# SISTEMA DE MONITORAMENTO DE EMPILHADEIRAS: CRISTOVÃO EMPILHADEIRAS LTDA

João Vitor Pereira dos Santos (1), Maurilio Erik Batista Nóbrega de Souza Medeiros (2), Vinícius Heitor Siqueira Ribeiro (3). Orientador: Prof. Me. Ranieri Marinho de Souza. (1) 6-CCOMP-00344562, (2) 6-CCOMP-00343603, (3) 6-CCOMP-00333687.

# RESUMO

O estudo analisa a criação de um sistema digital para uma empresa de locação de empilhadeiras, visando substituir os checklists físicos por um processo digital de controle de manutenção. A empresa utiliza atualmente checklists em papel, o que gera falhas no controle de dados e atrasos no diagnóstico. O novo sistema facilitará o registro e o monitoramento das informações de manutenção por colaboradores e clientes, promovendo maior precisão e acessibilidade. O público-alvo inclui colaboradores responsáveis pela manutenção e clientes que alugam os equipamentos. O projeto busca otimizar o processo de manutenção, permitindo a inserção de dados diretamente no local de operação e a consulta remota pela equipe de gestão. O objetivo é criar uma plataforma digital que registre, armazene e monitore os checklists, substituindo o processo manual existente. A metodologia inclui entrevistas com o responsável pela empresa para identificar necessidades e funcionalidades desejadas. A equipe decidiu desenvolver uma aplicação web, inicialmente, para facilitar a adoção. O desenvolvimento utiliza tecnologias como HTML, CSS, JavaScript para o front-end, e Python com Django para o back-end, com Oracle SQL como banco de dados. O sistema terá funcionalidades como cadastro de usuários, login seguro, registro de checklists, armazenamento de dados, consulta de informações e uma interface intuitiva. Os requisitos não funcionais incluem usabilidade, suporte técnico e segurança. O projeto prevê melhorias futuras, como acessibilidade remota, validação de dados, segurança aprimorada e geração de relatórios. O desenvolvimento busca garantir uma gestão mais eficiente e proativa para a empresa de locação.

**Palavras-Chave:** Checklist de Manutenção; Empilhadeiras; Sistema de Monitoramento.

# Introdução

O presente estudo analisa a criação de um sistema para uma empresa de locação de empilhadeiras, que digitaliza o processo de controle de manutenção das máquinas [1]. Tradicionalmente, a empresa utiliza checklists físicos para acompanhar as condições dos equipamentos, o que pode gerar falhas no controle de dados e atrasos no diagnóstico de problemas. A proposta, conforme a visão da empresa, digitaliza esse processo de controle de manutenção das máquinas, facilitando o registro e o monitoramento das informações de manutenção pelos colaboradores da empresa e pelos colaboradores do cliente (onde as máquinas estão alocadas), promovendo maior precisão e acessibilidade, além de facilitar o acesso para consulta, acompanhamento e validação de checklists passados.

O público-alvo deste sistema abrange os colaboradores da empresa responsáveis pela manutenção das empilhadeiras e os clientes que alugam os equipamentos e precisam acompanhar o estado das máquinas durante o período de uso. A demanda principal envolve a otimização do processo de manutenção, permitindo a inserção de dados diretamente no local onde a empilhadeira opera e a consulta desses dados de maneira remota pela equipe de gestão. O objetivo geral deste projeto consiste em criar uma plataforma digital capaz de registrar, armazenar e monitorar os checklists de manutenção das empilhadeiras, com acessos distintos para clientes e colaboradores, substituindo todo o processo manual e físico atualmente realizado. A justificativa para o desenvolvimento da aplicação reside na necessidade crescente de automatização e digitalização de processos nas empresas, reduzindo a dependência de documentos físicos e proporcionando um controle mais eficiente sobre os ativos da empresa. O aplicativo permitirá, portanto, uma gestão mais proativa, além de reduzir o risco de falhas operacionais nas máquinas.

# Material e métodos (ou metodologia)

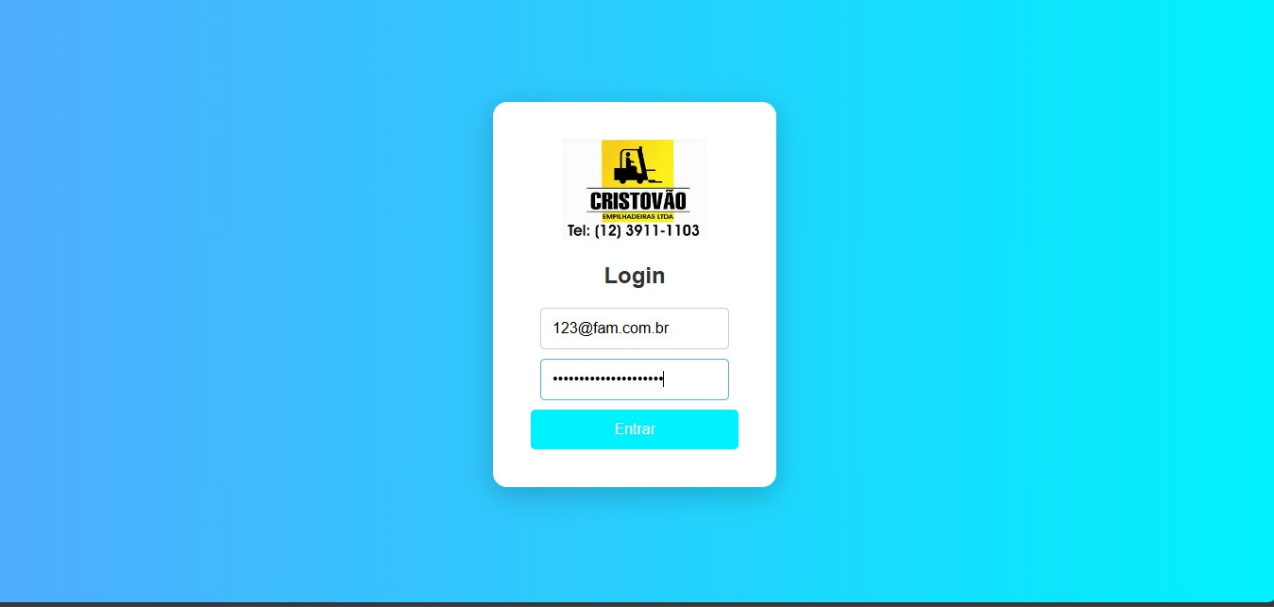
Foi realizada uma entrevista com o responsável pela empresa (Cristovão Empilhadeiras LTDA) para entender o atual fluxo de trabalho em relação ao monitoramento/controle das empilhadeiras, identificamos as necessidades e funcionalidades desejadas para aplicarmos no software que estamos desenvolvendo. Inicialmente foi decidido desenvolver uma aplicação web, por conta de o público-alvo estar acostumado com o controle (físico) por papel, pensamos em conjunto com o responsável que uma aplicação web seria mais aceita e fácil de adotar, mas nada impede de seguirmos com uma aplicação híbrida futuramente (como melhoria de projeto). A metodologia proposta busca assegurar que o aplicativo atenda às necessidades da empresa de locação de empilhadeiras, promovendo uma digitalização eficiente dos checklists de manutenção e melhorando a gestão dos ativos. Os elementos necessários para a construção do projeto são: computadores com acesso à internet, para consultas bibliográficas e para as instalações dos softwares necessários para o desenvolvimento, como por exemplo: PyCharm Community Edition, Oracle SQL Developer, Visual Studio Code etc. As funções e tarefas serão divididas em 4 itens principais: i – front-end; ii – back-end; iii – banco de dados; iv – documentação.

# Desenvolvimento

O desenvolvimento do aplicativo está sendo feito com as seguintes tecnologias: para o front-end usamos HTML, CSS e JavaScript para termos uma interface de usuário acessível e dinâmica; para o back-end usamos Python com o framework Django, assim poderemos aproveitar mais recursos, como autenticação e gerenciamento de usuários; o banco de dados utilizado será o Oracle SQL, usando a biblitoeca cx\_Oracle no Python; usamos a API RESTful.

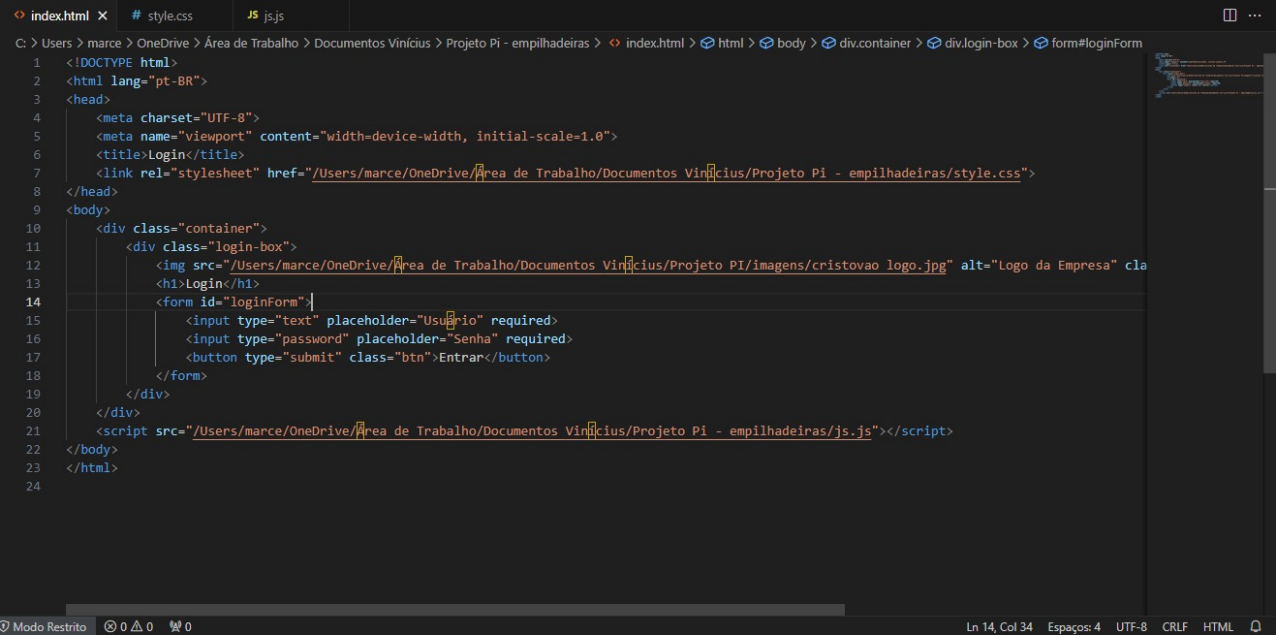
Para o front-end, foi utilizado o Visual Studio versão 1.95.1. A página de login foi desenvolvida utilizando HTML, CSS e JavaScript. O HTML foi utilizado para estruturar a página, criando os elementos necessários, como campos de entrada para o nome de usuário e senha, além do botão de envio. O CSS serviu para estilizar a página, proporcionando um design atraente. Isso ajudou a melhorar a usabilidade e a estética da interface. Já o JavaScript foi empregado para adicionar interatividade ao formulário. Assim, o uso dessas três linguagens permitiu a criação de uma página de login funcional e visualmente agradável.

**Figura 1. Página de login da aplicação.**



Fonte: Autoria própria.

**Figura 2. Código em HTML da página de login.**



Fonte: Autoria própria.

Para o desenvolvimento do banco de dados, utilizou-se o Oracle SQL Developer Version 24.3.0.284.2209 - October 17, 2024 [2]. Para o planejamento e criação do DER e MER, utilizou-se o Oracle Developer Data Modeler Version 24.3.0.240.1210 - October 17, 2024]. Ambos requerem o JDK (Java Development Kit) 17.

O banco de dados é relacional [3] e foi identificado inicialmente a necessidade de 6 tabelas, podendo novas serem criadas, se necessário. Abaixo na figura 3, o DER/MER do banco de dados relacional do projeto [4]. Não foi possível fazer a validação com o responsável da empresa até a entrega do artigo por indisponibilidade de agenda dele.

**Figura 3. DER do banco de dados.**

Diagrama

Descrição gerada automaticamente com confiança média

Fonte: Autoria própria.

Após identificar e descrever as entidades e os atributos foram identificados os relacionamentos existentes abaixo:

Relacionamento entre Clientes e Local: um cliente pode ter vários locais (locais seriam galpões ou armazéns) e um local pode ser apenas de um cliente.

Relacionamento Local e Empilhadeiras: um local pode ter várias empilhadeiras e uma empilhadeira pode estar apenas em um local por vez.

Relacionamento Empilhadeiras e Checklist: uma empilhadeira pode estar (e estará) em vários checklists, mas um checklist terá apenas uma empilhadeira.

Relacionamento Usuarios e Checklist: um usuário poderá criar vários checklists, mas um checklist será feito por apenas um usuário.

Relacionamento Itens\_Checklist e Checklist: os itens do checklist estarão em vários checklists.

Os requisitos funcionais abaixo visam atender às necessidades da empresa de locação de empilhadeiras, para uma gestão eficiente e eficaz das manutenções das máquinas.

Cadastro de Usuários: o sistema deve permitir o cadastro de usuários, diferenciando entre colaboradores da empresa e clientes.

Login e Autenticação: o sistema deve possibilitar login seguro para usuários cadastrados, garantindo acesso controlado às funcionalidades.

Registro de Checklists de Manutenção: o aplicativo deve permitir que colaboradores e clientes registrem informações sobre a manutenção das empilhadeiras diretamente no local de operação.

Armazenamento de Dados: os dados registrados nos checklists devem ser armazenados de forma centralizada e acessível pela equipe de gestão.

Consulta de Dados: o sistema deve permitir que os usuários consultem checklists passados e o estado atual das empilhadeiras em tempo real.

Interface Intuitiva: o aplicativo deve possuir uma interface amigável e intuitiva para facilitar o uso por colaboradores e clientes.

Os requisitos não funcionais abaixo são essenciais para garantir que o sistema não apenas funcione conforme esperado, mas também ofereça uma experiência de usuário satisfatória e confiável ao longo do tempo.

Usabilidade: a interface do aplicativo deve ser intuitiva e fácil de usar, permitindo que usuários com diferentes níveis de experiência tecnológica possam operar o sistema sem dificuldades.

Suporte e Manutenção: o sistema deve contar com um plano de suporte técnico para resolver problemas e fornecer assistência aos usuários, com tempos de resposta bem definidos.

Descrições do Diagrama de Caso de Uso:

Os atores identificados e os casos de uso incluem:

Atores: Colaborador: Usuário que registra e monitora manutenção; Cliente: Usuário que registra informações e consulta o estado das empilhadeiras; Sistema: O aplicativo em si.

Casos de Uso: Cadastrar Usuário; O colaborador ou cliente pode se cadastrar no sistema; Login; Usuário se autentica no sistema; Registrar Checklist de Manutenção; Colaborador e cliente podem registrar informações sobre a manutenção; Consultar Checklists; Usuário pode consultar checklists passados e o estado atual das empilhadeiras; Gerar Relatórios; Usuário pode gerar relatórios de manutenção; Receber Alertas de Manutenção; Usuário recebe notificações sobre manutenções necessárias; Visualizar Dados; Usuário pode visualizar dados em tempo real.

Descrições do Diagrama de Classe:

Classes:

Usuário – Atributos: id, nome, tipo (Colaborador ou Cliente), email, senha

Métodos: cadastrar(), login(), gerarRelatório()

Cliente – Atributos: id, nome

Métodos: registrarChecklist()

Colaborador – Atributos: id, nome

Métodos: registrarChecklist()

Local – Atributos: id, endereço

Métodos: adicionarEmpilhadeira()

Empilhadeira – Atributos: id, modelo, status

Métodos: vincularChecklist()

Checklist – Atributos: id, data, idEmpilhadeira, idUsuario

Métodos: adicionarItem(), visualizar()

ItemChecklist – Atributos: id, descrição, status

Métodos: atualizarStatus()

**Figura 4. QR Code para acesso ao repositório Github do projeto.**

Código QR

Descrição gerada automaticamente

Fonte: Autoria própria.

# Considerações finais

O desenvolvimento deste aplicativo para a digitalização dos checklists de manutenção de empilhadeiras trará avanços significativos para a empresa de locação, que tradicionalmente depende de processos físicos. A ideia de aumentar a eficiência na gestão de manutenção e facilitar o acesso às informações, irá ser vista com sucesso para a empresa e seus clientes. O contato próximo com o responsável facilita bastante a troca de feedback para entregar a aplicação o mais próximo possível do que ele imaginou.

Para melhorias futuras, são consideradas as seguintes funcionalidades:

Acessibilidade Remota: para permitir que os clientes acessem as informações de manutenção remotamente, facilitando o acompanhamento do estado das máquinas.

Validação de Dados: incluir a funcionalidade que permita a validação dos dados inseridos nos checklists, garantindo a precisão das informações.

Segurança de Dados: implementar medidas de segurança para proteger os dados armazenados, assegurando a privacidade e integridade das informações.

Alertas de Manutenção: enviar alertas para os usuários sobre manutenções necessárias ou prazos que estão se aproximando.

Geração de Relatórios: incluir a funcionalidade de gerar relatórios sobre a manutenção das empilhadeiras, apresentando dados relevantes para a gestão.

Treinamento e Suporte: oferecer um suporte técnico e treinamento para os usuários, garantindo que saibam utilizar todas as funcionalidades iniciais e futuras do sistema.

Para melhorias futuras não funcionais, são consideradas:

Acessibilidade e Compatibilidade: ser acessível em diferentes dispositivos, incluindo smartphones, tablets e desktops, garantindo uma experiência consistente em todas as plataformas. Ser compatível com as principais plataformas de sistemas operacionais (iOS, Android, Windows) e navegadores modernos (Chrome, Firefox, Safari). Dependerá se o responsável decidirá em usar migrar para uma aplicação híbrida.

Segurança: garantir a proteção dos dados por meio de criptografia, autenticação de dois fatores e controle de acesso baseado em funções, prevenindo acessos não autorizados.

Compatibilidade: ser compatível com as principais plataformas de sistemas operacionais (iOS, Android, Windows) e navegadores modernos (Chrome, Firefox, Safari).

Backup e Recuperação: incluir funcionalidade de backup automático e recuperação de dados para proteger contra perdas de informação.

Documentação: ter uma documentação completa do sistema, incluindo manuais de usuário, guias de instalação e documentação técnica para desenvolvedores.

# Referências

[1] GESTÃO DA MANUTENÇÃO PREVENTIVA DE EMPILHADEIRAS EM UMA EMPRESA DE PEQUENO PORTE UTILIZANDO UMA FERRAMENTA IOT INTEGRADA A UM SOFTWARE. Disponível em: https://www.researchgate.net/publication/355040572\_GESTAO\_DA\_MANUTENCAO\_PREVENTIVA\_DE\_EMPILHADEIRAS\_EM\_UMA\_EMPRESA\_DE\_PEQUENO\_PORTE\_UTILIZANDO\_UMA\_FERRAMENTA\_IOT\_INTEGRADA\_A\_UM\_SOFTWARE

[2] ORACLE. Oracle SQL Developer Documentation. Disponível em: https://docs.oracle.com/en/database/oracle/sql-developer/index.html

[3] OLIVERIA, DANIELLE. MER e DER: Definições, Banco de Dados e Exemplos. Disponível em: https://www.alura.com.br/artigos/mer-e-der-funcoes

[4] MARIN, CAROLINE. Modelo Relacional – Oracle SQL Developer Data Modeler 3.3 PT-BR. Disponível em: https://youtu.be/A9vMC\_6IxRo?si=O0\_VJ7HMr7gIHpmr