la aplicación

Diagramas y definición de casos de uso

La sección de alquiler permite al empleado añadir, modificar, eliminar y buscar alquileres:

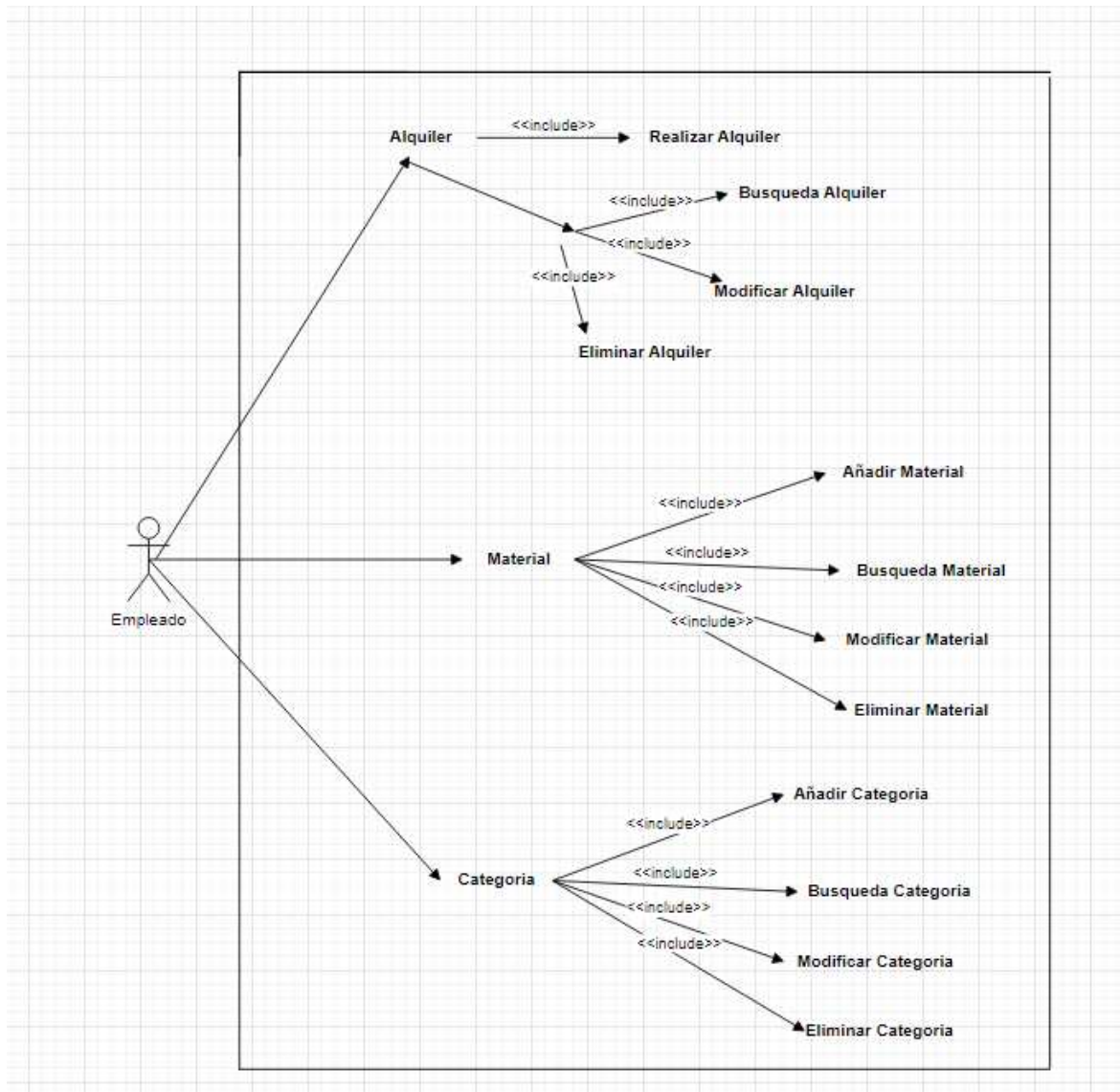
- **Insertar Alquiler:** En esta funcionalidad, el empleado puede seleccionar el material que se va a alquilar, indicando tanto una fecha de inicio como una fecha de finalización. Dependiendo del material y de las fechas seleccionadas para la creación del alquiler, el precio puede variar.
- **Buscar Alquiler:** En esta funcionalidad, el empleado podrá realizar una búsqueda de un alquiler.
- **Modificar Alquiler:** En esta funcionalidad, el empleado puede modificar un alquiler si fuese necesario.
- **Eliminar Alquiler:** En esta funcionalidad, el empleado puede eliminar un alquiler.

La sección material permite al empleado añadir, modificar, eliminar y buscar materiales:

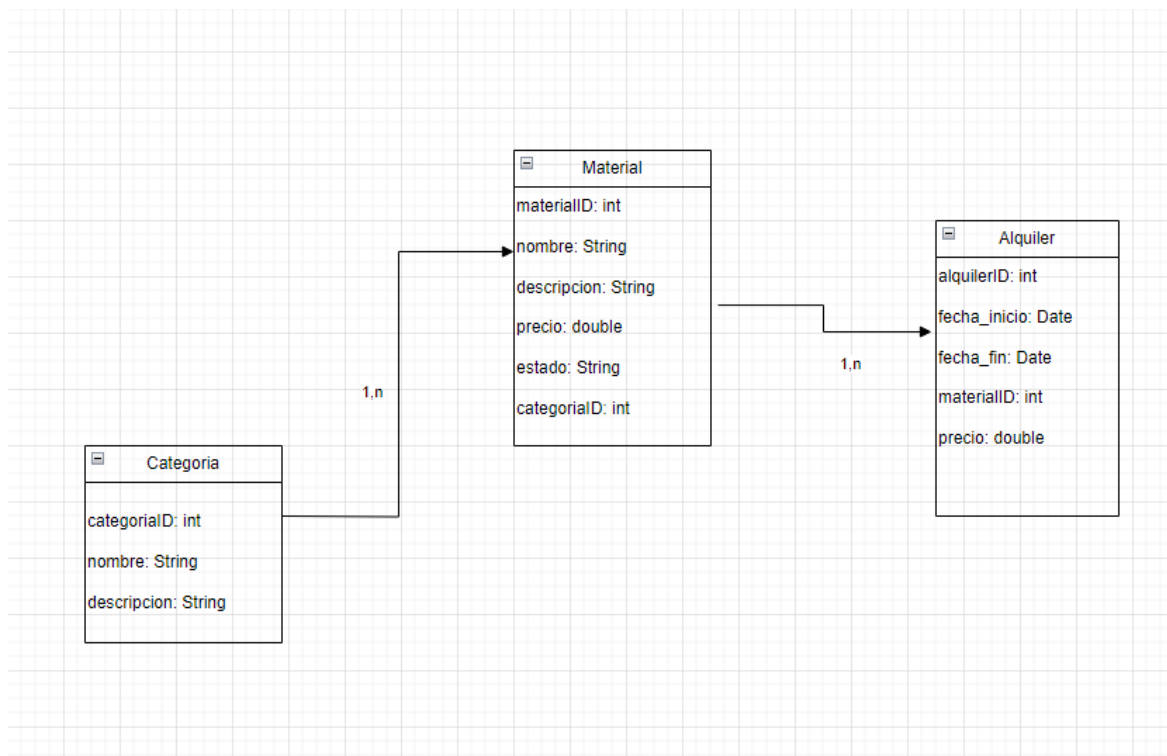
- **Insertar Material:** En esta funcionalidad, el empleado puede añadir un nuevo material que haya salido recientemente.
- **Buscar Material:** En esta funcionalidad, el empleado podrá realizar una búsqueda de un material.
- **Modificar Material:** En esta funcionalidad, el empleado puede modificar un material si fuese necesario.
- **Eliminar Material:** En esta funcionalidad, el empleado puede eliminar un material.

La sección categorías permite al empleado añadir, modificar, eliminar y buscar categorías:

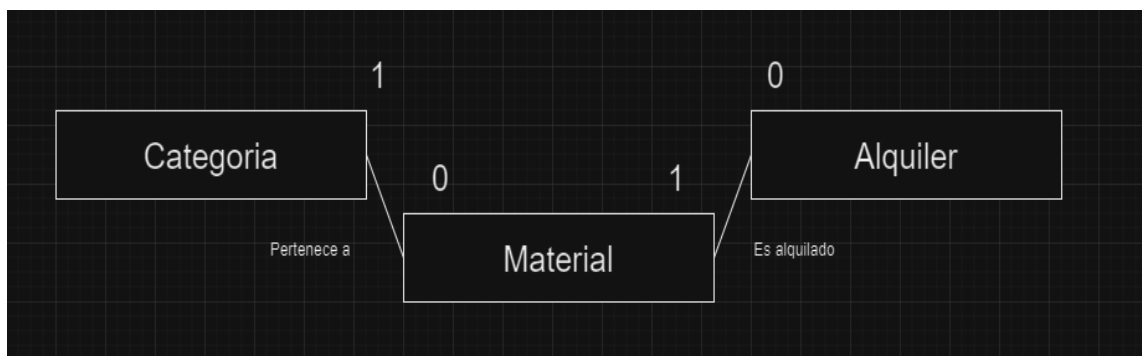
- **Insertar Categoría:** En esta funcionalidad, el empleado puede añadir una nueva categoría que haya salido recientemente.
- **Buscar Categoría:** En esta funcionalidad, el empleado podrá realizar una búsqueda de una categoría.
- **Modificar Categoría:** En esta funcionalidad, el empleado puede modificar una categoría si fuese necesario.
- **Eliminar Categoría:** En esta funcionalidad, el empleado puede eliminar una categoría.



Diagramas de clase



Modelo entidad relación



6. Arquitectura de la aplicación

Estructura del proyecto

La arquitectura del proyecto se divide en tres capas:

- **Backend:** Esta capa se encargará de gestionar la lógica de negocio de la aplicación. Emplearemos hibernate-REST para su desarrollo en Java. Esta capa será responsable de la realización de operaciones CRUD en la base de datos.
- **Frontend:** En esta capa, desarrollaremos la interfaz de usuario de la aplicación utilizando tecnologías web como HTML, CSS y JavaScript, complementadas con el framework Vue.js para crear interfaces dinámicas e interactivas. La comunicación entre la interfaz de usuario y el backend se realizará mediante API REST, lo que permitirá ejecutar operaciones CRUD sobre la base de datos de manera eficiente.
- **Base de Datos:** Es la capa en la que se almacenarán los datos de la aplicación. El sistema de gestión es MYSQL, en la propia base de datos almacenará las siguientes tablas:
 - **Categorías.**
 - **Material.**
 - **Alquiler.**

Librerías externas utilizadas

Axios

- **Peticiones HTTP:** Axios es una librería de JavaScript que facilita la realización de peticiones HTTP desde el front-end. Es compatible con promesas y proporciona una API fácil de usar para interactuar con los servicios REST del back-end.
- **Manejo de Errores y Interceptores:** Axios permite configurar interceptores para manejar respuestas y errores de manera centralizada, mejorando la gestión de la comunicación entre el cliente y el servidor.

PDFKit

- **Generación de Documentos PDF:** PDFKit es una herramienta poderosa para la creación de documentos PDF desde el código JavaScript. Es esencial para Rent.it, ya que permite generar y enviar contratos y recibos en formato PDF, facilitando la gestión documental.
- **Personalización y Flexibilidad:** PDFKit permite personalizar el contenido y diseño de los PDFs generados, asegurando que cumplan con los estándares y necesidades de los usuarios.

NPM

- **Gestión de Dependencias:** NPM (Node Package Manager) es el gestor de paquetes para Node.js. Facilita la instalación y actualización de las librerías y frameworks utilizados en el proyecto, asegurando que todas las dependencias estén actualizadas y compatibles.

7. Manual de despliegue

Requisitos Previos:

- Java JDK 17 o superior.
- MySQL Server.
- Tener instalado GIT.
- Eclipse.
- Visual Studio Code.
- Node.js.

Implementación del proyecto

Clonar el repositorio del proyecto:

- `git clone https://github.com/saaanchez08/TFG.git`

Configuración de la Base de Datos:

- Copiar el archivo `rentit_db.txt` y crear la base de datos.

Configuración Frontend:

- Tener la última versión de Visual Studio Code instalada.
- Con el proyecto desplegado en Visual Studio Code, abrir una terminal nueva e introducir los siguientes comandos: `npm install` (nos instalará todas las dependencias necesarias).
- Instalar las librerías necesarias para que la parte de frontend funcione correctamente (Axios y PDFKit):
 - `npm install axios`
 - `npm install pdfkit`
- Por último, `npm run serve` (iniciará el proyecto por la parte frontend).

```
DONE Compiled successfully in 149ms

App running at:
- Local: http://localhost:8080/
- Network: http://192.168.0.152:8080/
```

IMPORTANTE:

La aplicación de Front deberá estar corriendo en un puerto diferente al de la aplicación de Back o no funcionará.

Configuración Backend:

- Al implementar el proyecto en Eclipse, sería recomendable realizar Update Project.
- Configurar el archivo `application.properties` (el username y password se pueden cambiar dependiendo de los mismos).
- Al igual que en DBUtils, también puedes cambiar el user y pass.
- Por último, una vez configurado todo, iniciar la aplicación en Eclipse.