

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

VINÍCIUS BRAUN SCHEFFEL

**Machine Rental Management - Sistema de
Gerenciamento de Aluguel de Máquinas
utilizando QR Code como integração**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em Ciência
da Computação

Orientador: Prof. Dr. Marcelo Primenta

Porto Alegre
2022

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões Mendes

Vice-Reitora: Prof^a. Patricia Helena Lucas Pranke

Pró-Reitoria de Ensino (Graduação e Pós-Graduação): Prof^a. Cíntia Inês Boll

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência da Computação: Prof. Marcelo Walter

Bibliotecário-Chefe do Instituto de Informática: Alexander Borges Ribeiro

RESUMO

O presente trabalho é o desenvolvimento e a implantação de um software de gerenciamento de aluguéis de máquinas para uma empresa real. O sistema possui suporte para gerar contratos e faturas de maneira prática, além de proporcionar um controle simples do estado das máquinas da empresa. O diferencial do sistema é a integração dos processos físicos de saída e chegada das máquinas no estoque da empresa com o banco de dados. Essa funcionalidade é realizada através de QR codes impressos nas máquinas e com o aplicativo de celular que é capaz de ler o código e atualizar o estado da máquina no sistema.

Palavras-chave: Aluguél de máquinas. gerenciamento. QR Code. ERP.

Machine Rental Management - Machine Rental Management System using QR Code as integration

ABSTRACT

The present work is the development and implementation of a machine rental management software for a real company. The system has support for generating contracts and invoices in a practical way, in addition to providing a simple control of the state of the company's machines. The system's differential is the integration of the physical processes of departure and arrival of the machines in the company's stock with the database. This functionality is carried out through QR codes printed on the machines and with the mobile application that is able to read the code and update the status of the machine in the system.

Keywords: machine rental. management. QR Code. ERP.

LISTA DE FIGURAS

Figura 2.1 Exemplo de um compressor.....	13
Figura 3.1 Painel do website	19
Figura 3.2 Tela de faturamento	19
Figura 3.3 Tela de inventário.....	20
Figura 3.4 Tela de eventos.....	20
Figura 3.5 Exemplo do código QR de uma máquina	21
Figura 3.6 Tela principal do aplicativo mobile.....	21
Figura 3.7 Tela que exibe as informações da máquina	22
Figura 3.8 Tela de registro de eventos.....	22
Figura 3.9 Tela de operação concluída com sucesso.....	23
Figura 3.10 Visão do frontend do sistema	30
Figura 3.11 Visão do backend do sistema	31
Figura 3.12 Visão geral do sistema.....	32
Figura 3.13 Serviços de infraestrutura do sistema	33
Figura 4.1 Trello do Trabalho	35
Figura 4.2 Diagrama ER	37
Figura 4.3 Tela de Inventário	39
Figura 4.4 Criar locação.....	40
Figura 4.5 Exemplo de um Contrato	41
Figura 4.6 Gráfico de distribuição por sexo	43
Figura 4.7 Gráfico de distribuição por faixa etária.....	43
Figura 4.8 Gráfico de distribuição do nível de escolaridade	43
Figura 4.9 Gráfico das respostas para a pergunta “O sistema gma-admin relaciona-se a problemas que me deparo nas minhas atividades profissionais.”	44
Figura 4.10 Gráfico das respostas para a pergunta “As seguintes funcionalidades resolvem problemas que aparecem frequentemente para mim.”	45
Figura 4.11 Gráfico das respostas para a pergunta “Sinto falta no sistema de uma funcionalidade para resolver um problema específico”	45
Figura 4.12 Gráfico das respostas para a pergunta “O sistema é útil para as minhas atividades.”	46
Figura 4.13 Gráfico das respostas para a pergunta “Eu usaria o sistema em minhas atividades.”	46
Figura 4.14 Gráfico das respostas para a pergunta “Eu recomendaria o sistema para outros profissionais.”	46

LISTA DE TABELAS

Tabela 4.1 Tabela mostrando as divisões das questões do questionário de avaliação.....42

LISTA DE ABREVIATURAS E SIGLAS

ERP Enterprise resource planning

MRM Machine Rental Management

SUMÁRIO

1 INTRODUÇÃO	9
1.1 Motivação	9
1.2 Objetivo	9
1.3 Estrutura do Trabalho	10
2 CONCEITOS, TECNOLOGIAS E TRABALHOS RELACIONADOS	11
2.1 Gerenciamento Manual de Locações e Conceitos Fundamentais	11
2.1.1 Locação	11
2.1.2 Contrato.....	12
2.1.3 Fatura	12
2.1.4 Produto.....	13
2.2 Tecnologias	14
2.2.1 JavaScript	14
2.2.2 Angular	15
2.2.3 Android - Java.....	15
2.2.4 Node.js	15
2.2.5 Express.....	16
2.3 Trabalhos relacionados	16
3 UM SISTEMA INFORMATIZADO PARA GERENCIAR LOCAÇÕES	18
3.1 Visão Geral	18
3.1.1 Website.....	18
3.1.2 Aplicativo Mobile	20
3.1.3 Servidor.....	23
3.2 Requisitos da aplicação	28
3.2.1 Requisitos Não-Funcionais	29
3.2.2 Requisitos Funcionais	29
3.3 Arquitetura	30
3.3.1 Frontend	30
3.3.2 Backend.....	31
3.3.3 Visão geral da arquitetura	32
3.3.4 Outros serviços de infraestrutura	33
4 IMPLEMENTAÇÃO E AVALIAÇÃO	35
4.1 Implementação	35
4.1.1 Primeiros passos.....	36
4.1.2 Fase de requisitos	36
4.1.3 Fase de projeto	36
4.1.4 Fase de implementação	37
4.1.5 Fase de testes.....	38
4.1.6 Fase de produção.....	38
4.1.7 Fase de manutenção	38
4.2 Exemplo de caso de uso	38
4.3 Avaliação	41
4.3.1 Perfil dos Participantes.....	42
4.3.2 Análise dos Resultados	44
5 CONCLUSÃO	47
5.1 Resumo de resultados	47
5.2 Limitações e melhorias	47
REFERÊNCIAS	49

1 INTRODUÇÃO

Este trabalho teve o seu início pela percepção de que uma empresa familiar conhecida, chamada GMA Compressores e Geradores, estava passando por dificuldades reais de negócio devido a falta de tecnologias para auxiliar nos processos da mesma.

A GMA é uma empresa localizada em Campo Bom, Rio Grande do Sul, que está em grande expansão, fornecendo serviços de aluguel de compressores e geradores. A empresa possui um grande galpão onde mantém as máquinas, que podem estar em manutenção, disponíveis para aluguel ou prestes a sair para uma locação.

Existe um grande fluxo de entrada e saída de máquinas, que retornam de clientes após o término do contrato de aluguel, e saem para novos clientes assim que um novo contrato de aluguel é realizado.

Este texto vai descrever o desenvolvimento, as decisões, argumentos e fundamentos relacionados a essa questão, como também da solução proposta. O sistema desenvolvido já está em uso, mas se encontra em fase de manutenção e evolução. Isso significa que novas funcionalidades ainda podem ser implementadas e problemas corrigidos.

1.1 Motivação

Tendo-se ciência de que a empresa possuía problemas em diversos setores, como organização do estoque e gerenciamento dos aluguéis das máquinas, contratos e faturamentos, viu-se a oportunidade de fornecer uma solução que unificasse as atividades exercidas de ponta a ponta, desde aquelas dos técnicos de manutenção até as da equipe de administração e gerência das locações e finanças.

1.2 Objetivo

Com base na informatização das atividades exercidas pelos funcionários da empresa, o sistema é alimentado, e a informação acerca das máquinas e dos aluguéis está integrada e disponível conforme a necessidade de cada funcionário.

Dessa maneira, uma série de otimizações são possíveis. Primeiro, o tempo para que a informação acerca do estado das máquinas entre técnicos e gerentes é minimizado. Segundo, os técnicos podem consultar informações fidedignas acerca das máquinas, que

estão atualizadas no sistema. Terceiro, a equipe gerencial pode contar com um sistema confiável, prático e centralizado para as suas atividades gerenciais e financeiras.

Este trabalho tem como objetivo descrever o sistema proposto como solução para os problemas descritos acima.

1.3 Estrutura do Trabalho

Este trabalho é dividido em 5 capítulos. O capítulo 2 apresenta os conceitos de gerenciamento manual de locações e tecnologias utilizadas para o desenvolvimento deste trabalho. O capítulo 3 descreve o sistema informatizado para o gerenciamento das locações, apresentando uma visão geral, os requisitos e a arquitetura do sistema. O capítulo 4 descreve a implementação do sistema, em suas diversas fases, e a avaliação do mesmo. O capítulo 5 apresenta as conclusões do trabalho e os resultados alcançados, além das limitações do trabalho e sugestões para adições futuras.

2 CONCEITOS, TECNOLOGIAS E TRABALHOS RELACIONADOS

Este capítulo visa introduzir conceitos básicos da locação de produtos, mostrando como são utilizados no gerenciamento manual de locações, as tecnologias utilizadas no trabalho, e também descrever alguns trabalhos relacionados.

2.1 Gerenciamento Manual de Locações e Conceitos Fundamentais

Como este trabalho visa atender a demanda de uma empresa real de locação, é necessário entender os conceitos fundamentais na visão da empresa, para então poder utilizá-los na construção do sistema. Além disso, é necessário entender como os conceitos se encaixam dentro da empresa, no seu processo manual de locações, para que então se possa propor um sistema informatizado. Antes da implementação do sistema informatizado, a atividade dos diversos setores da empresa era realizada de forma manual, com sistemas antiquados e não confiáveis encontrados na internet, e com diversas planilhas.

2.1.1 Locação

Uma locação é quando a empresa cede à outra parte, que é o cliente que eles atendem, o uso dos produtos da empresa, através de um contrato de comodato, por um determinado prazo, mediante o pagamento de uma certa quantia. As locações funcionam com períodos pré-definidos. A primeira opção é diária, com o mínimo de 5 dias, que renova-se a cada 5 dias, e possui um valor mais alto. A segunda opção é quinzenal, que renova-se a cada 15 dias, e o valor é menor em relação à primeira opção. A terceira e última opção é mensal, que renova-se a cada 30 dias, e é a melhor relação custo-benefício. A empresa incentiva os clientes a optarem por essa opção.

Outra variável que a empresa define com o cliente é o regime de uso diário do equipamento. Na locação de compressores e secadores, que são tipos de produtos que a empresa oferece, há menos opções de regime, normalmente 8 horas diárias ou 24 horas. Por outro lado, no caso dos geradores, a variabilidade é maior, podendo ser 4 horas diárias, 20 horas ininterruptas, ou somente para uso em stand-by, onde a máquina liga em caso de falta de energia.

A empresa também alinha o tipo de frete, podendo ser terceirizado ou próprio do

locatário, que busca o produto na sede da empresa. Quando é terceirizado, a empresa contrata um frete e coloca o custo adicional do frete na locação.

Assim que a máquina sai da sede da empresa, o técnico, na maioria das vezes, desloca-se para instalar o equipamento. Outras vezes, o próprio cliente realiza a instalação, seguindo as instruções dos técnicos.

A locação, que vincula uma ou mais máquinas com um cliente através de um contrato, é organizada e mantida através de diversas planilhas. Por não ter todas as informações centralizadas, a equipe de gerenciamento perde tempo para processar tanto novos como antigos contratos.

Parte das tarefas da equipe de gerenciamento é produzir um novo contrato todavez que um novo cliente aluga uma máquina. Hoje, esse contrato é escrito manualmente em um arquivo de texto.

2.1.2 Contrato

O contrato de locação registra de modo formal a negociação entre o locador e o locatário. Ele é um guia para as duas partes, onde consta todas as informações da locação, inclusive acessórios que venham a ser solicitados na locação (cabos, mangueiras e conexões). A cada renovação da locação, quando o cliente quer alugar o produto por mais tempo, é gerado um novo aditivo do contrato.

As máquinas são enviadas com nota de remessa para comodato referenciando o número do contrato. Nenhum equipamento é deslocado da sede da empresa sem estes dois documentos, contrato e nota de remessa. Dentre os motivos, o principal é por ser determinação fiscal, onde, em caso de falta, pode acarretar em multa e até apreensão da máquina.

2.1.3 Fatura

A fatura é um documento de cobrança que possui valor fiscal como uma darfe, contendo informações importantes sobre o produto e serviço negociado. Assim, a fatura deve conter a descrição da máquina alugada, o valor total do aluguel, a data de vencimento e número da fatura e contrato. Os números das faturas precisam seguir uma ordem que se adeque a data de emissão do documento.

Assim, outra tarefa necessária é a de produzir faturas, uma vez por mês, para cada locação ativa, e enviá-las aos clientes a fim de cobrar o valor do aluguel. Hoje, essa fatura é escrita manualmente, assim como o contrato. As faturas precisam ser enviadas no dia certo, conforme combinado com o cliente, e, também, quando vencidas, devem ser cobradas uma segunda vez. Por isso, a cada dia, a equipe de gerenciamento necessita verificar em suas planilhas quais faturas devem ser enviadas, e quais estão com status de vencidas, o que é dispendioso, complexo e propenso a erros.

2.1.4 Produto

Os produtos da empresa são geralmente máquinas, compressores e geradores, relacionados a geração de energia, como mostra a figura 2.1. São esses produtos que são locados, e não há opção de compra. São máquinas grandes, e normalmente transportadas por um caminhão. Por exemplo, o compressor de ar a parafuso elétrico é uma máquina geradora de ar comprimido, que é um tipo de energia. O gerador de energia é um dispositivo utilizado para a conversão da energia mecânica, química, ou outra forma de energia, em energia elétrica.

Figura 2.1: Exemplo de um compressor



Fonte: O Autor

As informações do produto são salvas em um primeiro sistema antiquado e pouco

confiável. Para manter as informações do produto atualizadas, este sistema requer ações dispendiosas. Como os produtos estão em contato com os técnicos, mas o sistema somente é acessado pelos gerentes, é necessário que os primeiros comuniquem pessoalmente os segundos. Com a informação acerca do novo estado da máquina, os gerentes agora podem atualizá-la no sistema.

Por exemplo, quando uma máquina retorna de um cliente que a alugou, um técnico de manutenção recebe essa máquina no pátio da empresa. Depois disso, ele precisa ir até a equipe de gerenciamento, que fica em outro local, para comunicar que esta máquina agora voltou ao pátio e está disponível. Quando o técnico não possui tempo de ir comunicar essa informação, alguém da equipe de gerenciamento precisa percorrer o pátio procurando quais máquinas voltaram, quais estão disponíveis.

Assim, quando um cliente liga para saber se a empresa possui uma máquina com certas especificações pronta para ser alugada, é necessário um tempo de espera até que essa informação esteja disponível. Além de ser inconveniente, nem sempre o cliente pode esperar, o que leva a perda de uma locação.

2.2 Tecnologias

Nesta seção, vou descrever as tecnologias utilizadas no trabalho, fornecendo as razões de terem sido escolhidas quando em comparação com outras.

2.2.1 JavaScript

O JavaScript (MOZILLA, 2022) é uma linguagem de programação criada pela European Computer Manufacturer's Association (ECMA). Ele é utilizado em páginas web que dependem de certo dinamismo, não exibindo somente itens estáticos. Por exemplo, quando é necessário que aconteça alguma ação após clicar em um botão, o script escrito em JavaScript será responsável por isso. O JavaScript é uma linguagem orientada a objetos, paradigma de programação fundamental nos dias de hoje por uma série de vantagens.

Pode-se dividir uma página Web em três partes principais. A primeira parte é o HTML, uma linguagem de marcação utilizada na construção de páginas na Web, que descreve a informação contida na página, isto é, seu conteúdo. A segunda parte é o CSS,

que é um mecanismo para adicionar estilo a um documento web, descrevendo, então, como o conteúdo da primeira parte será exibido. A terceira e última parte é o JavaScript, responsável pelo comportamento da página. A terceira parte, o JavaScript, manipula as duas primeiras, o HTML e o CSS, proporcionando movimentos dinâmicos à página (TABLELESS, 2022).

2.2.2 Angular

Pensando em tecnologias para a construção de aplicações web, Angular é sem dúvidas uma das mais evidentes. Angular (ANGULAR, 2022) é uma plataforma de desenvolvimento que utiliza Typescript como linguagem. Typescript é, no fundo, JavaScript, mas com camadas de açúcar sintático (TYPESCRIPTLANG, 2022). O Angular é escrito em código aberto, e conta com um repositório bem movimentado, indicando a preferência do mercado em utilizá-lo e a sua grande popularidade. Além disso, toda a experiência profissional do autor com o desenvolvimento de aplicações web é utilizando Angular.

2.2.3 Android - Java

Os celulares dos técnicos da empresa rodam Android (ANDROID, 2022), plataforma operacional para dispositivos móveis. É possível desenvolver aplicativos Android utilizando Java ou Kotlin. Por isso, para a construção do aplicativo Mobile, utilizou-se Java. Kotlin também seria uma boa opção, mas o autor não possuía experiência com essa linguagem.

2.2.4 Node.js

Node.js (NODEJS, 2022) é uma tecnologia multiplataforma que permite ao desenvolvedor criar diversos tipos de sistemas do lado do servidor (backend) em JavaScript. Uma das razões principais para a adoção desta tecnologia é que, quando comparada com outras tecnologias, Node.js conta com grande quantidade de pacotes reutilizáveis. O chamado Gerenciador de Pacotes do Node (NPM, na sigla em inglês), fornece uma maneira rápida e simples de gerenciar esses pacotes. Essa é uma vantagem significativa neste caso,

já que o sistema precisa realizar algumas operações específicas como: renderizar arquivos PDF para contratos e faturas; conexão com banco de dados; RestAPI com autenticação para que o frontend possa interagir com o servidor em segurança. Através dos pacotes corretos, essas tarefas poderiam ser desenvolvidas mais rapidamente, com a segurança de não estar reinventando algo que já existe, qualidade compatível com os princípios da Orientação à Objetos.

Outra vantagem considerável é que o autor já possuía grande experiência com Node.js, dependendo apenas do aprendizado dos pacotes a serem utilizados.

2.2.5 Express

O pacote Node mais importante que foi utilizado é o Express (MOZILLA, 2022), um framework bastante popular, capaz de gerenciar requisições HTTP. Como o sistema não fornece, através de sua API, operações HTTP muito complexas, mas, em geral, operações simples de criação, visualização, atualização e deleção dos dados, o caráter minimalista do Express é bastante vantajoso. Investe-se pouco tempo escrevendo o código, e obtém-se com poucas linhas de código o comportamento desejado. Através do framework, poucas linhas bem escritas pelo desenvolvedor, se tornam um novo processo disponível na API. Este fator também reflete na hora da manutenção, já que um código coeso e simples é mais fácil de ser mantido. A API do backend, escrita utilizando o Express, segue os padrões arquiteturais REST (REST, 2022) e, portanto, pode-se dizer que é uma REST API.

2.3 Trabalhos relacionados

Existem hoje no mercado algumas outras soluções para empresas de locação. Uma delas, por exemplo, é o Actloc (ACTLOC, 2022), que era anteriormente utilizado pela empresa. O Actloc é um sistema de gerenciamento de locação genérico, e, por ser genérico, possui muitas funcionalidades a mais do que a empresa precisa. Esse fato causa uma diminuição na usabilidade do sistema por parte da equipe de administração da empresa, já que é preciso interagir com mais opções, campos, formulários e telas do que o necessário.

Além disso, esses sistemas genéricos não possuem funcionalidades específicas para abarcar os casos de usos mais próprios da empresa em questão. Um exemplo de caso

de uso importante é a atualização do estado das máquinas no sistema. Utilizando sistemas de locação genéricos, a atualização do estado das máquinas depende de, no mínimo, uma comunicação a mais. Quando uma máquina muda de estado, o técnico precisa informar a equipe gerencial acerca da mudança, porque é a equipe gerencial que atualiza as informações no sistema. Através do sistema específico desenvolvido nesse trabalho, o técnico conta com um aplicativo capaz de atualizar o estado da máquina sem a necessidade de informar a equipe gerencial.

Outra questão importante é a da confiabilidade dos sistemas genéricos. O sistema que a empresa utilizava anteriormente, o Actloc, não era confiável do ponto de vista da empresa. Aconteceu, mais de uma vez, o sumiço de dados inseridos no sistema, e problemas na hora de atualizar entidades. Além disso, o suporte da equipe desse sistema era mais lenta e as vezes inexistente. Frente a essas dificuldades, o sistema proposto é mais confiável por contar com uma série de testes de unidade, integração e de uso, proporcionando um alto nível de qualidade, e garantindo a sua confiabilidade. Em caso de problemas, dada a qualidade do software e o suporte próximo do cliente, a correção ocorre de forma ágil.

Dessa maneira, uma das motivações para este trabalho é o desenvolvimento de um produto customizado, próprio para a empresa em questão, englobando todos os casos de uso necessários, sem poluir o sistema com opções e operações supérfluas. Além disso, é preciso lembrar que, com o tempo, novas necessidades surgem por parte da empresa, e funcionalidades que antes não estavam em evidência, agora recebem importância. Uma vez que esteja em funcionamento, o produto customizado recebe manutenção e incrementos, evoluindo conforme as necessidades da empresa.

Além do propósito de solucionar o problema da empresa, existem outras motivações intelectuais para o trabalho. Por exemplo, este trabalho fornece uma ampliação do domínio do problema. Isso significa entender melhor e com maior profundidade os termos e processos envolvidos no serviço de locação, com base em uma empresa real. Também, este trabalho fornece entendimento para o processo de criação da solução, desde a comunicação com o cliente, para entender os requerimentos e casos de uso, até a implementação da mesma, integrando o conhecimento do domínio do problema com o desenvolvimento da aplicação.

3 UM SISTEMA INFORMATIZADO PARA GERENCIAR LOCAÇÕES

Frente às diversas desvantagens de gerenciar as locações manualmente, em diversas planilhas, e do controle do estoque de máquinas manual, cabe um sistema informatizado para auxiliar no processo de negócio da empresa.

3.1 Visão Geral

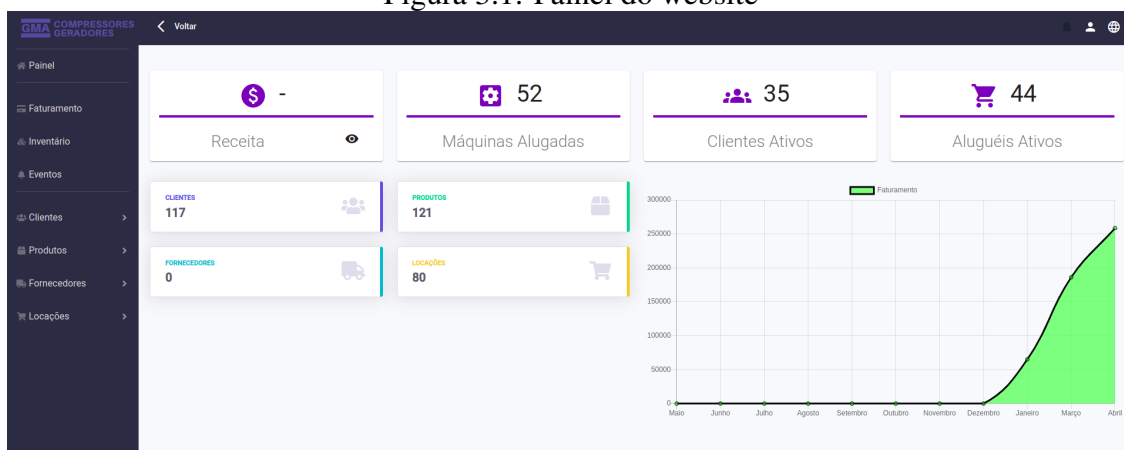
O sistema conta com um website, utilizado principalmente pela equipe de gerenciamento e financeira, um aplicativo mobile, utilizado principalmente pelos técnicos de manutenção, e um servidor backend, que serve às duas aplicações e onde os dados se encontram centralizados.

3.1.1 Website

O website gerencia a criação, atualização, visualização e deleção de todas as entidades do sistema. As principais entidades são os clientes, os produtos e as locações. É no website que os contratos de locação são gerenciados, através da entidade locação. A locação associa clientes com produtos por um determinado período de locação, e é a partir dela que contratos e faturas são geradas. Os clientes podem ser pessoa física ou jurídica, e os produtos podem ser de diversos tipos, como geradores, compressores e secadores. Dentro de cada locação, é possível acessar os seus respectivos aditivos, que são criados quando um cliente quer renovar uma locação.

O painel do website, conforme a figura 3.1, exibe informações e métricas importantes para a equipe, como faturamento, quantidade de máquinas alugadas e contratos de aluguéis ativos, e também um gráfico de faturamento dos últimos 12 meses. Também fornece botões de navegação rápida para a criação das entidades do sistema, e a quantidade de entidades já criadas. No canto esquerdo, pode-se observar um menu que fornece uma maneira rápida de acessar qualquer área do site, como voltar para o painel, acessar a tela de faturamento, inventário, eventos, e também as telas das entidades, como clientes, produtos, fornecedores e locações. Esse menu permanece o mesmo em todas as telas do sistema, isto é, sempre está disponível para ser usado pelo usuário. Existe, também, no canto superior esquerdo, um botão para voltar de página.

Figura 3.1: Painel do website



Fonte: O Autor

Existem três telas principais, responsáveis por auxiliar a equipe de gerenciamento. A primeira tela, a de faturamento, auxilia a tarefa de faturar locações, e cobrar clientes com faturas atrasadas, porque centraliza em uma tela todas as locações a serem faturadas, e inclui uma filtragem pelo status da fatura. A figura 3.2 mostra como é a tela de faturamento.

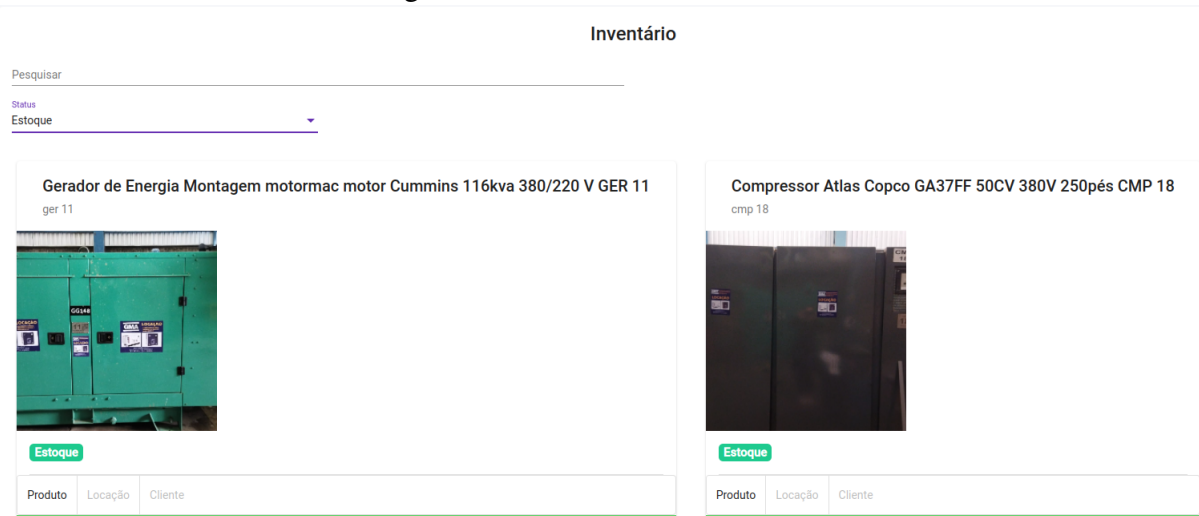
Figura 3.2: Tela de faturamento

Faturamento											
Pesquisar											
Status da Fatura											
Faturado											
Ações	Início	Término	Vencimento	Fatura ↓	Período	Valor Total	Status da Fatura	Contrato	Número do Aditivo	Cliente	Progresso
☰	07/09/2022	22/09/2022	20/09/2022	1374	15	R\$1.800	Faturado	441	1	ICL BENEFICIAMENTO DE COUROS LTDA	0%
☰	15/08/2022	14/09/2022	23/09/2022	1373	30	R\$6.800	Faturado	449	0	ESTRUTURAS METALICAS E SISTEMAS CONSTRUTIVOS DEMUTH LTDA.	60%
☰	07/09/2022	07/10/2022	26/09/2022	1372	30	R\$9.936	Faturado	443	1	TRIVIMUM USINAGEM INDUSTRIAL LTDA	0%
☰	05/09/2022	05/10/2022	23/09/2022	1371	30	R\$3.650	Faturado	377	6	SEMEATO SA INDUSTRIA E COMERCIO	0%
☰	01/07/2022	26/08/2022	20/09/2022	1370	56	R\$77.300	Faturado	457	0	INTERCEMENT BRASIL S.A.	100%

Fonte: O Autor

Outra tela importante é a de inventário, porque exibe todas as máquinas do sistema, com a sua foto, e o estado em que aquela máquina se encontra. Se algum usuário quiser saber quais máquinas estão no pátio neste momento, basta filtrar por elas, como na figura 3.3.

Figura 3.3: Tela de inventário



Fonte: O Autor

A figura 3.4 mostra a terceira tela, a de eventos. Essa tela auxilia a equipe a saber o que aconteceu recentemente com as máquinas, porque exibe um evento para cada mudança de estado que uma máquina sofre, associado com o horário dessa mudança.

Figura 3.4: Tela de eventos

Data	Item	Ação
02/09/2022 - 9:49	UNIDADE CHILLER E015s- motor: 1X5TR cap: 15.000 KCAL/H, Tensão: 220 / 380 / 440 v trifásica -temp. trabalho: 10°C, temp. condensação : 43°C, vazão: 17,9m³/h SEC 30 - SEC 30	Foi alugado.
02/09/2022 - 9:45	UNIDADE CHILLER E015s- motor: 1X5TR cap: 15.000 KCAL/H, Tensão: 220 / 380 / 440 v trifásica -temp. trabalho: 10°C, temp. condensação : 43°C, vazão: 17,9m³/h SEC 30 - SEC 30	Retornou ao estoque.
01/09/2022 - 16:54	Gerador de Energia Montagem motormac motor Cummins 116kva 380/220 V GER 11 - ger 11	Retornou ao estoque.
01/09/2022 - 16:24	Gerador de energia 260kva - GER 08 - ger 08	Retornou ao estoque.
01/09/2022 - 8:44	Compressor Atlas Copco GA90FF 125CV 8.9bar N.série 061341 630pés CMP 98 - cmp 98	Foi alugado.

Fonte: O Autor

3.1.2 Aplicativo Mobile

Através do aplicativo mobile, os usuários, normalmente os técnicos da empresa, são capazes de visualizar informações de uma máquina e criar eventos de mudança de estado para a máquina. Para visualizar as informações de uma máquina, é possível escanear o Código QR presente em uma placa metálica colada na máquina, como mostra a figura 3.5, ou informar o código da máquina manualmente.

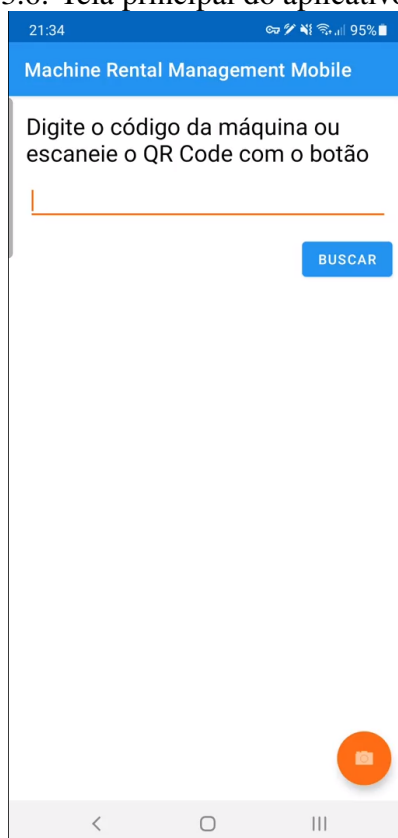
Figura 3.5: Exemplo do código QR de uma máquina



Fonte: O Autor

A tela principal do aplicativo, conforme mostra a figura 3.6, permite ao técnico clicar no botão da câmera, para escanear o QR Code da máquina, e também digitar o código da máquina manualmente.

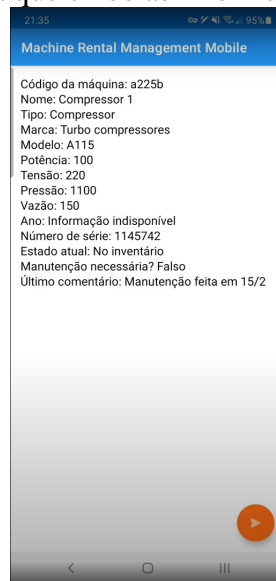
Figura 3.6: Tela principal do aplicativo mobile



Fonte: O Autor

Após ter escaneado o código QR de uma máquina, ou ter digitado o código da máquina manualmente, o técnico será encaminhado para a segunda tela do aplicativo, que exibe uma série de informações técnicas a respeito da máquina. A figura 3.7 mostra um exemplo dessa tela.

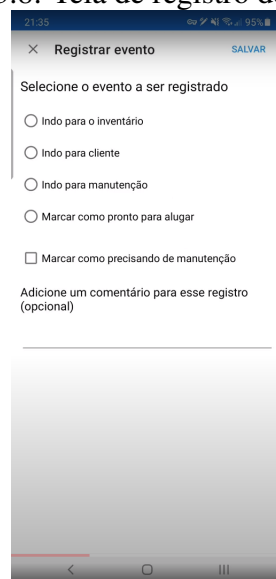
Figura 3.7: Tela que exibe as informações da máquina



Fonte: O Autor

A partir dessa tela, o técnico pode escolher voltar para a tela principal, para interagir com alguma outra máquina, ou clicar no botão laranja, para alterar o estado da máquina atual através de um evento. Uma vez que queira mudar o estado da máquina atual e tenha clicado no botão, ele será direcionado para a tela de registro de eventos, figura 3.8, para registrar um novo evento relativo a esta máquina, podendo alterar o estado da mesma.

Figura 3.8: Tela de registro de eventos

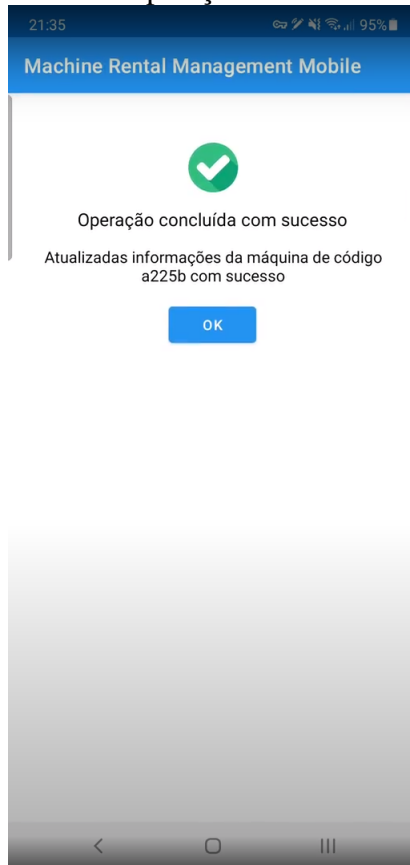


Fonte: O Autor

Como, a partir desta ação de alterar o estado da máquina, o aplicativo precisa

alterar dados que estão no servidor, é importante exibir uma mensagem de confirmação, caso a operação tenha sido executada com sucesso, ou de falha, caso tenha ocorrido algum problema. A figura 3.9 mostra a mensagem exibida no aplicativo quando a operação é concluída com sucesso.

Figura 3.9: Tela de operação concluída com sucesso



Fonte: O Autor

Assim, por exemplo, quando uma máquina está indo para manutenção, não podendo estar disponível para ser alugada, basta que o técnico abra o aplicativo, escaneie o código QR da máquina, e coloque ela em manutenção. A partir desse momento, se a equipe de gerenciamento for criar uma nova locação, esta máquina não estará disponível para ser alugada.

3.1.3 Servidor

O servidor é responsável pela gerência dos dados e dos modelos das entidades, através da sua interface com o banco de dados, e fornece acesso à eles através de uma Rest API. Tanto o website quanto o aplicativo mobile utilizam esta API para visualizar os

dados e realizar as suas operações.

Os modelos do sistema são: locações, aditivos, clientes, produtos, eventos de produtos, itens de locação, fornecedores e endereço. Para interagir com os modelos, o servidor fornece uma API Rest como descrita abaixo.

Modelo Locação (Rent Contract)

GET /api/rentcontracts - Retorna uma lista de todas as locações presentes no banco de dados, em formato JSON, com informações sobre seu cliente.

GET /api/rentcontracts/active - Retorna uma lista de todas as locações que estão em andamento, ou que tenham um aditivo em andamento, em formato JSON, com informações sobre seu cliente e aditivos.

GET /api/rentcontracts/revenue/from/:start_date/to/:end_date - Retorna a receita total para o período entre :start_date e :end_date no formato JSON. O formato esperado para o valor das datas é o seguinte: AAAA-MM-DD.

GET /api/rentcontracts/revenue - Retorna a receita total do mês atual e do mês anterior.

GET /api/rentcontracts/:id - Retorna a locação com id = :id, caso exista, com informações sobre seus itens de locação, produtos, aditivos e cliente.

POST /api/rentcontracts - Cria uma nova locação. Requer uma descrição da locação no corpo da chamada HTTP. Os produtos relacionados a essa nova locação terão seu status alterado para "RENTED".

PUT /api/rentcontracts/:id - Atualiza os campos da locação com o id = :id. Requer uma descrição da locação com os campos atualizados, em formato JSON, no corpo da chamada HTTP. Os produtos adicionados por meio desta operação não terão seu status atualizado.

DELETE /api/rentcontracts/ - Exclui todas as locações presentes no banco de dados.

DELETE /api/rentcontracts/:id - Exclui a locação com o id = :id do banco de dados.

Modelo Aditivos (Additives)

GET /api/additives - Retorna uma lista de todos os aditivos presentes no banco de dados, em formato JSON, com as informações da locação associada e seu cliente.

GET /api/additives/:id - Retorna o aditivo com id = :id, caso exista, com as infor-

mações da locação associada, seu cliente e todos os seus itens de locações.

POST /api/additives - Cria um novo aditivo. Requer uma descrição do aditivo, em formato JSON, no corpo da chamada HTTP. Retorna o status HTTP "201 Created" em caso de sucesso ao criar. Em caso de falha, retorna o código associado ao erro.

PUT /api/additives/:id - Atualiza os campos do aditivo com o id = :id no banco de dados. Requer uma descrição do aditivo com os campos atualizados, em formato JSON, no corpo da chamada HTTP.

DELETE /api/additives/ - Exclui todos os aditivos do banco de dados.

DELETE /api/additives/:id - Exclui o aditivo com o id = :id do banco de dados.

Modelo Clientes (Customers)

GET /api/customers - Retorna uma lista de todos os clientes no banco de dados, no formato JSON, com as informações de endereço.

GET /api/customers/pastdue/amounts - Retorna uma lista dos ids dos clientes que possuem dívidas vencidas, com o valor devido, e ordenado pelo valor.

GET /api/customers/pastdue/days - Retorna uma lista dos ids dos clientes que possuem débitos vencidos, com a soma dos dias vencidos em cada locação, ordenado por número de dias.

GET /api/customers/with_active_contracts - Retorna uma lista de clientes com locações ativas, em ordem decrescente de data de início da locação, com informações sobre a locação ativa e, se a locação estiver ativo por causa de um aditivo, informações sobre o aditivo. Se a locação não estiver ativa por causa de um aditivo, os campos referentes ao aditivo serão todos nulos.

GET /api/customers/:id/rentedItems - Retorna uma lista de itens de locação que estão atualmente em posse do cliente com id = :id, no formato JSON. Cada item de locação também contém as informações sobre a locação que o vincula a esse cliente.

GET /api/customers/:id - Retorna o cliente com id = :id, caso exista, com as informações de endereço, locações e aditivos vinculados a esse cliente, e também os itens de locação e produtos relacionados.

POST /api/customers - Cria um novo cliente. Requer uma descrição do cliente no corpo da chamada HTTP, em formato JSON.

PUT /api/customers/:id - Atualiza os campos do cliente com o id = :id. Requer uma descrição do cliente, com os campos atualizados, no corpo da chamada HTTP, em formato JSON.

DELETE /api/customers/ - Exclui todos os clientes do banco de dados.

DELETE /api/customers/:id - Exclui o cliente com o id = :id do banco de dados.

Modelo Produtos (Stock Items)

GET /api/stockitems - Retorna uma lista de todos os produtos presentes no banco de dados, no formato JSON.

GET /api/stockitems/list/needmaintenance - Retorna uma lista de todos os produtos que estão definidos como "necessitando de manutenção", no formato JSON, com os eventos de produtos associados e informações de fornecedores.

GET /api/stockitems/list/inmaintenance - Retorna uma lista de todos os produtos que estão definidos como "em manutenção", no formato JSON, com os eventos de produtos associados e informações de fornecedores.

GET /api/stockitems/:id - Retorna o produto com id = :id, se houver, incluindo todos os eventos de produto associados e informações do fornecedor.

GET /api/stockitems/:id/events - Retorna uma lista de todos os eventos de produto para o produto com id = :id, no formato JSON.

GET /api/stockitems/code/:code - Retorna o produto com code = :code, se houver, incluindo todos os eventos de produto associados e informações do fornecedor.

GET /api/stockitems/rented - Retorna uma lista de todos os produtos com status = 'RENTED', no formato JSON.

POST /api/stockitems - Cria um novo produto. Requer uma descrição do produto, no corpo da chamada HTTP, em formato JSON.

PUT /api/stockitems/:id - Atualiza os campos do produto com o id = :id. Requer uma descrição do produto atualizado, no corpo da chamada HTTP, em formato JSON.

PUT /api/stockitems/code/:code - Atualiza os campos do produto com o código = :code. Requer uma descrição do produto atualizado, no corpo da chamada HTTP, em formato JSON.

DELETE /api/stockitems/ - Exclui todos os produtos do banco de dados.

DELETE /api/stockitems/:id - Exclui o produto com o id = :id do banco de dados.

Modelo Eventos de Produtos (Stock Item Events)

GET /api/stockitemevents - Retorna uma lista de todos os eventos de produtos, no formato JSON, com informações do produto associado.

GET /api/stockitemevents/:id - Retorna o evento de produto com id = :id, se hou-

ver, com informações sobre o produto associado e seu fornecedor.

GET /api/stockitems/:id/events - Retorna uma lista de todos os eventos de produtos para o produto com id = :id, no formato JSON.

POST /api/stockitemevents - Cria um novo evento de produto. Requer uma descrição do evento de produto, no corpo da chamada HTTP, em formato JSON.

PUT /api/stockitemevents/:id - Atualiza os campos do evento de produto com o id = :id. Requer uma descrição do evento produto, com os campos atualizados, no corpo da chamada HTTP, em formato JSON.

DELETE /api/stockitemevents/ - Exclui todos os eventos de produtos do banco de dados.

DELETE /api/stockitemevents/:id - Exclui o evento de produto com o id = :id do banco de dados.

Itens de Locação (Item Rentals)

GET /api/itemrentals - Retorna uma lista de todos os itens de locação presentes no banco de dados, em formato JSON, com informações sobre seu produto, locação e cliente.

GET /api/itemrentals/:id - Retorna o item de locação com id = :id, caso exista, com informações sobre seu produto, locação e cliente.

POST /api/itemrentals - Cria um novo item de locação. Requer uma descrição do item de locação no corpo da chamada HTTP, em formato JSON.

PUT /api/itemrentals/:id - Atualiza os campos do item de locação com o id = :id. Requer uma descrição do item de locação, com os campos atualizados, no corpo da chamada HTTP, em formato JSON.

DELETE /api/itemrentals/ - Exclui todos os itens de locação do banco de dados.

DELETE /api/itemrentals/:id - Exclui o item de locação com o id = :id do banco de dados.

Modelo Fornecedores (Suppliers)

GET /api/suppliers - Retorna uma lista de todos os fornecedores presentes no banco de dados, no formato JSON, com as informações de endereço.

GET /api/suppliers/:id - Retorna o fornecedor com id = :id, caso exista, com as informações de endereço.

POST /api/suppliers - Cria um novo fornecedor. Requer uma descrição do forne-

cedor, no corpo da chamada HTTP, em formato JSON.

PUT /api/suppliers/:id - Atualiza os campos do fornecedor com o id = :id. Requer uma descrição do fornecedor, com os campos atualizados, no corpo da chamada HTTP, em formato JSON.

DELETE /api/suppliers/ - Exclui todos os fornecedores do banco de dados.

DELETE /api/suppliers/:id - Exclui o fornecedor com o id = :id do banco de dados.

Modelo Endereços (Addresses)

GET /api/addresses - Retorna uma lista de todos os endereços presentes no banco de dados, no formato JSON.

GET /api/addresses/:id - Retorna o endereço com id = :id, se existir.

POST /api/addresses - Cria um novo endereço. Requer uma descrição do endereço, em formato JSON, no corpo da chamada HTTP.

PUT /api/addresses/:id - Atualiza os campos do endereço com o id = :id no banco de dados. Requer uma descrição do endereço com os campos atualizados, em JSON, no corpo da chamada HTTP.

DELETE /api/addresses/ - Exclui todos os endereços do banco de dados.

DELETE /api/addresses/:id - Exclui o endereço com o id = :id do banco de dados.

3.2 Requisitos da aplicação

Como a empresa conta com uma equipe de administração, e utilizava uma série de aplicações e planilhas para dar conta dos processos, a obtenção dos requisitos pôde contar com várias fontes para cruzar os dados obtidos. Assim, o primeiro esboço dos requisitos iniciou com reuniões com a equipe administrativa da empresa, obtendo uma visão geral de como a empresa funciona. Essa equipe é formada por três pessoas, das quais duas são gerentes administrativas e financeiras, e uma é dona da empresa. Essas reuniões continuaram ao longo de todo o processo, porque dúvidas específicas surgiam ao longo da implementação. Os sistemas anteriormente utilizados pela empresa também entraram como fontes importantes para entender os casos de uso que já eram realizados ali, e que deveriam ser mantidos, mesmo que houvesse refinamento ao transformá-los em requerimentos. Por fim, a equipe de administração da empresa utilizava uma série de planilhas, e a sua funcionalidade foi transformada em requerimentos para o sistema, conforme conversado com a equipe da empresa.

3.2.1 Requisitos Não-Funcionais

Acerca dos requisitos não-funcionais do sistema, a primeira observação coletada é que o sistema precisaria estar disponível o tempo inteiro e acessível de qualquer lugar. Disso, se concluiu que a melhor alternativa seria o desenvolvimento do sistema através de um website hospedado em um servidor dedicado. Dessa maneira, o sistema pode ser acessado a partir de variados aparelhos como celular, tablet e computadores, e também de qualquer lugar, desde que o dispositivo tenha conexão com a internet.

Por uma série de fatores, coletados através de conversas com os técnicos da empresa, cerca de seis pessoas, o sistema precisa estar atualizado com o estoque o mais rápido possível. Por exemplo, para que as locações sejam bem atendidas com os produtos corretos, o estado atual de cada produto deve estar atualizado. Além disso, sempre que existe uma atualização do estado de um produto por parte dos técnicos, é importante que ela ocorra rapidamente, porque cada demora que vier a ocorrer nesse processo, é menos tempo que o técnico terá para trabalhar diretamente com os produtos. Dessa forma, se evidencia mais um requisito não-funcional que se torna necessário para o sistema. Ele precisa possuir um nível de performance que seja capaz de abarcar esses fatores.

Quando se trata do aplicativo mobile, existe um requisito de usabilidade necessário para a fácil operação por parte dos técnicos, já que eles não possuem grande conhecimento e experiência na utilização de sistemas, e não utilizavam nenhum aplicativo anteriormente.

3.2.2 Requisitos Funcionais

Pode-se dividir os requisitos funcionais do sistema em duas classes: os básicos e os customizados. Os requisitos funcionais básicos são aqueles mais estruturais, como a manipulação das entidades do sistema, e que foram de fácil obtenção. Entram nessa classe a criação, visualização, atualização e deleção das quatro entidades principais do sistema, que são as entidades do cliente, produto, fornecedor e locação. Também entra aqui a necessidade de uma tela inicial do sistema, com informações básicas como faturamento, quantidade de máquinas alugadas e clientes ativos, conforme requisitado em reunião remota com a equipe administrativa da empresa.

A segunda classe é a dos requisitos funcionais customizados, que são aqueles mais próprios dessa empresa, que fazem sentido para casos de uso específicos dessa empresa. Esses requisitos foram mais complexos de serem obtidos, e alguns deles sofreram altera-

ções ao longo do caminho de desenvolvimento do sistema. Aqui entram questões como faturamento da locação, exportação em PDF do contrato e fatura da locação, criação e visualização de aditivos para as locações, mudanças específicas no contrato exportável de cada locação, visualização do estoque da empresa em uma tela intuitiva, visualização e atualização de produtos por meio do aplicativo que lê o QR Code do produto. Um exemplo de um requisito funcional que mudou ao longo do caminho é o do faturamento. Primeiro, se estabeleceu que o sistema iria gerar o número da fatura no mesmo instante em que a locação fosse criada, independente de quando essa locação seria faturada. Depois, já na fase de testes do sistema, a equipe administrativa, em contato com os contadores, descobriu que o número da fatura precisaria seguir uma ordem relativa ao momento em que a locação é faturada. Assim, esse requisito mudou para que o número da fatura fosse gerado na hora em que a locação é faturada, não mais na hora da criação da locação.

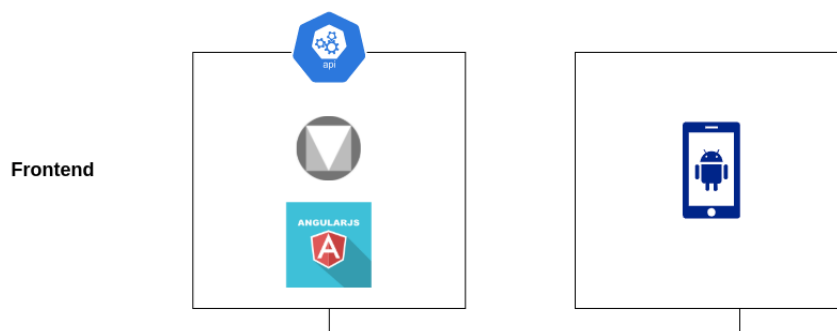
3.3 Arquitetura

Nesta seção, será descrita a arquitetura do sistema, que envolve o frontend, o backend e a infraestrutura do sistema.

3.3.1 Frontend

O frontend do produto é composto por duas aplicações. A primeira aplicação é um website, e a segunda, um aplicativo mobile Android. A figura 3.10 nos mostra uma visão geral do frontend.

Figura 3.10: Visão do frontend do sistema



Fonte: O Autor

O website, à esquerda na figura 3.10, é uma aplicação desenvolvida utilizando o

framework Angular, que é baseado em TypeScript. Acima do Angular, é utilizado o Google Material Design, que facilita o desenvolvimento da parte de estilo, proporcionando um aspecto bonito e padronizado em todo o website.

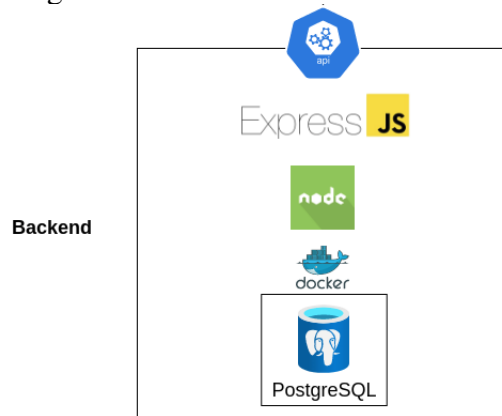
O aplicativo mobile android, à direita na figura 3.10, foi escrito utilizando Java, e é instalado nos celulares da empresa. O aplicativo tem a funcionalidade de ler códigos QR e se comunicar com o servidor backend. Os códigos QR estão colados nas máquinas, e cada um representa um código único associado a uma máquina específica. Através disso, os técnicos da empresa podem obter informações sobre uma máquina, e alterar o seu estado no banco de dados.

Através da ação dos técnicos, o sistema é alimentado com informação a respeito do estado das máquinas, mantendo o banco de dados atualizado. Uma máquina pode estar em cinco estados diferentes: “estoque”, “manutenção”, “alugado”, “pronto para sair” e “com o cliente”. Assim, existe um controle do estado das máquinas a partir do aplicativo, que era uma das demandas do cliente.

3.3.2 Backend

O backend do produto é um servidor NodeJS, que é um software baseado em JavaScript. Acima do NodeJS, é utilizado, para facilitar o desenvolvimento, o framework ExpressJS. O servidor é acessado através de uma Rest API, e deve permanecer ativo o tempo inteiro, para ouvir as requisições e servi-las. É este servidor que interage diretamente com o banco de dados, gerenciado pelo SGBD PostgreSQL, e instanciado utilizando Docker. A figura 3.11 nos mostra uma visão geral do backend.

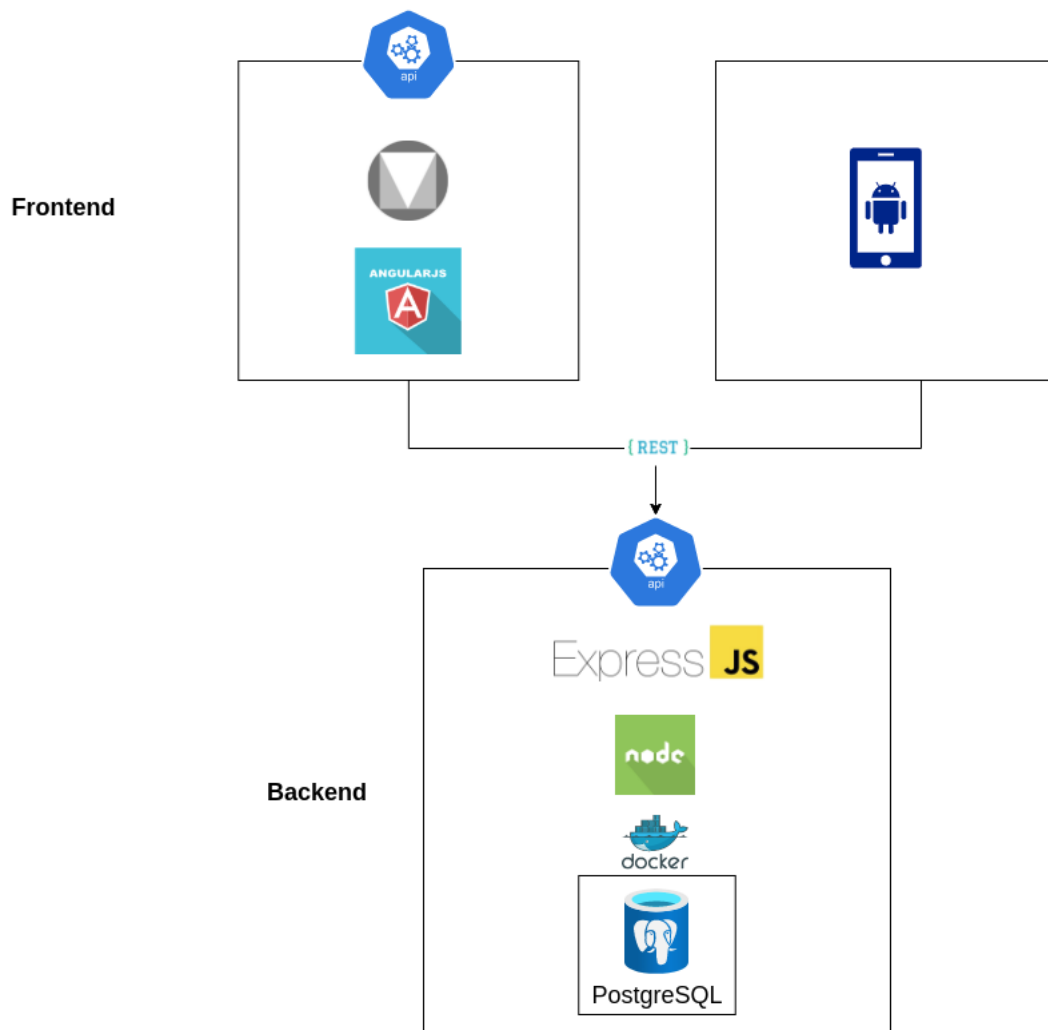
Figura 3.11: Visão do backend do sistema



3.3.3 Visão geral da arquitetura

Através da REST API definida no servidor backend, as demais aplicações frontend podem acessar e manipular os dados do sistema. Um bom exemplo para entender a visão geral da arquitetura como um todo, é que enquanto os técnicos utilizam o aplicativo de celular para atualizar o estado das máquinas, a equipe gerencial procura alocar essas máquinas da melhor forma possível para novos contratos de locação. A velocidade de atualização do estado das máquinas é diretamente proporcional à qualidade dessas alocações, já que cada cliente busca máquinas específicas, e uma máquina que retornou ao estoque esta manhã, já estará visivelmente disponível no sistema para novas locações. A figura 3.12 nos mostra uma visão geral da arquitetura do sistema, unindo o frontend com o backend.

Figura 3.12: Visão geral do sistema

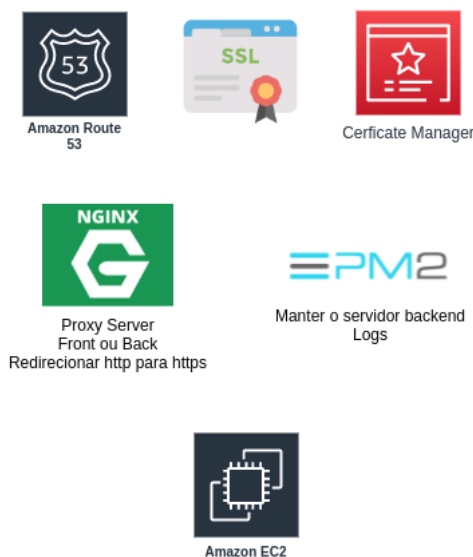


Assim, através dessa comunicação integrada entre as aplicações e o servidor, as manipulações de dados que acontecem por parte das aplicações frontend ficam disponíveis para as demais em tempo real.

3.3.4 Outros serviços de infraestrutura

Além das tecnologias utilizadas para construir o frontend e o backend especificamente, foi necessário uma série de tecnologias adicionais para criar a infraestrutura do sistema. A infraestrutura não envolve tanto desenvolvimento, sendo mais voltada para configurações e tecnologias necessárias para termos um software funcionando comercialmente. A figura 3.13 nos mostra um panorama geral dessas tecnologias, que serão detalhadas em seguida.

Figura 3.13: Serviços de infraestrutura do sistema



Fonte: O Autor

Visando a produção de um Software comercial, que realmente pode ser utilizado pela empresa, foi entendido que seria necessário estabelecer um domínio público de acesso ao website, e garantir a segurança do mesmo. Dessa maneira, alguns serviços disponibilizados pela AWS foram utilizados.

O domínio de acesso do website foi registrado através do serviço Amazon Route S3, que auxilia na compra e registro de domínios na internet, assim como na configuração DNS.

A API foi desenvolvida para suportar HTTPS. Para que esse protocolo pudesse ser

utilizado, foi necessário gerar um certificado válido e autenticado, através do serviço Certificate Manager, serviço emissor de certificados validados pela AWS. Depois de instalar o certificado no servidor, também se configurou a porta HTTPS (443) para que seja aberta. Dessa maneira, o usuário que acessar o website poderá verificar que o "cadeado" ao lado esquerdo do endereço do site estará fechado, sinal de que o Browser verifica e entende o certificado do site como válido, e, por conseguinte, considera o website como seguro. A porta HTTP (80) continua aberta, mas existe uma aplicação Nginx que roda no servidor como proxy, responsável por redirecionar todas as requisições HTTP para HTTPS.

Quando qualquer requisição chega ao servidor, é necessário saber se é uma requisição de arquivos de frontend, ou uma requisição de API para o backend. Essa necessidade se dá pelo fato de que, seja qual for, todas as requisições acontecem através do mesmo endereço de entrada. Dessa forma, a aplicação Proxy Server Nginx possui uma segunda responsabilidade. A cada requisição, o Nginx verifica se consta a palavra "/api/" na URL requisitada. Se sim, essa será uma chamada redirecionada para o backend, interagindo com a Rest API. Caso contrário, ela será redirecionada para o frontend, buscando exibir no browser do usuário os arquivos estáticos do site.

Como o resultado do frontend é composto de arquivos estáticos que são exibidos no browser do usuário, eles ficam parados no disco do servidor esperando serem servidos. O backend, de maneira contrária, é um serviço que deve ficar rodando constantemente para servir as requisições, interagindo com o banco de dados. Por isso, foi utilizada a aplicação PM2, que mantém o servidor backend rodando no background da máquina, e também facilita a visualização de logs do mesmo.

Por fim, todas estas coisas precisam estar em algum lugar, em alguma máquina. O serviço EC2 foi utilizado para essa finalidade. Na instância EC2, uma máquina Ubuntu, todas as configurações do servidor foram feitas, assim como a instalação do frontend e do backend. Este serviço gerencia automaticamente uso de disco, memória, rede, dados, e processamento.

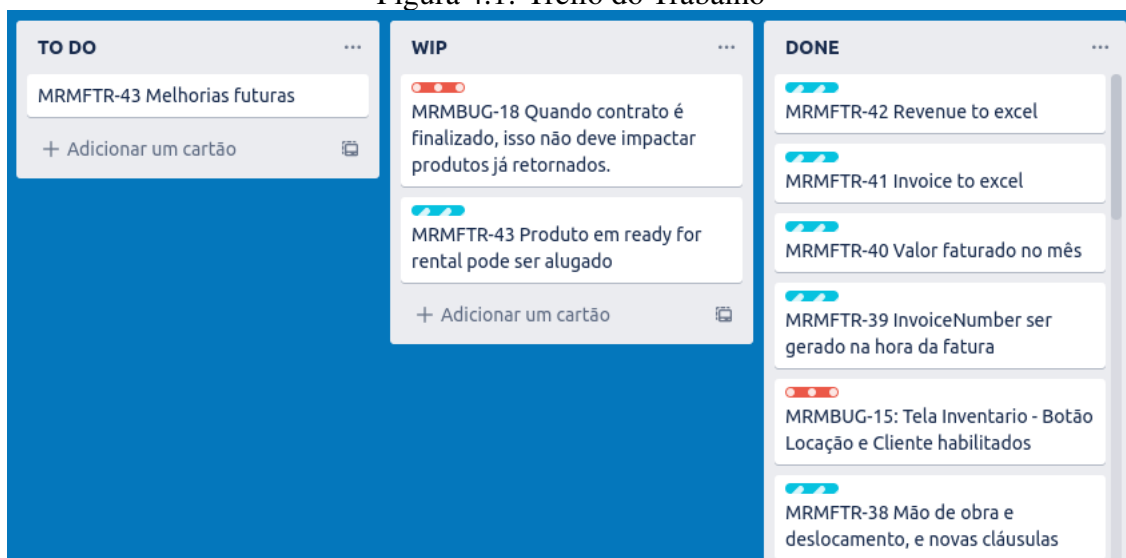
4 IMPLEMENTAÇÃO E AVALIAÇÃO

Neste capítulo, descreve-se como a implementação do sistema ocorreu, através de explicações acerca da metodologia utilizada, organização das tarefas, e das diversas fases do desenvolvimento. Também, fornece uma explicação detalhada de um caso de uso, e mostra-se como se avaliou o resultado final e as conclusões que se podem observar a partir dessa avaliação.

4.1 Implementação

A implementação foi realizada através de abordagem Ágil com Ressalvas, utilizando o framework SCRUM adaptado para um indivíduo, porque a equipe de desenvolvimento era de uma pessoa. Para maior organização, cada um dos requisitos e funcionalidades foram transformados em tarefas e organizados em um kanboard, utilizando a plataforma Trello para isso.

Figura 4.1: Trello do Trabalho



Fonte: O Autor

O autor representou os papéis de desenvolvedor, Scrum Master e Product Owner, e o cliente, que são os funcionários da empresa, representaram os stakeholders. Dessa maneira, o próprio autor priorizou as tarefas, mas com base naquilo que era necessário para os stakeholders, a empresa. Dessa maneira, foi possível, através de boa comunicação, mostrar para a empresa quais tarefas faziam sentido acontecer primeiro por um ponto de vista técnico, mas também priorizando as funcionalidades mais desejadas pela empresa

sempre que tecnicamente possível.

Algumas práticas e rituais não eram realizados, como por exemplo a reunião diária. O SCRUM para uma pessoa é uma metodologia ágil que adapta e aplica práticas de SCRUM a projetos de uma pessoa. Essa metodologia promove a produtividade individual através da observação, adaptação, elaboração progressiva, priorização e avaliação do peso das tarefas. Nesta abordagem customizada, as sprints são chamadas de fases.

4.1.1 Primeiros passos

O início do processo de implementação foi o entendimento, por parte do autor, de que havia espaço para informatizar a empresa. Enxergava-se a organização manual do estoque e a utilização das planilhas como dispendiosa. A partir disso, surgiu a primeira conversa com a empresa, que visou verificar a realidade dessa suposição.

4.1.2 Fase de requisitos

Uma vez entendida a ideia pela empresa, e a concordância entre os dois lados de que havia demanda para o sistema, passou-se a desenvolver a ideia em uma série de reuniões presenciais e remotas. A partir desse momento, iniciou-se a fase de requisitos, já buscando obter resultados concretos: os requisitos do sistema. Os requisitos obtidos foram transformados nos primeiros e mais básicos User Stories do backlog do projeto. Além disso, tarefas de infraestrutura foram definidas, e atividades como a produção de diagramas para ajudar na compreensão dos requisitos.

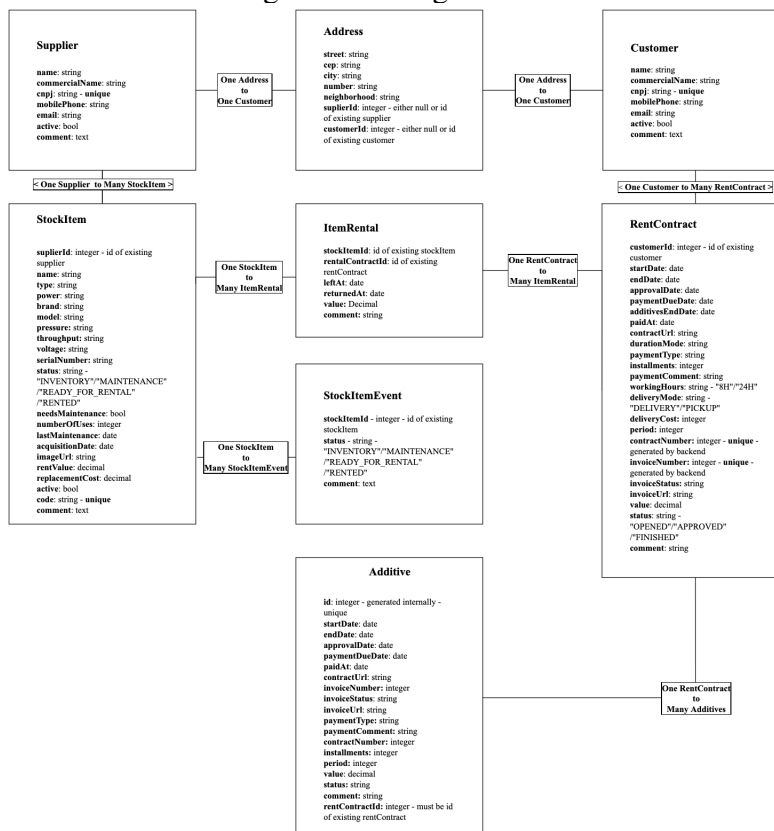
4.1.3 Fase de projeto

Depois da fase de requisitos, iniciou-se a fase de projeto. Com esse foco, foi produzido um diagrama de Entidade-Relacionamento, visando confirmar o que se obteve na fase anterior, alterar aquilo que fosse necessário, e fornecer uma primeira visão do projeto e de como o sistema seria implementado. O diagrama foi de fundamental importância para exemplificar os modelos do sistema e e manter uma boa comunicação com o cliente, aprimorando os requisitos e encontrando requisitos novos.

Este diagrama foi seguido com cuidado, e se mantém até hoje refletido no sistema.

Dessa forma, todos os modelos do sistema estão descritos no diagrama. Contudo, uma série de mudanças e adições de atributos em certos modelos foram necessárias para suportar funcionalidades que não estavam planejadas nessa fase de projeto. Um exemplo de funcionalidade não prevista nesta fase, que necessitou adicionar um possível estado a mais em um dos modelos do sistema, é a de poder reservar os produtos.

Figura 4.2: Diagrama ER



Fonte: O Autor

Todas as funcionalidades que foram vistas como necessárias para produzir um MVP foram escritas em um backlog, no formato de User Stories, seguindo os requisitos e ordenadas conforme prioridade.

4.1.4 Fase de implementação

Após ter uma visão ampla e também específica do sistema, iniciou-se a fase de implementação. Todas as User Stories foram transformadas em tarefas realizáveis pelo desenvolvedor. Algumas User Stories foram descartadas durante o processo de desenvolvimento, outras foram adicionadas, sempre em contato com o cliente. A comunicação

direta com o cliente durante todo o processo de desenvolvimento foi de suma importância para aprimorar as ideias, antes de implementá-las, e focar energias nas tarefas de maior valor. A comunicação se deu presencialmente e através de aplicativo de comunicação online.

4.1.5 Fase de testes

Assim que o MVP foi obtido, iniciou-se a fase de testes do produto. O foco dessa fase foi aprimorar e resolver problemas de visualização, criação, atualização e deleção dos modelos do sistema, assim como das suas relações. Além disso, uma quantidade considerável de ajustes, que antes não tinham sido pensados, foi requerida pela empresa. Todas as principais features necessárias para que o produto atendesse as demandas comerciais da empresa foram finalizadas e testadas.

4.1.6 Fase de produção

Depois da fase de testes, iniciou-se a fase de produção. O sistema foi atualizado com a versão mais recente, e o banco de dados foi reiniciado. A implantação do sistema como um todo foi realizada, incluindo o aplicativo mobile android, que foi instalado nos celulares da empresa. A partir desse momento, a empresa passou a utilizar o sistema de forma comercial.

4.1.7 Fase de manutenção

Nessa fase, que é a atual, pequenas features e ajustes são requisitados ao longo do tempo pela empresa. Os pedidos são transformados em tarefas, e realizados assincronamente. Quando entregues e integrados ao sistema em produção, a empresa é notificada.

4.2 Exemplo de caso de uso

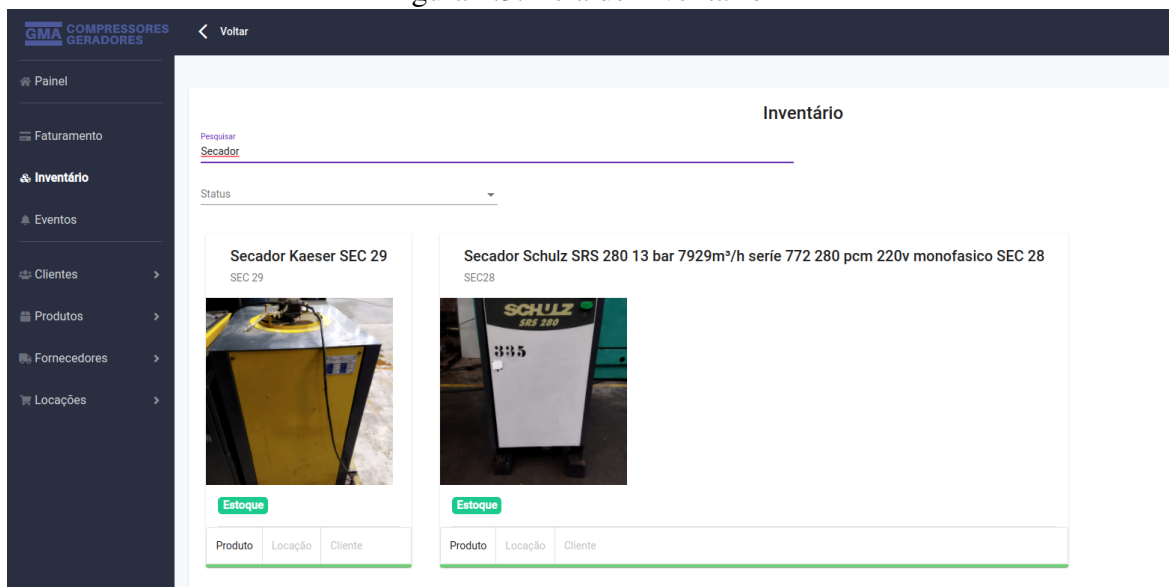
Um bom caso de uso, para exemplificar o sistema, é o de uma nova locação. Já está estabelecido, como ponto de partida, a existência de um cliente e de um produto, para

que sejam usados no decorrer do exemplo.

Esse caso de uso começa quando um cliente entra em contato solicitando um produto para alugar. Essa demanda pode surgir tanto pela internet, quanto por ligação, caso em que é necessário agilidade para saber se a empresa possui o produto necessário para as necessidades do cliente. O cliente fornece as especificações da máquina que precisa, como o tipo da máquina e a potência. Neste exemplo, será considerado como demanda um secador de dez cavalos de potência. Uma vez obtidas as informações acerca da máquina desejada, a equipe de administração pode utilizar o sistema para pesquisar os produtos atualmente no estoque, através da tela "Inventário", para encontrar uma máquina que cumpra as necessidades do cliente.

Na tela de inventário, que exibe todas as entidades de produtos do sistema, com o nome do produto, estado atual e foto, é possível filtrar pelo atributo desejado para verificar se há disponível em estoque algum produto com tais características informadas pelo cliente. Neste exemplo, basta inserir a palavra "secador" no campo de filtro para visualizar todos os secadores do inventário. Pode-se ver a tela de inventário, já com a pesquisa realizada, na figura 4.3.

Figura 4.3: Tela de Inventário



Fonte: O Autor

Depois de confirmada a disponibilidade do produto, e de verificado que o produto é suficiente para as cumprir as necessidades do cliente, a equipe de administração precisa criar um documento de contrato para enviar ao cliente, assim como atualizar o estado do produto, informando ao sistema que o produto em questão foi alugado. Os dois processos são realizados através da criação, no sistema, da entidade de locação.

Na tela de criar a locação, será informado o cliente, o produto, o preço da locação, e uma série de outras informações importantes, como data de início e término da locação, o custo do frete e o status da fatura. É normal que, futuramente, o mesmo cliente queira renovar a locação poderá futuramente ser renovada, caso desejado pelo cliente, através de aditivos, outra entidade do sistema, que é exibida na segunda aba da tela da entidade de locação. A tela para criar a locação é exibida na figura 4.4.

Figura 4.4: Criar locação

Criar Locação

Cliente * Produtos *

Produtos Selecionados

Nome	Status	Tipo	Potência	Valor
Tipo de duração <input checked="" type="radio"/> Duração personalizada <input type="radio"/> 15 dias <input type="radio"/> 30 dias		Data de Início * 26/08/2022	Período 8 Horas	Status Aprovado
Frete Retirar	Custo do Frete R\$0	Custo Mão de Obra e Deslocamento R\$0	Status da Fatura Pendente	
Valor Total * R\$0		Data em que foi pago	Fatura	

Salvar

Fonte: O Autor

Uma vez gerada a locação, já é possível exportar o contrato referente a ela, que é gerado dinamicamente através do botão com o texto "Exportar Contrato" na tela que lista as locações. Com o contrato em mãos, a equipe de administração envia-o para o cliente. Toda a formatação do contrato, assim como as suas cláusulas e informações, foram definidas previamente em contato com a equipe administrativa, e são sempre iguais independente do cliente. Existe um campo de observações caso seja necessário adicionar alguma especificidade a certo contrato. A figura 4.5 mostra um exemplo de um contrato.

Além disso, o produto relacionado a esta locação é colocado, pelo sistema, para o estado de alugado, e, portanto, não estará mais disponível para locações até que retorne para o estado de estoque. Uma vez que o contrato tenha sido assinado pelo cliente, a

A equipe não é grande, contando ao todo com 6 pessoas, e, por isso, a avaliação conta com um número reduzido de dados. Todavia, o fato dos entrevistados já estarem com a experiência de usar o sistema por quase um ano qualifica muito as respostas coletadas. Tanto a equipe de administração, composta por duas pessoas, como a equipe técnica, composta por quatro pessoas, responderam ao questionário.

O questionário foi formulado através da ferramenta Google Forms e é composto por doze questões, que foram divididas em duas seções de acordo com a tabela 4.1.

Tabela 4.1: Tabela mostrando as divisões das questões do questionário de avaliação

	<i>Nº de Questões</i>	<i>Tipo de Questão</i>
Seção 1	1	Relativo à idade
	1	Relativo ao sexo
Seção 2	3	Relativo ao nível de escolaridade
	2	Relativo à aplicabilidade do sistema
	2	Relativo à falta de funcionalidades no sistema
	2	Relativo à utilidade do sistema
	1	Relativo à recomendação do sistema

Fonte: O Autor

4.3.1 Perfil dos Participantes

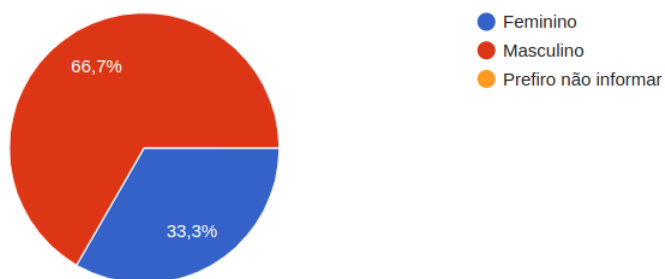
O objetivo desta parte da avaliação foi coletar informações importantes acerca dos participantes no que concerne ao perfil de cada um. Assim, viabiliza-se a categorização dos mesmos em grupos de usuários com certas características relevantes, entendendo melhor a sua finalidade e interação com o sistema.

Com base na experiência do desenvolvimento do sistema relatado por este trabalho, já é possível esperar diferenças relativas aos dois grupos de funcionários da empresa, que possui uma equipe administrativa e uma equipe técnica. Ambas as equipes foram entrevistadas.

Desta forma, a partir da análise das respostas da primeira seção das perguntas, através da figura 4.6, é possível perceber que, dos seis participantes, quatro são do sexo masculino, e dois do sexo feminino. Todos preferiram informar o gênero. A figura 4.7 nos mostra que a faixa etária é bastante diversa, englobando participantes que estão na faixa dos 18 anos até os 24 anos, até um participante que possui mais de 55 anos.

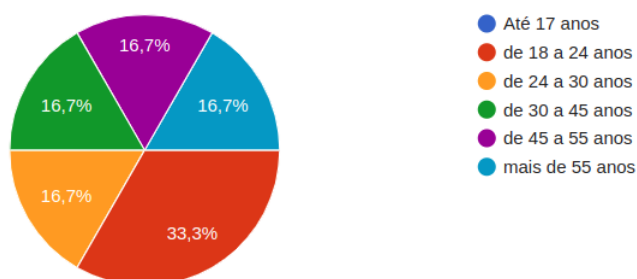
A figura 4.8 apresenta a distribuição do nível de escolaridade dos participantes. De acordo com ele, existe uma grande variação entre os níveis de escolaridade dos funcionários da empresa. É provável que a equipe de administração pertença ao grupo mais

Figura 4.6: Gráfico de distribuição por sexo



Fonte: O Autor

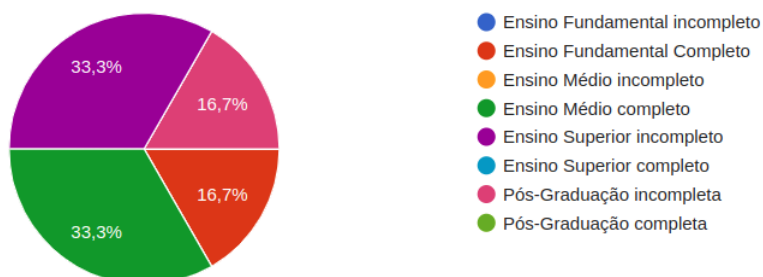
Figura 4.7: Gráfico de distribuição por faixa etária



Fonte: O Autor

bem formado, ao passo que os técnicos possuem um nível de escolaridade inferior. Isso se confirma através do outro gráfico que mostra a relação de qual curso superior foi ou está sendo cursado pelo participante. Dos três funcionários que ao menos cursaram o curso superior, dois deles cursaram cursos da área contábil ou comercial.

Figura 4.8: Gráfico de distribuição do nível de escolaridade



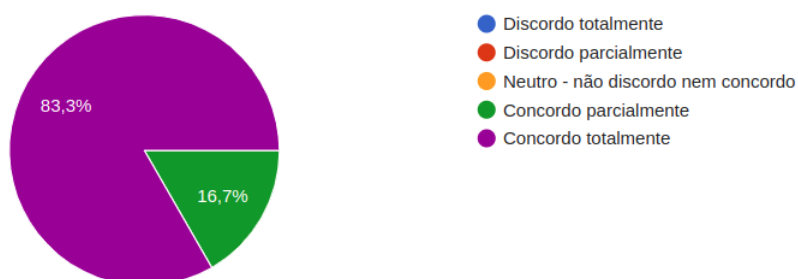
Fonte: O Autor

4.3.2 Análise dos Resultados

Um dos desafios do desenvolvimento de software é alinhar o que a parte interessada - no caso deste trabalho, a empresa GMA - precisa para resolver os seus problemas com aquilo que está sendo desenvolvido. É crucial que, em primeiro lugar, a parte interessada tenha muito claro as suas necessidades, os problemas que enfrenta, para que não se engane nas decisões a tomar durante todo o processo de criação do software. Em segundo lugar, é vital que o desenvolvedor saiba ouvir a parte interessada, para que não ocorra desvios de trajeto e se acabe desenvolvendo um sistema que, ao invés de suprir as necessidades e desejos da parte interessada, resolva problemas que estão somente na cabeça do desenvolvedor.

Dessa maneira, através da figura 4.9, pode-se perceber que o sistema desenvolvido relaciona-se com os problemas que os funcionários da empresa enfrentam, já que todas as respostas estão nos níveis altos de concordância com isso.

Figura 4.9: Gráfico das respostas para a pergunta “O sistema gma-admin relaciona-se a problemas que me deparo nas minhas atividades profissionais.”

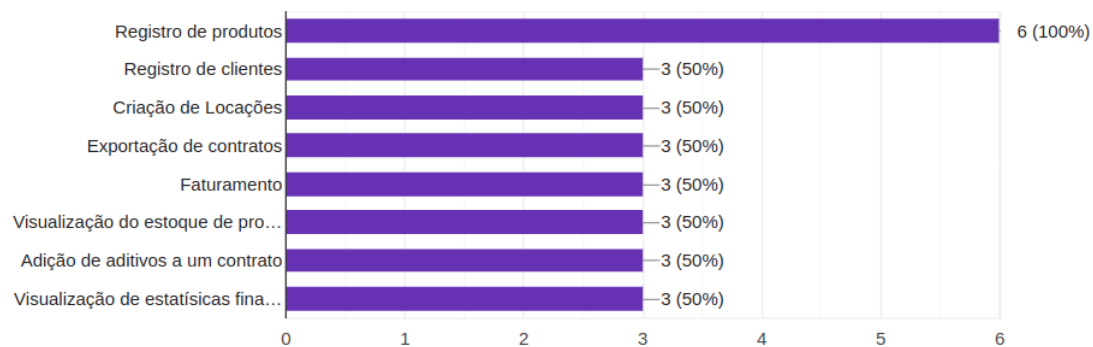


Fonte: O Autor

Segundo a figura 4.10, das funcionalidades desenvolvidas, somente uma é utilizada por todos os funcionários da empresa, o que já era esperado. Como os técnicos não interagem com o website, mas somente com o aplicativo, a única interação deles com o sistema é no registro do estado de produtos. Quanto às outras funcionalidades, que estão disponíveis somente através do website usado pela equipe administrativa, pode-se ver que no mínimo três pessoas as utilizam, o que significa metade dos funcionários. Nenhuma funcionalidade ficou sem uso.

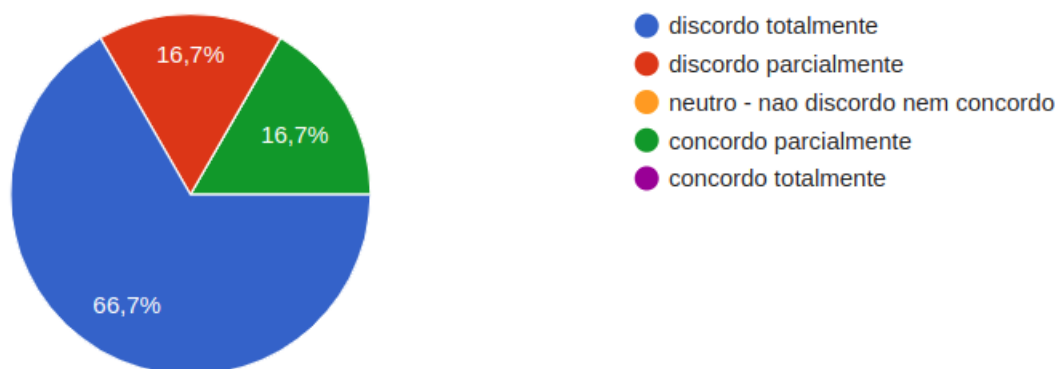
Também foi avaliado se alguma funcionalidade desejada foi esquecida, informação valiosa para o futuro do sistema. Através da figura 4.11, percebe-se que somente um funcionário sente falta de alguma funcionalidade no sistema.

Figura 4.10: Gráfico das respostas para a pergunta “As seguintes funcionalidades resolvem problemas que aparecem frequentemente para mim.”



Fonte: O Autor

Figura 4.11: Gráfico das respostas para a pergunta “Sinto falta no sistema de uma funcionalidade para resolver um problema específico”

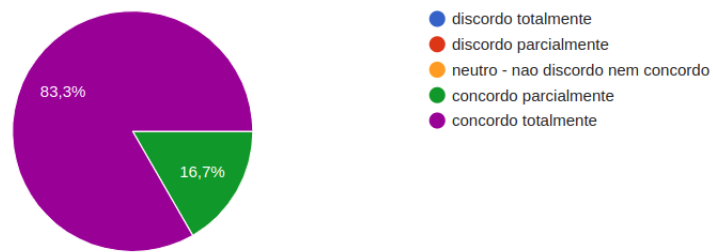


Fonte: O Autor

Para aqueles que responderam que sentem falta de uma funcionalidade no sistema, foi dada a oportunidade de descreverem essa funcionalidade, em uma caixa de texto. Assim, descobriu-se que a funcionalidade desejada é a possibilidade de ver o cliente da empresa que alugou uma máquina através do aplicativo. O fato da funcionalidade desejada ser no aplicativo nos indica que esse funcionário é da equipe técnica. Por escolha da equipe administrativa, nem todos os técnicos foram incluídos nas discussões a respeito do sistema, e, por isso, faz sentido que essa funcionalidade não tenha sido desenvolvida.

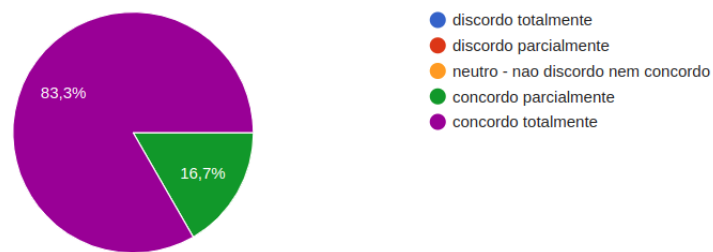
O sistema foi desenvolvido com base nos problemas que a parte interessada sofria, mas outro fator importante é o da utilidade do sistema. Com base nas figuras 4.12 e 4.13, percebe-se que todos os funcionários concordam que o sistema é útil para as suas respectivas atividades, e que usaria o sistema para resolvê-las.

Figura 4.12: Gráfico das respostas para a pergunta “O sistema é útil para as minhas atividades.”



Fonte: O Autor

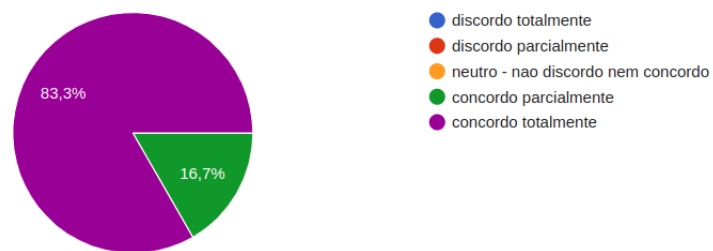
Figura 4.13: Gráfico das respostas para a pergunta “Eu usaria o sistema em minhas atividades.”



Fonte: O Autor

Por fim, a figura 4.14 nos mostra que todos os funcionários recomendariam o sistema para outros profissionais.

Figura 4.14: Gráfico das respostas para a pergunta “Eu recomendaria o sistema para outros profissionais.”



Fonte: O Autor

5 CONCLUSÃO

Neste capítulo, descreve-se os resultados do trabalho, levando em conta, também, a opinião do cliente. Além disso, aborda-se as limitações e possíveis melhorias para o sistema.

5.1 Resumo de resultados

O sistema foi desenvolvido com sucesso e já se encontra em pleno uso desde fevereiro. Obteve-se conhecimento acerca do processo de criação de um sistema real, assim como uma melhora nas habilidades de comunicação e aquisição de requisitos de clientes reais. O domínio do problema foi entendido, e uma série de novas funcionalidades para o sistema visando aumentar a eficiência dos processos já estão sendo planejadas. Foi realizada uma avaliação acerca do produto em funcionamento e, conforme os resultados desta, o sistema tem resolvido bem os problemas inicialmente percebidos, atendendo as necessidades descritas pela equipe da empresa. Ainda existem pequenos problemas para serem solucionados na fase de manutenção, mas que não estão impedindo nenhuma das funcionalidades do sistema.

5.2 Limitações e melhorias

É possível enxergar algumas limitações do sistema, como apenas funcionar tendo acesso à internet. Uma melhoria seria o funcionamento do sistema em modo offline. Para as funcionalidades que não dependam de banco de dados, essa mudança seria menos complexa. Para as que dependem, seria necessário realizar uma análise, verificando se poderiam operar sem acesso à internet, e, depois de obter o acesso, sincronizar os dados. Todavia, essa funcionalidade seria muito custosa para a empresa, porque o sistema teria que ser todo adaptado para uma aplicação desktop, ao invés de web. Por isso, não há planos para a sua execução.

Outra limitação é o fato do aplicativo mobile precisar necessariamente do QR Code na máquina para a sua leitura através da câmera. Assim, uma melhoria seria a possibilidade de reconhecimento por imagem ao invés de QR code. Para isso, poderia ser utilizada uma biblioteca open source de redes neurais, como a Darknet (PJREDDIE,

2022). Além do custo de implementação dessa melhoria, seria necessário gastar um bom tempo treinando a rede neural para que ela seja capaz de reconhecer cada máquina. Assim, não há planos para realizar essa funcionalidade no curto prazo.

Uma nova funcionalidade viável e que já está aprovada pela equipe da empresa é a de enviar notificações para o aplicativo dos técnicos assim que um produto for alugado. Dessa maneira, o processo de locação torna-se mais eficiente por não depender de uma comunicação manual da equipe de administração à equipe técnica. Com essa funcionalidade, uma vez que o produto tenha sido alugado, os técnicos terão ciência disso através da notificação, e o primeiro deles que estiver disponível já poderá preparar a máquina para o envio.

REFERÊNCIAS

MOZILLA **que é JavaScript?** 2022. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript>. Acesso em: 22 ago. 2022.

NODEJS **Sobre Node.js.** 2022. Disponível em: <<https://nodejs.org/pt-br/about/>>. Acesso em: 22 ago. 2022.

TABLELESS **O que é JavaScript?**. 2022. Disponível em: <<http://tableless.github.io/iniciantes/manual/js/>>. Acesso em: 23 ago. 2022.

TYPESCRIPTLANG **TypeScript for the New Programmer.** 2022. Disponível em: <<https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>>. Acesso em: 23 ago. 2022.

ANGULAR **What is Angular?.** 2022. Disponível em: <<https://angular.io/guide/what-is-angular>>. Acesso em: 23 ago. 2022.

ANDROID **O que é o Android.** 2022. Disponível em: <https://www.android.com/intl/pt-BR_br/what-is-android/>. Acesso em: 23 ago. 2022.

MOZILLA **Express/Node introduction.** 2022. Disponível em: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction>. Acesso em: 23 ago. 2022.

ACTLOC **ERP para Locadora.** 2022. Disponível em: <<https://actloc.com.br/>>. Acesso em: 23 ago. 2022.

REST **Representational State Transfer (REST).** 2022. Disponível em: <https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm>. Acesso em: 23 ago. 2022.

PJREDDIE **Darknet: Open Source Neural Networks in C.** 2022. Disponível em: <<https://pjreddie.com/darknet/>>. Acesso em: 26 ago. 2022.